Simplification for Efficient Decision Making Under Uncertainty with General Distributions

Andrey Zhitnikov

Simplification for Efficient Decision Making Under Uncertainty with General Distributions

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Andrey Zhitnikov

Submitted to the Senate
of the Technion-Israel Institute of Technology
Sivan, 5784 Haifa June 2024

This research was carried out under the supervision of Assoc. Prof. Vadim Indelman, in the Technion Autonomous Systems and Robotics Program (TASP).

The author of this thesis states that the research, including the collection, processing and presentation of data, addressing and comparing to previous research, etc., was done entirely in an honest way, as expected from scientific research that is conducted according to the ethical standards of the academic world. Also, reporting the research and its results in this thesis was done in an honest and complete manner, according to the same standards.

The author of this thesis received the following scholarships:

- Zeff Scholarship
- Jacobs Scholarship
- Excellence Scholarship Faculty Funding

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Vadim Indelman, for his endless support, and patience. I'm immensely thankful to him for believing in me and guiding me through these years. I would like to thank my mother Olga Zhitnikova for her continual support during my Ph.D. work.

The generous financial help of the Technion is gratefully acknowledged.

Publications

- Andrey Zhitnikov and Vadim Indelman, "Simplified Risk-aware Decision Making with Belief-dependent Rewards in Partially Observable Domains", Elsevier Artificial Intelligence 2022.
- Andrey Zhitnikov and Vadim Indelman, "Simplified Continuous High-Dimensional Belief Space Planning With Adaptive Probabilistic Belief-Dependent Constraints", TRO, Transactions on Robotics 2023.
- Andrey Zhitnikov, Ori Sztyglic, and Vadim Indelman, "No Compromise in Solution Quality: Speeding Up Belief-dependent Continuous POMDPs via Adaptive Multilevel Simplification", International Journal Of Robotic Research 2024.
- Andrey Zhitnikov and Vadim Indelman, "Safe Adaptive Belief-dependent Probabilistically Constrained and Chance Constrained Continuous Approximate POMDP Planning", submitted to JAIR, Journal of Artificial Intelligence Research.
- Andrey Zhitnikov and Vadim Indelman, "Anytime Probabilistically Constrained Provably Convergent Online Belief Space Planning", submitted to TRO, Transactions on Robotics.

Contents

List of Figures

Abstract

Markovian Belief Space Planning (BSP), also known as Partially Observable Markov Decision Process (POMDP) planning, is a necessary task in robotics and artificial intelligence. In a partially observable setting, due to uncertainty stemming from noisy sensors, imperfect actuation, and possibly an unknown environment, the robot makes decisions using the belief over the state instead of the state itself, which is hidden. Being undecidable in finite time in an exact way, the POMDP model gave rise to a multitude of approximations leveraging different assumptions. However, providing strict guarantees on the impact of such approximations remains a challenge. In this thesis, we take a different path. Instead of leveraging approximations, we are aiming to simplify the POMDP and provide guarantees on the impact of such a simplification. In addition, risk awareness is an indispensable part of robust and reliable autonomy. Yet, the classical POMDP formulation utilizes the expectation as the operator to compare the distributions stemming from future rewards. The expectation poorly accounts for risk. In this dissertation we tackle this gap.

This thesis includes five works. In the first work, we introduce the simplification paradigm, in conjunction with a risk aware objective. In our second work, we focus on the exploration problem. In this problem, the robot must explore an unknown map. Usually, the dimension of the map is large, therefore, the state over which the belief is maintained becomes high dimensional. Thus, such a POMDP exhibits a great computational complexity. This computational burden makes solving this problem in real time a major challenge. At the core of our second work is our novel formulation of a belief-dependent Probabilistic Constraint. We utilize this constraint to speed-up the autonomous exploration or serve as a stopping exploration criterion. In our third work, we apply the simplification paradigm to a belief-dependent continuous POMDP. In support of this concept, we present several contributions. We begin with a thorough description of a novel and general theoretical framework of the simplification. We, then, discuss two settings, a given belief tree and a Monte Carlo Tree Search (MCTS). In both settings, we accelerate POMDP planning without compromising the quality of the obtained solution. In our fourth work, we consider our probabilistically-constrained belief-dependent POMDP again. Here we focus on the safety aspect. In this work, we suggest fast algorithms for probabilistically-constrained and chance-constrained nonparametric continuous POMDP, considering the setting of a given belief tree. We continue to the setting of MCTS in our fifth paper. We demonstrate the benefits of our methods on several robotics related problems, namely active Simultaneous Localization And Mapping (SLAM) for mobile robots, autonomous navigation to the goal, sensor placement, autonomous robotic cleaner, and target tracking.

Overall, the results of this research improve the robot's efficiency and quality of online decision making under uncertainty.

Abbreviations

AI Artificial Intellegence. 7, 8

AO* A heuristic search procedure for AND/OR graphs. 8

BMDP Belief-MDP. 6–8

BRM Belief Road Map. 8

BSP Belief Space Planning. 1, 4, 5, 7, 9, 10

CVaR Conditional Value at Risk. 9

DESPOT Determinized Sparse Partially Observable Tree. 7, 8

FIRM Feedback-based Information RoadMap. 8, 9

HSVI Heuristic Search Value Iteration. 7, 8

IPFT Information Particle Filter Tree. 8

LC Lipshitz Continuous. 8

LIDAR Laser Imaging, Detection, And Ranging. 4

MAB Multi Armed Bandit. 9

MCTS Monte Carlo Tree Search. 1, 7, 8

MDP Markov Decision Process. 6, 7, 9, 192

PDF Probability Density Function. 6

PFT Particle Filter Tree. 8

POMCP Partially Observable Monte Carlo Planning. 7, 9

POMDP Partially Observable Markov Decision Process. 1, 4–8, 10

PSR Predictive State Representation. 4

PWLC PieceWise Linear Convex. 6, 8

RAO* Risk Bounded AO*. 8

RRBT Rapidly-exploring Random Belief Tree. 8

SARSOP Successive Approximations of the Reachable Space under Optimal Policies. 7, 8

SLAM Simultaneous Localization And Mapping. 1, 4, 9

SS Sparse Sampling. 7, 189, 192

UCB Upper Confidence Bound. 7, 8

VaR Value at Risk. 188, 189

Nomenclature

$x = y$ The two random variables x and y are equal if they are equal as functions on their measurable space: $x(\omega) = y(\omega) \quad \forall \omega$.

$\mathbf{1}_A(\cdot)$ Indicator function defined on set A. Typically set deined by inequality so we indicate only the inequalty as the set. Given a probability space $(\Omega, \mathcal{F}, \mathrm{P})$ with $A \in \mathcal{F}$, the indicator random variable $\mathbf{1}_A \colon \Omega \mapsto \mathbb{R}$ is defined by $\mathbf{1}_A(\omega) = 1$ if $\omega \in A$, otherwise $\mathbf{1}_A(\omega) = 0$.

$a \vee b$ $\max\{a, b\}$ where $a, b \in \mathbb{R}$.

$a \wedge b$ $\min\{a, b\}$ where $a, b \in \mathbb{R}$.

$f \equiv g$ for two functions f, g if we have $f(x) = g(x) \quad \forall x$.

Chapter 1

Introduction

The demand for autonomous systems has increased drastically in various fields such as autonomous navigation, robotic arm operations, humanoid robots, and artificial intelligence. One of the most important applications is autonomous vehicles operating as a single autonomous entity and relying solely on onboard sensors. Other examples include unmanned aerial robots for inspection and testing, agricultural robots, etc. Applications are ubiquitous. Decision making under uncertainty is at the heart of any autonomous system acting with imperfect information. The renowned framework for making decisions using incomplete information is the POMDP. In a partially observable setting, which is common in real world scenarios, there is no direct access to the POMDP state. Instead, the robot must make a decision relying on the history of performed actions and received observations in an interleaving manner. The history also includes the prior knowledge about the POMDP state in the form of a distribution over the state, which is called the prior belief.

The robot utilizes the history in the form of Predictive State Representation (PSR) [32], [19] or distribution over the POMDP state given the concurrent history, the belief. In this thesis we focus on the **belief** approach. The belief is the posterior distribution over the state given all information available up to the current time. The state could regard the robot's position (pose) as in the localization problem where the map is known and given. In the more challenging active SLAM problem, the state also consists of a map to be explored.

The robot maintains a belief conditioned on the available history at each time instant. This history is obtained by the robot while it performs its task (Inference rectangle in Fig. 1.1). In the planning stage, the posterior belief about the POMDP state serves as input to the planner (Orange rectangle in Fig. 1.1). Within the decision making stage the robot simulates future histories, maintains corresponding beliefs, and reasons about their evolution while accounting for different sources of uncertainty. There are two types of uncertainty, namely outcome uncertainty and state uncertainty. The outcome uncertainty stems from imperfect robot actuators and materializes in the form of a probabilistic transition model. The state uncertainty is the result of observing the state through the lens of imperfect onboard sensors. Examples of sensing devices include a camera or Laser Imaging, Detection, And Ranging (LIDAR). When the robot operates in the field, it perceives the world with the available sensors, gathers information, and plans its next action(s) (Fig 1.1). The operation continues in a cyclical interleaving manner of information gathering using onboard sensors and BSP.

This research focuses on the active problem of planning under uncertainty in the robot's loop, Fig 1.1. When a robot performs planning in the belief space, it must solve a POMDP. Solving a POMDP, i.e., calculating the "right decision" in terms of an optimal action sequence or policy, involves anticipating every imaginable turn of future events with a predefined number of steps ahead into the future (the planning horizon). Each future event is defined by the simulated future history and the corresponding belief. The robot's task is defined by a belief-dependent reward function. In this thesis, we focus on information-theoretic rewards such as differential entropy, expected (with respect to belief) distance to goal, and minus trace of the covariance matrix of the belief. The future instant rewards, that correspond to a future observations episode (decision epoch or script) under a particular execution policy, are then combined into a single value called the *return* (long-term reward in some papers, e.g. [38]). One typical example of the return is the future cumulative reward.

The POMDP planning can be done with finite or infinite horizons. In a finite horizon setting, the agent performs planning with a fixed number of steps ahead of time. As the horizon grows, the robot can contemplate and reason about a more distant future. Within the planning session, equipped with motion and observation models, the agent must ponder over every possible realization of the future observations episode, of the length of the horizon, for every available reactive action sequence (policy). For each realization of the future observations and actions episode (decision epoch or script), it calculates a series of beliefs the size of a horizon, on each of which it calculates a reward and sums up the rewards. The realization of a cumulative reward is a single realization of the future. In a sampled form, this abundance of possible realizations of action-observation pairs constitutes a belief tree (Fig. 1.2). If the belief-dependent reward is merely the expectation, with respect to belief, of a state-dependent reward, the belief tree will not necessarily contain the entire belief at each node.

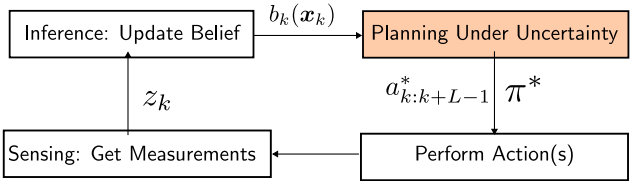


Figure 1.1: Deployed robot's loop. By $b_k(\boldsymbol{x}_k)$ we denote the belief in time index k over possibly high dimensional state $\boldsymbol{x}_k$. Output of BSP is the best static action sequence $a^*_{k:k+L-1}$ or reactive one π^*, namely the policy. Having performed an action/actions the robot senses environment and obtain measurements z_k. With performed actions and received observations it updates its belief.

In this case, instead of the belief tree, the robot has a history tree with different belief approximations at each node [41]. In our setting of general belief-dependent rewards the robot has a belief tree. Building the full belief tree is intractable since each node in the tree repeatedly branches with all possible actions and all possible observations. The number of belief nodes grows exponentially with the horizon. Here emerges the problem: to be more productive and reach the goal faster, the robot must cogitate (speculate) about a more distant future. Namely, to increase the horizon as large as possible. However, a larger horizon corresponds to a deeper belief tree, meaning a heavier decision making problem from the computational point of view. Therefore, choosing an optimal action (policy) is exceptionally computationally demanding and prohibitively expensive. POMDPs are notoriously hard to solve. This issue is exacerbated when the state over which the belief is maintained is high dimensional.

Alternatively, BSP (POMDP) can be formulated with an infinite horizon. Each immediate reward is multiplied by a non-negative discount factor smaller than one [42]. Such an infinite series is convergent if the reward is bounded [45]. The agent is given a terminating action or terminating condition on the belief. Using dynamic programming under the assumption of bounded reward and a not too large discrete state, action, and observation spaces, POMDP with an infinite horizon can be solved exactly. This means that, in some limited cases, reasoning about the belief tree can be avoided. Instead, one employs tabular, also known as full width methods [37]. We focus in this thesis on continuous state and observation spaces, leading to a continuous space of the reachable future beliefs space at any future time instance. The reachable future beliefs space is the space constituted by the future beliefs, in a particular time index, that can be obtained from the belief given as a result of the inference stage (Fig. 1.1). This belief is fed to the planner. As a result of the inference stage, the robot can have any belief. All in all, even to employ tabular methods for a given belief, one shall visit an infinite uncountable number of beliefs. This means that, in continuous domains, the robot has to solve BSP online. One way to do that is to resort to sampling and construct a belief (history) tree such that, even the infinite horizon formulation is approximated by the belief tree of the depth as large as possible. Offline approximate methods do not produce satiable results.

General belief-dependent rewards such as information theoretic rewards are essential for BSP. However, rewards of this type introduce an additional computational burden. The common POMDP formulation assumes that the belief-dependent reward is *nothing but an expectation, with respect to belief, over the state reward*. So far we found a single state dependent reward, minus distance to a goal state, which accounts for uncertainty when averaged with respect to belief. We show a proof of that claim in our fourth work. In many other continuous problems, e.g., autonomous exploration, localization, and sensor deployment, the reward is belief-dependent. It can be a minus trace of the covariance matrix of the belief or a minus of differential entropy. These rewards cannot be represented as an expectation of a state dependent reward.

Typically the algorithms for online planning under uncertainty use approximations without accounting for their impact on the decision making outcome. On the contrary, we focus on the *simplification paradigm*. To simplify the decision making problem means carefully examining all the components and possibly making them more simple or coarse through accounting for the impact of the simplification. An inherent part of the simplification paradigm is the *guarantees* on the effect of the simplification.

Our key insight is as follows. When the candidate action sequences or policies are far enough in terms of the objectives over the returns, computationally demanding components of the decision making can be simplified. Such a simplification incurs no impact on the precedence of the candidates in terms of the objectives. In other words, it preserves the trend. We call such a phenomenon *action consistency*.

We now provide a literature survey on the current state-of-the-art methods to make a decision under uncertainty with general belief distributions. This problem is often referred to as BSP, POMDP, stochastic control, and active perception. We aim to provide an overview of state-dependent and belief-dependent POMDPs solvers that provide an approximate solution [30, 37, 48, 41, 13] as well as safety and risk aware approaches [6], [14] [36].

POMDP [21] has proven to be a celebrated mathematical framework for planning under uncertainty. POMDP consists of a transition (motion) model, observation model, reward/cost for being at some particular state, and

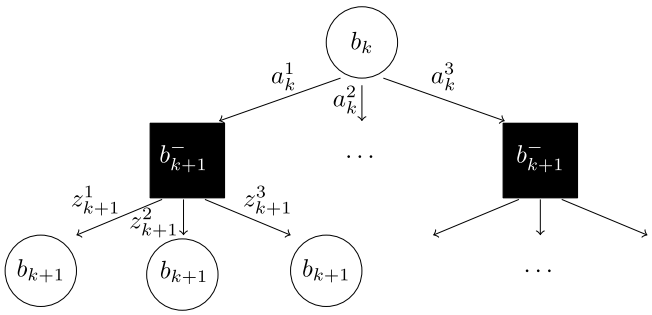


Figure 1.2: Objective function tree, b_k is the belief at time k, a_k^1, a_k^2, a_k^3 are the actions, b_{k+1}^- is the belief at time $k+1$ propagated with action applied at time index k (belief action node), $z_{k+1}^1, z_{k+1}^2, z_{k+1}^3$ are sampled observations.

executing some action. The transition model is the Probability Density Function (PDF) of being at some momentary state given the previous momentary state and action (control). The observation model is the PDF to receive some observation given the momentary POMDP state. Notably, such POMDP formulation can be converted to Belief-MDP (BMDP), where the belief is the state of the resulting Markov Decision Process (MDP) and the reward over the belief is the expected, with respect to belief, reward over the state or general belief-dependent reward. The fundamental problem in POMDP is to devise an optimal policy from the belief to the action. An equivalent definition of the meaning of solving POMDP is to find a policy from the history to the action. The belief is merely a way to accumulate and represent the history. A conventional and most common objective operator is the expected value over the returns. The journey to solve POMDP started from offline methods [26],[25]. These methods are designed to perform most of the, unbearable online to the robot, computations before execution. As we mentioned before, in very small discrete problems it is possible to visit all the states and observations.

These earlier offline attempts are based on dynamic programming and provide poor performance when the state and observation spaces grow.

Algorithms to solve POMDP online operate on the belief tree (Fig. 1.2) or history tree [41] depending on the reward structure. The exponential growth with the horizon is not the only problem of belief tree based approaches. Additionally, the number of possible states grows exponentially with the state space dimension, and consequently, an adequate representation of the belief requires more particles. Those last two problems are known as the *curse of history* and the *curse of dimensionality* respectively.

Some of the approaches are designed solely for discrete spaces. Further, we elaborate separately on each method whether or not it extends to continuous domains and general belief-dependent rewards. Note that in large discrete spaces and continuous spaces, one should resort to sampling. However, in continuous spaces, the probability of receiving the same sample twice is zero. Therefore, continuous spaces require additional treatment.

Offline solvers and state-dependent rewards Classical offline methods [25] are intended to find offline a policy that is optimal for all possible beliefs. These methods are based on α-vectors and point-based value iteration [30, 34, 38]. The α-vectors approach leverages the fact that the reward is state-dependent and the belief-dependent reward is obtained by the averaging with respect to belief. Another name for policy is the reactive plan [25]. One can interpret a policy as a conditional plan represented as a tree. We also can define a conditional subplan as a policy associated with a particular observation, see [25]. Each possible observation defines a conditional subplan. Note that since the reward is state-dependent, in an infinite horizon setting, the conditional subplan depends on the observation instead of the entire history. This is because in discrete spaces we exhaustively enumerate all the states and the observations. In a finite horizon setting, the conditional subplan depends also on the horizon and it gets smaller when the time index progresses. The α-vector is the vector of the values of state-dependent utility functions, starting from the state realizations or samples from the belief distribution [13], under the conditional plan, the set of α-vectors, each annotated with an action, can represent the policy for all beliefs. The application of such policy is to find α-vector maximizing the inner product with the belief, meaning maximizing the utility function. The action appropriate to such α-vector is optimal. Importantly, representing optimal value function as the maximum over α-vectors of the inner product of an alpha vector with the belief indicates its PieceWise Linear Convex (PWLC) property and enables pruning suboptimal α-vectors safely. Alternatively, a one-step look-ahead can be used to avoid keeping track of the actions associated with the alpha vectors. This strategy utilizes the Bellman form of the belief-dependent utility function and selects using α-vectors the maximal utility in the next step, appropriate to every possible action and observation.

Point-based value iteration is based on a set of representative beliefs. For each belief, one α-vector is associated. Each α-vector is initialized, such that, when multiplied with the corresponding belief, it yields a lower bound of the optimal value function for this belief. The algorithm updates the α-vector of each belief. The backup process is applied to each belief from the set individually. It iterates through actions. For each action, it iterates through observations, updates belief with the action and the observation, and chooses the best α-vector from the set with regard to the posterior belief. It then performs a *backup* operation for each state using selected α-vectors appropriate to the observations at the next step. At the end of the loop through actions, the algorithm is left with one α-vector per action, and it returns the best one [26]. This algorithm improves the set of α-vectors. Since each α-vector is appropriate to the best action for the corresponding belief from the set, when a new belief comes for query the planner to return the best action, the best action is set by selecting the best α-vector. Point-based value iteration provides an approximate solution but substantially speeds up the exact solution by the α-vector method. These methods are designed for discrete state, action, and observation spaces. Moreover, α-vector is hyperplane [45] in the belief space, meaning that the reward over the belief is assumed to be merely expectation over the reward of the state as opposed to the general setting of belief-dependent rewards we consider.

The [38, 39] presented Heuristic Search Value Iteration (HSVI) algorithm which augmented the point-based value iteration method with a forward exploration heuristic leveraging lower and upper bounds of the optimal value function to the full extent. The heuristic that drives the exploration of the reachable belief space is the gap, obtained by the Bellman update, between the upper and lower bounds of the value function. Their lower bound is based on α-vectors and point-based value iteration described above. The upper bound is based on a set of points belief/value function bound, known as Sawtooth Upper Bound [25]. HSVI periodically prunes dominated elements in both the lower bound vector set and the upper bound point set. This approach received the name Sawtooth Heuristic Search [25]. While the lower bound is updated using a backup operation and adding a new α-vector to the set, the upper bound is updated with a local Bellman update, and adding a new belief/value bound point to the set. These bounds are limited to state-dependent rewards or specific forms of belief-dependent rewards which we discuss further. The α-vectors by definition can not accommodate a general belief-dependent reward. The Sawtooth Upper Bound leverages the convexity of value function and it is based on an initial set of belief-value pairs and has to contain all of the standard basis beliefs. Typically, Fast Informed Bound is used on the values corresponding to the beliefs from this initial set, which is again based on α-vectors [25]. In addition, evaluation of this bound on the new belief requires linear interpolation; so to remain bound for the new belief point it requires convexity of the optimal value function [15].

Algorithms which use point based value iteration and sawtooth upper bound require a set of beliefs. Algorithm Successive Approximations of the Reachable Space under Optimal Policies (SARSOP) uses successive approximations of the optimal policy to iteratively generate the above mentioned set of beliefs. The SARSOP algorithm build upon HSVI. Extension of the point-based value iteration solvers to continuous domains requires additional research. Although α-vectors are theoretically valid for continuous spaces, utilizing sampling may introduce additional complexities. As we will further discuss, [13] can possibly be extended to continuous spaces through determinization [25].

Online solvers and state-dependent rewards More recently, online methods have become successful. These algorithms are approximations and some of them are suitable for continuous state and observation spaces. The output of these methods is an action recommended for the current belief. The algorithm itself is a (random) policy that maps from beliefs to actions online.

Many of these algorithms were designed for an MDP setting of a fully observable state and further extended to POMDP. One example is the Sparse Sampling (SS) algorithm [22]. This algorithm applies to continuous MDP by sampling the next states from the generative model. Recursively, it opens all the actions until the depth is equal to the horizon. Naturally, it extends to POMDP through BMDP formulation. It can be thought of as a sampling version of forward search [26]. Another distinguished algorithm for MDP is MCTS. This algorithm tackles the curse of history by building a belief tree incrementally, and revealing only the "promising" parts of the tree. It can also operate with high dimensional and large state space since it only samples the state from a generative model (Similar to SS) so it applies to BMDP.

An inherent part of MCTS based algorithms is the exploration technique, e.g., Upper Confidence Bound (UCB) [3], [27]. The exploration technique is designed to balance exploration and exploitation while building the belief tree. Let us describe an out-of-the-box MCTS for discrete MDP. MCTS opens the belief tree by looping over the simulations (tree queries). Each simulation progresses with expanding state-action nodes and state nodes along the way in an alternating manner. To expand the state-action node from the upper level state node it firstly opens all the actions and then at each arrival to the state node it selects an action to go with UCB. It then samples the next state from the generative model, to create a state node. (Reward generally depends on the state and action emanating from it and the subsequent state.) If the drawn state is already a child of the current state action node, the simulation continues down the tree. In the case where the drawn

state is new, a rollout with random or offline policy is carried out from the new state and the algorithm backups up to the root.

Note that this out-of-the-box MCTS algorithm does not apply to BMDP with continuous state and observation spaces. In case that state space is continuous, the same state will never be drawn, resulting in an extremely shallow and non-representative tree. There are other renowned algorithms such as Partially Observable Monte Carlo Planning (POMCP) [37] (extension of MCTS to POMDP) and Determinized Sparse Partially Observable Tree (DESPOT) [48]. However, these algorithms employ the assumption/limitation that reward over the belief is the expected value of reward over the state. Therefore, they apply to Artificial Intellegence (AI) problems and are less relevant to BSP where the reward is a general function of the belief. Such rewards, e.g., information-theoretic, pose additional computational difficulty as we already mentioned. Moreover, information-theoretic rewards such as differential entropy are not bounded, rendering foundations of UCB invalid; Since UCB is based on Hoeffding bound [16]. To our knowledge, there is also no proof that differential entropy is Lipshitz.

Offline solvers and simple forms of belief dependent rewards Earlier attempts such as [2], [11], [7] were tackling offline solvers with simple forms of belief-dependent rewards, such as PWLC [2] or Lipshitz Continuous (LC) [11]. Concretely, the authors of [2] proposed to perform a piecewise linear approximation to the convex belief-dependent reward and extend point-based value iteration, e.g, HSVI [38]. The authors from [11] extended HSVI further to LC belief-dependent rewards. They showed that the Bellman update operator(also known as the Bellman optimality operator) preserves LC for finite horizons. The authors show how to define, initialize, update, and prune LC upper and lower bounding approximators of the optimal value function and derive a variant of the HSVI algorithm. Authors of [7] present SARISA - an extension of SARSOP to information-seeking actions. This algorithm is also limited to PWLC rewards.

Information theoretic reward in the context of online solvers for continuous POMDP Incorporation of information-theoretic reward into POMDP is a long-standing effort. In the context of robotics, such rewards are especially important for precise navigation. The robot should be able to steer itself to areas of high visibility to decrease uncertainty about its state. Info-theoretic reward allows robot to reason about its own uncertainty. Monte Carlo Tree Search made a significant breakthrough in overcoming the course of history. However, when the reward is a general function of the belief, the origin of the computational burden is shifted towards the reward calculation. Moreover, belief-dependent reward prescribes the complete set of belief particles at each node in the belief tree. Therefore, algorithms such as POMCP [37], and its numerous predecessors are inapplicable since they simulate, each time, a single particle down the tree when expanding it. DESPOT based algorithms behave similarly [48], with the DESPOT-α being an exception [13]. DESPOT-α simulates a complete set of particles. However, this algorithm depends on α-vectors. In particular, as in other DESPOT-like algorithms, the belief tree is determinized. Therefore, sibling belief nodes have identical particles and are distinct solely by their weights. DESPOT-α leverages this regard and uses the α-vectors to efficiently approximate the lower bound of the value function of the sibling belief nodes without expanding them. Since DESPOT-like methods are based on the gap heuristic search [25], this lower bound is an essential part of the exploration strategy. In other words, the DESPOT-α tree is built using α-vectors, such that they are an indispensable part of the algorithm. Note that an integral part of this approach is that the reward is state-dependent, and the reward over the belief is merely an expectation of the state reward with respect to belief. Thus, DESPOT-α does not support belief-dependent rewards since it contradicts the application of the α-vectors.

The only approach posing no restrictions on the structure of belief-dependent reward is the Particle Filter Tree (PFT) [41]. The idea behind PFT is to apply MCTS over BMDP. [41] augmented PFT with Double Progressive Widening and coined the name PFT-DPW. PFT-DPW utilizes the UCB strategy and maintains a complete belief particle set at each belief tree node. It applies to a continuous space, which is the space of the particle represented beliefs. Recently, [12] presented Information Particle Filter Tree (IPFT), a method to incorporate information-theoretic rewards into PFT. The IPFT planner is remarkably fast. It simulates small subsets of particles sampled from the root of the belief tree and averages entropies calculated over these subsets. However, differential entropy estimated from a small-sized particle set can be significantly biased. This bias is unpredictable and unbounded and, therefore, may severely impair the performance of the algorithm. The authors from [12] provide guarantees solely for the asymptotic case, i.e, the number of state samples (particles) tends to infinity. Asymptotically their algorithm behaves precisely as the PFT-DPW in terms of running speed and performance. Yet, in practice, the performance of IPFT in terms of optimality can degrade severely compared to PFT-DPW. Moreover, [12] does not provide any reliable study of comparison of IPFT against PFT-DPW with an information-theoretic reward.

Risk aware, chance-constrained and robust POMDP The discussed algorithms stem from the AI community and are designed for general POMDP problems. The Robotics community is primarily concerned with navigation problems. In these problems, notions of *safety and risk* are fundamental. One example is collision

and obstacle avoidance. Some prominent attempts are Belief Road Map (BRM) [35], Rapidly-exploring Random Belief Tree (RRBT) [6], and Feedback-based Information RoadMap (FIRM) [1] which consider Gaussian beliefs. The paper [6] explicitly defines the probability of collision as a chance constraint of the decision-making problem. The authors utilize closed-loop control within the planning and consider the probability of collision on top of the controller steering the robot to the nominal trajectory. Another example of chance constraint can be the probability that the path is too long and that fuel in a flying vehicle will run out before reaching the goal. Namely, for a possible sequence of future observations (episode) and the beliefs, along a subset of state trajectories, is considered to be valid. A more recent algorithm tackling general distributions but discrete state and observation spaces is Risk Bounded AO* (RAO*) [36]. This algorithm extends A heuristic search procedure for AND/OR graphs (AO*) [33] to the Risk-bounded variant. The authors of [36] exemplify their algorithm on automated planning for the science agents problem, visualizing the importance of the chance constraints. The science agent starts from some initial position and operates on the map with obstacles. The agent can visit four different sites on the map. At each site, it can find a discovery with some probability. Since the agent's position is uncertain, collision is probable. The agent is required to finish at the relay station. In a limited time, the agent has to gather as much information as possible and arrive at the relay station. The duration of each traversal is uncontrollable but bounded. The authors of [36] used a single chance constraint that the event "arrives at the relay location on time" happens with a probability of at least some given parameter. There are other notable works, e.g., [23] combines POMCP and FIRM, and very recent [47].

Simplification paradigm The computational burden incurred by the complexity of POMDP planning inspired many research works to focus on approximations of the problem, e.g., [17]. Typically, approximation based planners show asymptotical guarantees, e.g., the convergence of the algorithms. Recently, the novel paradigm of simplification has appeared in literature [44, 8, 18, 24]. The simplification is concerned with carefully replacing the nonessential elements of the decision-making problem and quantifying the impact of this relaxation. Specifically, simplification methods are accompanied by stringent guarantees, while alleviating the computational burden of the decision-making problem.

Thus far, simplification has been utilized in the high-dimensional setting with Gaussian beliefs. However, general belief distributions and risk-aware formulations received less attention. Recently [43] developed novel adaptive bounds on a differential entropy estimator based on a belief representation by weighted particles [5]. A technique that appears to be close to our notion of simplification is described in [17]; where the authors describe successive multi-level approximations of the models from coarse to fine. Another interesting work utilizes confidence intervals to eliminate actions in the context of MDP and Multi Armed Bandit (MAB) problem [9].

The extremely challenging high dimensional active SLAM problem was addressed when the belief is modeled as a Gaussian [28, 29], as well as general belief distributions [10]. These works reduce computational complexity by reusing the calculations between common parts of non-myopic candidate actions [29], and between planning sessions [10]. Another line of works builds upon the *simplification paradigm* in the context of Gaussian beliefs such as [8] sparsifies the belief, while [24] bypasses costly calculations, substituting the reward by a topological signature.

The remainder of this thesis is organized as follows. In Chapter 2 we discuss our methodology. We then continue to the Chapter 3 where we present our published papers. Chapter 4 is a collections our unpublished papers. Chapter 5, the conclusion, discusses all the published and unpublished papers in this thesis.

Chapter 2

Methodology

In this chapter, we discuss our methodology. Works such as [46], [9], [38] are especially close to our methods. In general, our approach leans on adaptive bounds over the objective. Let us, in this section, give an example of the bounds usage, namely providing guarantees using confidence intervals. Taking inspiration from [22], we will construct confidence intervals over all the candidate action sequences simultaneously. We will see that stochastic intervals do not convey the desired information. We would like to know absolutely if any impact has been made by simplification. This question translates to the presence of any overlap of the intervals, for each candidate policy, induced by the lower and upper bounds over the objective. Let us consider the decision-making with static action sequences. We will now describe how to make an optimal decision, with some probability, using an adaptive finite number of observation episodes laces.

2.1 Partially Observable Markov Decision Process

Purely for clarity of the exposition, we focus, in this chapter, on the **static candidate action sequences** $\mathcal{A}_k = \{a_{k:k+L-1}\}$ instead of **policies**. Further, we separately redefine POMDP and corresponding policies in each paper. The POMDP is a tuple

$$\langle \mathcal{X}, \mathcal{A}_k, \mathcal{Z}, \mathrm{T}, \mathrm{O}, \rho, \gamma, b_0 \rangle \tag{2.1}$$

where $x \in \mathcal{X}$, $z \in \mathcal{Z}$ and the individual state and observation and $\mathcal{X}$ and $\mathcal{Z}$ are state and observation spaces. The $\gamma \in (0, 1]$ is a discount factor and b_0 is a prior belief. The state evolves according to the transition model

$$\mathbb{P}_{\mathrm{T}}(x'|x, a) \tag{2.2}$$

that models outcome uncertainty. The state is accessible solely through observations using the observation model

$$\mathbb{P}_{\mathrm{O}}(z|x) \tag{2.3}$$

that models the state uncertainty. The $\mathcal{A}_k$ is the space of candidate action sequences obtained by an external process separately for each planning time index k. The agent has access and maintains the belief over the POMDP state

$$b_k(x_k) \triangleq \mathbb{P}(x_k|b_0, a_{0:k-1}, z_{1:k}). \tag{2.4}$$

The BSP objective is to find an action sequence $a_{k:k+L-1} \in \mathcal{A}_k$ that maximizes the value function

$$V(b_k, a_{k:k+L-1}) = \mathbb{E}\Big[\sum_{\ell=k}^{k+L-1} \gamma^{\ell-k} \rho(b_\ell, a_\ell, b_{\ell+1}) \Big| b_k, a_{k:k+L-1} \Big]. \tag{2.5}$$

In this research we assume that ρ is a general belief dependent reward.

2.2 Confidence Intervals

Suppose the reward is bounded $\rho^{\min} \leq \rho \leq \rho^{\max}$ and the maximal horizon is $L^{\max}$. Denote $V_{\max} = \rho^{\max} L^{\max}$ and denote $a^*_{k+} = \arg\max_{a_{k+} \in \mathcal{A}} V(b_k, a_{k+})$ where V is the theoretical Value function. Let also be $a^\dagger_{k+} = \arg\max_{a_{k+} \in \mathcal{A}} \hat{V}^{(m)}(b_k, a_{k+})$.

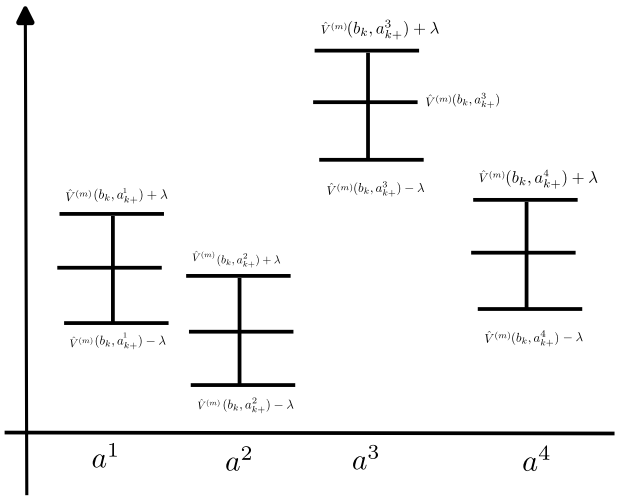


Figure 2.1: Simple application of confidence bounds. These bounds hold with some probability and the intervals themselves are stochastic.

Lemma 1 (Chernoff bound)**.** *For any given belief b_k and an action sequence a_{k+} with probability of at least $1 - e^{-\frac{\lambda^2 m}{V_{\max}^2}}$ it holds that*

$$|V(b_k, a_{k+}) - \hat{V}^{(m)}(b_k, a_{k+})| \leq \lambda \tag{2.6}$$

where $\hat{V}^{(m)}(b_k, a_{k+}) = \frac{1}{m}\sum_{i=1}^{m}\sum_{\ell=k}^{k+L-1}\rho(b^i_\ell, a_\ell, b^i_{\ell+1})$ and the probability is taken over the draw of the $\rho_{k:k+L}$ from $\mathbb{P}(\rho_{k:k+L}|b_k, a_{k+})$.

The proof is immediate using Chernoff bound and Hoeffding Lemma. Now we want to bound $|V(b_k, a^*_{k+}) - \hat{V}^{(m)}(b_k, a^\dagger_{k+})|$.

2.3 Probability Construction

Defining probability on a product space from $\mathcal{A}$ when each product is induced by the candidate action sequence, we obtain the new outcomes space $(\Omega_i, \mathcal{F}_i)_{i=1}^{|\mathcal{A}|}$ with an outcome being $\omega_1\omega_2\ldots\omega_{|\mathcal{A}|} \in \times_{i=1}^{|\mathcal{A}|}\Omega_i$ and σ-algebra $\otimes_{i=1}^{|\mathcal{A}|}\mathcal{F}_i$. The set $N \in \otimes_{i=1}^{|\mathcal{A}|}\mathcal{F}_i$ is defined as $N \triangleq \cap_{i=1}B_i$ with $B_i \in \mathcal{F}_i$. The PDF over this space is $\mathbb{P}((\rho^{a_{k+}}_{k:k+L})_{a_{k+}\in\mathcal{A}|}|b_k, \mathcal{A}) = \prod_{a_{k+}\in\mathcal{A}}\mathbb{P}(\rho^{a_{k+}}_{k:k+L}|b_k, a_{k+})$.

2.4 Probabilistic Intervals

Using the previously seen lemma we have that

$$\mathrm{P}\Big(\bigcap_{a_{k+}\in\mathcal{A}}\Big\{|V(b_k,a_{k+})-\hat{V}^{(m)}(b_k,a_{k+})|\leq\lambda\Big\}\Big|b_k,\mathcal{A}\Big)=\Bigg(\prod_{a_{k+}\in\mathcal{A}}\mathrm{P}\Big(|V(b_k,a_{k+})-\hat{V}^{(m)}(b_k,a_{k+})|\leq\lambda\Big|b_k,a_{k+}\Big)\Bigg)\geq(1-e^{-\frac{\lambda^2 m}{V_{\max}^2}})^{|\mathcal{A}|}. \tag{2.7}$$

We use

$$\hat{V}^{(m)}(b_k, a_{k+}) - \lambda \leq V(b_k, a_{k+}) \leq \hat{V}^{(m)}(b_k, a_{k+}) + \lambda \quad \forall a_{k+} \in \mathcal{A} \tag{2.8}$$

$$\max_{a_{k+}\in\mathcal{A}}\hat{V}^{(m)}(b_k, a_{k+}) - \lambda \leq \max_{a_{k+}\in\mathcal{A}}V(b_k, a_{k+}) \leq \max_{a_{k+}\in\mathcal{A}}\hat{V}^{(m)}(b_k, a_{k+}) + \lambda \tag{2.9}$$

with probability of at least $(1-e^{-\frac{\lambda^2 m}{V_{\max}^2}})^{|\mathcal{A}|}$. Note that here we know where the true optimal value can be using the estimated value, but we do not know if the same action sequence was selected. It will happen if no overlap is present, namely, we denote $a^\dagger = \arg\max_{a_{k+}\in\mathcal{A}}\hat{V}^{(m)}(b_k, a_{k+})$. If $\hat{V}^{(m)}(b_k, a^\dagger_{k+}) - \lambda \leq \max_{a_{k+}\in\mathcal{A}\setminus a^\dagger}\hat{V}^{(m)}(b_k, a_{k+}) + \lambda$. However, we do not know with which probability it will happen. This is because the intervals themselves are stochastic due to the dependence on the estimated value (Fig. 2.1). We need deterministic bounds over the objective. This brings us to this research beginning in the next section.

(a) **(b)**

Figure 2.2: These are analytical bounds. The intervals are deterministic. **(a)** The simplification loss is the overlap ; **(b)** No simplification impact in this case.

2.5 Deterministic Intervals and the Overlap

Having established the necessity for deterministic bounds we outline our approach. We will try to bound the objective deterministically as such

$$\underline{V}(b_k, a_{k+}) \leq V(b_k, a_{k+}) \leq \overline{V}(b_k, a_{k+}) \tag{2.10}$$

As we see in Fig. 2.2a the simplification loss is the overlap. On the contrary, in Fig. 2.2b no simplification impact is present due to absence of the overlap.

Chapter 3

Findings

Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)
Journal Track

Simplified Risk-aware Decision Making with Belief-dependent Rewards in Partially Observable Domains (Extended Abstract)*

Andrey Zhitnikov[1] , **Vadim Indelman**[2]
[1]Technion Autonomous Systems Program (TASP),
Technion - Israel Institute of Technology, Haifa 32000, Israel
[2]Department of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel
andreyz@campus.technion.ac.il, vadim.indelman@technion.ac.il

Abstract

It is a long-standing objective to ease the computation burden incurred by the decision-making problem under partial observability. Identifying the sensitivity to simplification of various components of the original problem has tremendous ramifications. Yet, algorithms for decision-making under uncertainty usually lean on approximations or heuristics without quantifying their effect. Therefore, challenging scenarios could severely impair the performance of such methods. In this paper, we extend the decision-making mechanism to the whole by removing standard approximations and considering all previously suppressed stochastic sources of variability. On top of this extension, we scrutinize the distribution of the return. We begin from a return given a single candidate policy and continue to the pair of returns given a corresponding pair of candidate policies. Furthermore, we present novel stochastic bounds on the return and novel tools, Probabilistic Loss (`PLoss`) and its online accessible counterpart (`PbLoss`), to characterize the effect of a simplification.

1 Introduction

While operating in a partially observable setting, the robot repetitively performs actions and receives observations from the environment in an interleaving manner. The result of each action is a imprecise change in the robot's state. The robot has access to the probability density of the state, given the history of its actions and the observations alongside the prior. We call this probability density a belief. In each planning session, the robot shall reason about future beliefs and select an optimal action based on its current belief using belief-dependent rewards and the objective operator. The robot shall look into the future as far as possible. With the growing horizon, however, the computational burden is becoming unbearable for the robot due to exponential growth in complexity [Papadimitriou and Tsitsiklis, 1987]. Many research efforts in Artificial Intelligence (AI) and Robotics communities have tackled the described problem. In AI community, it received the name Partially Observable Markov Decision Process (POMDP), whereas, in the Robotics community, it is known as Belief Space Planning (BSP). In classical POMDP the belief-dependent reward is assumed to be the average of the state-dependent reward with respect to belief. While alleviating the solution, this assumption hinders the ability to actively decrease uncertainty over the belief using general belief-dependent operators. In BSP, general belief-dependent rewards are essential, e.g., navigation, sensor placement problems. The classical assumption in BSP is that the belief follows Gaussian distribution [Indelman *et al.*, 2015].

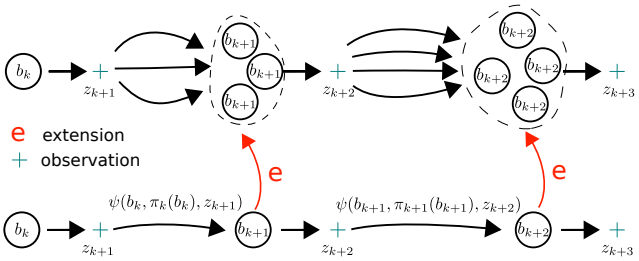


Figure 1: The Extended Belief Tree versus the standard.

The AI community began to introduce general belief-dependent rewards starting from the discrete domains [Araya *et al.*, 2010], [Fehr *et al.*, 2018], and limiting assumptions concerning the reward operators [Dressel and Kochenderfer, 2017]. More recent approaches such as Sparse Sampling (SS) [Kearns *et al.*, 2002], and Monte Carlo Tree Search (MCTS) [Sunberg and Kochenderfer, 2018] build upon Belief-MDP (BMDP). These methods are suitable for continuous domains. Still, in the continuous setting of states and observations, these methods give an approximate solution with only asymptotical optimality guarantees. On the other hand, the BSP community introduced a concept of *simplification* [Indelman, 2016],[Elimelech and Indelman, 2022], [Shienman and Indelman, 2022b], [Kitanov and Indelman, 2019]. As opposed to approximations, the simplification paradigm substitutes various parts of the decision-making problem while providing guarantees on the impact of such a substitution.

*The original journal paper: A. Zhitnikov and V. Indelman. Simplified Risk Aware Decision Making with Belief-dependent Rewards in Partially Observable Domains. Artificial Intelligence, Special Issue on "Risk-Aware Autonomous Systems: Theory and Practice", 2022.

In this work, we focus on the distribution of the rewards in a nonparametric setting. Our objective is to simplify the decision-making problem and analyze the impact of the simplification.

2 Notations and Problem Formulation

Let $\mathbb{P}$ be the probability density and P the probability. In this paper, we focus on the finite horizon setting. Further, to shorten notations, we shall often use $\square_{k+}$ to denote $\square_{k+1:k+L}$, where L is the planning horizon. By $\equiv$ we denote identity.

2.1 POMDP with Belief Dependent Rewards

ρ-POMDP [Araya *et al.*, 2010] is an eight tuple

$$\langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, \rho, \gamma, b_0 \rangle, \tag{1}$$

where $\mathcal{X}, \mathcal{A}, \mathcal{Z}$ are state, action, and observation spaces with $x \in \mathcal{X}, a \in \mathcal{A}, z \in \mathcal{Z}$ the momentary state, action, and observation, respectively, $T(x, a, x') \triangleq \mathbb{P}_T(x'|x, a)$ is the transition model from the past momentary state x to the next x' through action a, $O(z, x) \triangleq \mathbb{P}_Z(z|x)$ is the observation model, $\rho(b', z', a, b)$ is a scalar reward operator, $\gamma \in (0, 1]$ is the discount factor, and b_0 is the prior belief.

2.2 Belief Space Planning

The posterior belief at time instant k is given by

$$b_k(x_k) \approx \mathbb{P}(x_k|b_0, a_{0:k-1}, z_{1:k}). \tag{2}$$

The usual assumption is that the belief is a sufficient statistic for decision making objective [Bertsekas, 1995]. However, in practice, the belief requires some representation. This representation is not perfect, e.g., parametric or sampled form; thus, in (2), we used the $\approx$ sign. In a real life scenario $b_k = \psi(\psi(\dots\psi(b_0, a_0, z_1), a_{k-2}, z_{k-1}), a_{k-1}, z_k)$, where ψ is a method for updating the belief. By $\pi \triangleq \pi_{k:k+L-1}$ we denote a vector of policies for L time steps starting from time step k. Each such policy π_ℓ at time step ℓ maps belief to an action $\pi_\ell(b_\ell) = a_\ell$. The general decision making under uncertainty objective function is of the following form

$$V^L(b_k, \pi) = \varphi\big(\mathbb{P}(\rho_{k+1:k+L}|b_k, \pi_{k:k+L-1}), g_k\big) \tag{3}$$
$$\text{s.t. } b_\ell = \psi(b_{\ell-1}, \pi_{\ell-1}(b_{\ell-1}), z_\ell),$$

where L is the planning horizon, ρ_ℓ is a random immediate reward, φ is an objective operator, and $g_k \triangleq f_{g_k}(\rho_{k+1:k+L})$ is the return [Sutton and Barto, 2018]. A common choice for φ is expectation over the distribution of future rewards given all data available [Defourny *et al.*, 2008]. The return is a deterministic known function of the realization of $\rho_{k+1:k+L}$, e.g., it could correspond to the cumulative reward $g_k = \sum_{\ell=1}^{L} \rho_{k+\ell}$. Finally, ψ is a general method for propagating the belief with action and updating it with the received observation.

The objective (3) is ultimately based on the *distribution of the return* given all information available for planning under selected policy $\mathbb{P}(g_k|b_k, \pi_k)$, which decomposes via marginalization over future observations $z_{k+} \equiv z_{k+1:k+L}$ as

$$\mathbb{P}(g_k|b_k, \pi) = \int_{z_{k+}} \mathbb{P}(g_k|b_k, \pi, z_{k+}) \mathbb{P}(z_{k+}|b_k, \pi) \mathrm{d}z_{k+}. \tag{4}$$

A common assumption is that $\mathbb{P}(g_k|b_k, \pi, z_{k+},)$ is a Dirac delta function.

3 Foundations

In this section we introduce probabilistic ρ-POMDP and rigorously define the *simplification* paradigm. We further continue to the formulation of the general bounds on the reward/return which can be analytical or stochastic.

3.1 Extended Setting, Probabilistic ρ-POMDP

Sometimes the belief $b_{\ell-1}$ has a simple parametric form, where $\theta_{\ell-1}$ is a vector of parameters, e.g., a Gaussian belief. In this case, belief update ψ can be deterministic, and is denoted by $\psi_{\mathrm{dt}}(\theta_{\ell-1}, \pi_{\ell-1}(\theta_{\ell-1}), z_\ell)$. In more general and challenging scenarios the belief $b_{\ell-1}$ is given by a set of weighted samples $\{(w^i_{\ell-1}, x^i_{\ell-1})\}_{i=1}^N$. Therefore, ψ is a stochastic method, e.g., a particle filter [Thrun *et al.*, 2005]. Applying multiple times ψ on the same input will yield different sets of samples approximating the same distribution of the posterior belief. We denote the stochastic ψ by $\psi_{\mathrm{st}}(b_{\ell-1}, \pi_{\ell-1}(b_{\ell-1}), z_\ell)$. Another form to formulate the above is that the distribution

$$B(b_{\ell-1}, a_{\ell-1}, z_\ell, b_\ell) \triangleq \mathbb{P}_B(b_\ell|b_{\ell-1}, a_{\ell-1}, z_\ell), \tag{5}$$

is not a Dirac delta function. This aspect was disregarded so far, to the best of our knowledge. Note that in a Belief MDP (BMDP) formulation, the assumption is that B is a Dirac delta function. Similar arguments hold for the momentary reward operator of the belief. We extend $\rho(b', z', a, b)$ to

$$R(b_{\ell-1}, a_{\ell-1}, z_\ell, b_\ell, \rho_\ell) \triangleq \mathbb{P}_R(\rho_\ell|b_\ell, z_\ell, a_{\ell-1}, b_{\ell-1}), \tag{6}$$

To our knowledge, we are the first who treat these aspects as random.

Before introducing simplification formally and analyzing its impact, we shall account for all potential sources of variability. We remove conventional approximations by extending (1) to a probabilistic reward model R (6) and probabilistic belief update B (5), and introduce

$$M = \langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, R, \gamma, b_k, B \rangle, \tag{7}$$

which we name probabilistic ρ-POMDP ($\mathbb{P}\rho$-POMDP). The rationale behind these conditional distributions (R and B) is to capture additional sources of stochasticity, such as stochastic belief update, stochastic calculation of a given reward operator or simply not knowing the operator reward in an explicit analytic form.

As discussed earlier, the value function (3) is based on (4). These previously overlooked sources of stochasticity impact the likelihood of the observations

$$\mathbb{P}(z_{k+1:k+L}|b_k, \pi), \tag{8}$$

as well as the joint reward distribution $\mathbb{P}(\rho_{k+}|b_k, \pi, z_{k+}) \equiv \mathbb{P}(\rho_{k+1:k+L}|b_k, \pi_{k:k+L-1}, z_{k+1:k+L})$ given a realization of future observations. In contrast, in the regular setting of POMDP and ρ-POMDP $\mathbb{P}(\rho_{k+}|b_k, \pi, z_{k+})$ is Dirac's delta function. If B is a Dirac function, a sample from (8) uniquely defines the corresponding posterior beliefs $b_{k+1:k+L}$. This, therefore, corresponds to the classical belief tree (R could still be non a Dirac function). In contrast, our $\mathbb{P}\rho$-POMDP (7), corresponds to an *extended* belief tree, which, due to (5), allows many samples of the beliefs $b_{k+1:k+L}$ for each sample of $z_{k+1:k+L}$ from (8) (See Fig. 1).

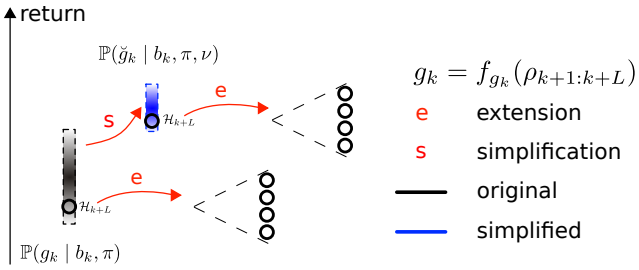


Figure 2: Our extension and the simplification in the context of a single candidate policy.

3.2 Simplification Formulation

To formally define the simplification procedure, we augment the $\mathbb{P}\rho$-POMDP tuple (7) with a simplification operator $\nu \triangleq \nu_k, \ldots, \nu_{k+L}$,

$$M_\nu = \langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, R, \gamma, b_k, B, \nu \rangle. \tag{9}$$

This general operator defines any possible modification of the original problem defined by (7) alongside with (3) to a new, simpler to solve, problem. The definition (9) allows us to retain the connection to the original nonsimplified problem (7) and examine the impact of the simplification on (7). The operator ν can be for example, sparsification of the initial belief b_k [Elimelech and Indelman, 2022], replacing the reward by its topological signature [Kitanov and Indelman, 2019], direct calculation of lightweight reward bounds [Sztyglic and Indelman, 2022], selecting a subset of hypotheses in a hybrid or mixture belief [Shienman and Indelman, 2022a], to name a few.

Generally, M and M_ν are different decision making problems. We shall be interested in working online with the latter while providing the guarantees with respect to the former. To distinguish a simplified reward from the original reward, we denote the former by $\breve{\rho}$ instead of ρ; similarly, we denote the simplified belief by $\breve{b}$ instead of b. Note the operator ν can be stochastic, as discussed below. Specifically, belief simplification is described by the distribution

$$\mathbb{P}(\breve{b}_\ell | b_\ell; \nu^b_\ell). \tag{10}$$

In general, the distribution (10) over the simplified belief $\breve{b}_\ell$ corresponds to a stochastic simplification operator ν^b_ℓ. This is the case, for example, when b_ℓ is represented by a set of N weighted samples and ν^b_ℓ is the operation of subsampling n samples according to weights; i.e., applying this operation on b_ℓ multiple times leads to different sets of n samples, each representing another realization of $\breve{b}_\ell$ from (10). Overall there are $\binom{N}{n}$ such combinations. For a deterministic operator ν^b_ℓ, (10) is a Dirac function.

Further, there are several cases of how a simplification affects belief update (5) from time $\ell - 1$ to ℓ.

1. Without any simplification we have $\mathbb{P}_B(b_\ell | b_{\ell-1}, \pi_{\ell-1}, z_\ell)$ from (5).
2. Given a simplified belief $\breve{b}_{\ell-1}$, while keeping the original stochastic belief update ψ_{st}, we have $\mathbb{P}_B(\breve{b}_\ell | \breve{b}_{\ell-1}, \pi_{\ell-1}, z_\ell)$, where each realization of $\breve{b}_\ell$ is obtained via ψ_{st}. Thus, given $\breve{b}_{\ell-1}$, this distribution is not a function of ν.
3. We can also simplify the belief update operator, ψ_{st}, to $\breve{\psi}_{\text{st}}$. Denoting the corresponding simplification operator ν^ψ_ℓ, this yields $\mathbb{P}_{\breve{B}}(\breve{b}_\ell | \breve{b}_{\ell-1}, \pi_{\ell-1}, z_\ell; \nu^\psi_\ell)$.
4. Finally, one can decide at time ℓ to apply simplification on the belief (determined by ν^b_ℓ) via (10). The corresponding belief update can be written as

$$\mathbb{P}_{\breve{B}}(\breve{b}_\ell | \breve{b}_{\ell-1}, \pi_{\ell-1}, z_\ell; \nu^b_\ell, \nu^\psi_\ell) = \int_{\tilde{b}_\ell} \mathbb{P}(\breve{b}_\ell | \tilde{b}_\ell; \nu^b_\ell) \mathbb{P}_{\breve{B}}(\tilde{b}_\ell | \breve{b}_{\ell-1}, \pi_{\ell-1}, z_\ell; \nu^\psi_\ell) \mathrm{d}\tilde{b}_\ell,$$

where $\tilde{b}_\ell$ is the integration variable.

We combine these cases and write

$$\breve{B}(\breve{b}_{\ell-1}, \pi_{\ell-1}, z_\ell, \breve{b}_\ell; \nu) \triangleq \mathbb{P}_{\breve{B}}(\breve{b}_\ell | \breve{b}_{\ell-1}, \pi_{\ell-1}, z_\ell; \nu^b_\ell, \nu^\psi_\ell). \tag{11}$$

Similarly, reward simplification could be, in general, stochastic, leading to the distribution

$$\mathbb{P}(\breve{\rho}_\ell | \rho_\ell; \nu^\rho_\ell). \tag{12}$$

Thus, given a simplified belief $\breve{b}_\ell$ and $\breve{b}_{\ell-1}$, and recalling (6), the distribution over $\breve{\rho}_\ell$ is

$$\mathbb{P}_{\breve{R}}(\breve{\rho}_\ell | \breve{b}_\ell, z_\ell, \pi_{\ell-1}(\breve{b}_{\ell-1}), \breve{b}_{\ell-1}; \nu) = \int_{\tilde{\rho}_\ell} \mathbb{P}(\breve{\rho}_\ell | \tilde{\rho}_\ell; \nu^\rho_\ell) \mathbb{P}_R(\tilde{\rho}_\ell | \breve{b}_\ell, z_\ell, \pi_{\ell-1}(\breve{b}_{\ell-1}), \breve{b}_{\ell-1}) \mathrm{d}\tilde{\rho}_\ell,$$

which we denote as the simplified reward model,

$$\breve{R}(\breve{b}_\ell, z_\ell, \pi_{\ell-1}(\breve{b}_{\ell-1}), \breve{\rho}_\ell; \nu) \triangleq \mathbb{P}_{\breve{R}}(\breve{\rho}_\ell | \breve{b}_\ell, z_\ell, \pi_{\ell-1}(\breve{b}_{\ell-1}), \breve{b}_{\ell-1}; \nu). \tag{13}$$

Throughout the document we assume that operator ν does not affect the observations likelihood. In other words, the measurements are sampled as in the original problem as in (8). For the further discussion we make the following shorthand notation. Let $\mathcal{H}_{k+L} \triangleq \{b_k, \pi, z_{k+}\}$ be future history at the time index $k + L$.

3.3 Online Stochastic and Analytical Bounds

We turn now to the joint distribution over original and simplified rewards, given the future history and operator ν, namely $\mathbb{P}(\rho_{k+}, \breve{\rho}_{k+} | \mathcal{H}_{k+L}, \nu)$. In an online setting we do not have access to the original rewards as calculating them explicitly defeats the purpose of simplification. Instead, we shall now utilize simplification to provide bounds over the original rewards. These bounds can be used to provide performance guarantees, and should be cheaper to calculate than the original unsimplified rewards. Further, the bounds can be analytical as in previous simplification approaches, e.g, [Elimelech and Indelman, 2022]. Ultimately for each realization of the return we are interested in the following relation

$$l \leq g_k \leq u. \tag{14}$$

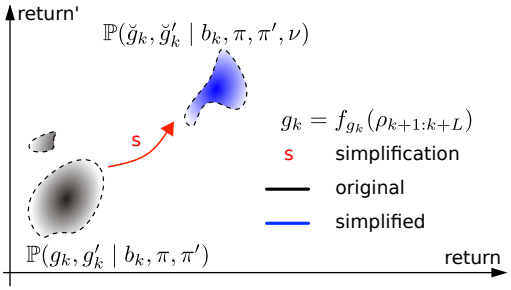


Figure 3: The simplification in our extended setting and its impact of the joint distribution of a pair of the returns corresponding to the pair of the candidate policies.

One way to do that is to develop analytical bounds, which will hold for any possible observation $z_{k+1:k+L}$ received and any corresponding return, e.g, as in [Sztyglic and Indelman, 2021].

Our extension allows R and B, as well as $\breve{R}$ and $\breve{B}$ to be any distributions. They can remain Dirac functions, e.g., if belief update and the reward calculation have a closed form. Successively, $\mathbb{P}(g_k|b_k, \pi, z_{k+},)$ remains Dirac delta. However, in the more general case, following our extension, there is a joint distribution of original and simplified returns given a realization of the future and the present

$$\mathbb{P}(g_k, \breve{g}_k|\mathcal{H}_{k+L}, \nu), \tag{15}$$

as illustrated in Fig. 2. Given the history $\mathcal{H}_{k+L}$, the return g_k as well as the simplified return $\breve{g}_k$ has variability, in contrast to the conventional approach. Ordinarily, the belief update is commenced once and treated as deterministic. So as the rewards and return do not have variance given the history of the actions and the observations. Since (15) is no longer a Dirac function, we can use knowledge about this distribution to design bounds, which will hold with *some* probability. In the main paper [Zhitnikov and Indelman, 2022], we show that it is possible to harness the structure of (15) to design the mentioned more lenient online bounds. Moreover, analytical bounds, designed in a conventional setting, can be used in our extended setting without any revision. In our extended setting, they will bound with probability one.

Having introduced the novel stochastic bounds, we proceed to the formulation of the constraints, that these bounds shall fulfill to be meaningful. Let the parameter controlling the confidence level be $\alpha \in [0, 1)$. For every possible sample $\breve{g}_k$ we do not know which sample g_k one could obtain in the original problem. However, if the bounds are designed such that $\mathbb{P}(g_k, l, u|\mathcal{H}_{k+L}, \nu)$ render

$$1 - \alpha \leq \mathrm{P}\left(\mathbf{1}\{l \leq g_k \leq u\} = 1|\mathcal{H}_{k+L}, \nu\right) \tag{16}$$

these bounds can be useful. Notably, the above equation does not involve simplified return, so is applicable also in the case bounds are directly formulated (and not via a simplified return). However, in this case the bounds are analytical and $\alpha = 0$. To summarize, there are three types of online reward/return bounds:

1. Deterministic bounds. These analytical bounds exist in case of a closed form belief update ψ_{dt} and a deterministic operator reward, e.g., belief is a Gaussian and the reward is differential entropy. In this case, even in our extended setting R and B remain Dirac functions.
2. Stochastic bounds that hold with probability one, namely $\alpha = 0$. These are also analytical bounds. In our extended setting R and B are no longer Dirac functions. However, these bounds hold for any realization of sample approximation, as stated around (14).
3. Stochastic bounds that hold at least with probability $1 - \alpha$. They exist only in our extended setting when R and B are not Dirac functions.

4 The Return Given a Candidate Policy

Applying the marginalization over the observations we obtain the distribution of the original and the simplified return given the candidate policy and the operator ν (See Fig. 2).

$$\mathbb{P}(g_k, \breve{g}_k|b_k, \pi, \nu) = \int_{z_{k+}} \mathbb{P}(g_k, \breve{g}_k|\mathcal{H}_{k+L}, \nu)\mathbb{P}(z_{k+}|b_k, \pi)\mathrm{d}z_{k+}.$$

For further discussion please see [Zhitnikov and Indelman, 2022].

5 The Pair of the Returns Corresponding to the Pair of Candidate Policies

Imagine a pair of a candidate policies. In such a setting we are interested in the following distribution (See Fig. 3)

$$\mathbb{P}(g_k, g'_k, \breve{g}_k, \breve{g}'_k|b_k, \pi, \pi', \nu). \tag{17}$$

On top of (17) we propose a tool to examine the simplification impact on the original not simplified problem. We call it Probabilistic Loss.

5.1 Probabilistic Loss (`PLoss`)

Consider a random variable $\mathcal{L} : \Omega \to \mathbb{R}$ over the events space Ω defined as such

$$\mathcal{L}(\omega) \triangleq \begin{cases} \max\{g'_k(\omega) - g_k(\omega), 0\} & \text{if } \breve{g}_k(\omega) > \breve{g}'_k(\omega) \\ \max\{g_k(\omega) - g'_k(\omega), 0\} & \text{if } \breve{g}_k(\omega) < \breve{g}'_k(\omega) \\ 0 & \text{if } \breve{g}_k(\omega) = \breve{g}'_k(\omega) \end{cases} \tag{18}$$

The realization of random variable $\mathcal{L}(\omega) = \Delta$ differs from zero if the simplification have switched the ordering of the original returns and the original difference between returns was Δ.

5.2 Online Bound on Probabilistic Loss (`PbLoss`)

Since the `PLoss` is inaccessible online we propose another random variable which is accessible.

$$\bar{\mathcal{L}}(\omega) \triangleq \begin{cases} \max\{u'(\omega) - l(\omega), 0\} & \text{if } \breve{g}_k(\omega) > \breve{g}'_k(\omega) \\ \max\{u(\omega) - l'(\omega), 0\} & \text{if } \breve{g}_k(\omega) < \breve{g}'_k(\omega) \\ 0 & \text{if } \breve{g}_k(\omega) = \breve{g}'_k(\omega) \end{cases} \tag{19}$$

To give to the reader a glimpse into the connection between `PLoss` and `PbLoss` suppose the bounds (14) are analytical. This implies that $\mathcal{L}(\omega) \leq \bar{\mathcal{L}}(\omega) \quad \forall \omega \in \Omega$ and this implies

$$\mathrm{P}(\Delta \leq \mathcal{L}(\omega)) \leq \mathrm{P}(\Delta \leq \bar{\mathcal{L}}(\omega)) \tag{20}$$

To the impact of the proposed ideas onto Decision Making please refer to the journal paper [Zhitnikov and Indelman, 2022].

References

[Araya *et al.*, 2010] Mauricio Araya, Olivier Buffet, Vincent Thomas, and Françcois Charpillet. A pomdp extension with belief-dependent rewards. In *Advances in Neural Information Processing Systems (NIPS)*, pages 64–72, 2010.

[Bertsekas, 1995] Dimitri Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

[Defourny *et al.*, 2008] Boris Defourny, Damien Ernst, and Louis Wehenkel. Risk-aware decision making and dynamic programming. In *NIPS Workshop on Model Uncertainty and Risk in RL*, 2008.

[Dressel and Kochenderfer, 2017] Louis Dressel and Mykel J. Kochenderfer. Efficient decision-theoretic target localization. In Laura Barbulescu, Jeremy Frank, Mausam, and Stephen F. Smith, editors, *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS 2017, Pittsburgh, Pennsylvania, USA, June 18-23, 2017*, pages 70–78. AAAI Press, 2017.

[Elimelech and Indelman, 2022] Khen Elimelech and Vadim Indelman. Simplified decision making in the belief space using belief sparsification. *The International Journal of Robotics Research*, 41(5):470–496, 2022.

[Fehr *et al.*, 2018] Mathieu Fehr, Olivier Buffet, Vincent Thomas, and Jilles Dibangoye. rho-pomdps have lipschitz-continuous epsilon-optimal value functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6933–6943. Curran Associates, Inc., 2018.

[Indelman *et al.*, 2015] V. Indelman, L. Carlone, and F. Dellaert. Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments. *Intl. J. of Robotics Research*, 34(7):849–882, 2015.

[Indelman, 2016] V. Indelman. No correlations involved: Decision making under uncertainty in a conservative sparse information space. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):407–414, 2016.

[Kearns *et al.*, 2002] Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine learning*, 49(2):193–208, 2002.

[Kitanov and Indelman, 2019] A. Kitanov and V. Indelman. Topological information-theoretic belief space planning with optimality guarantees. *arXiv preprint arXiv:1903.00927*, 3 2019.

[Papadimitriou and Tsitsiklis, 1987] C. Papadimitriou and J. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

[Shienman and Indelman, 2022a] M. Shienman and V. Indelman. D2a-bsp: Distilled data association belief space planning with performance guarantees under budget constraints. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2022.

[Shienman and Indelman, 2022b] M. Shienman and V. Indelman. Nonmyopic distilled data association belief space planning under budget constraints. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2022.

[Sunberg and Kochenderfer, 2018] Zachary Sunberg and Mykel Kochenderfer. Online algorithms for pomdps with continuous state, action, and observation spaces. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28, 2018.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Sztyglic and Indelman, 2021] Ori Sztyglic and Vadim Indelman. Online pomdp planning via simplification. *arXiv preprint arXiv:2105.05296*, 2021.

[Sztyglic and Indelman, 2022] Ori Sztyglic and Vadim Indelman. Speeding up online pomdp planning via simplification. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT press, Cambridge, MA, 2005.

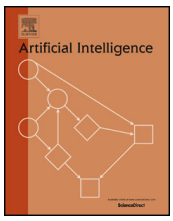
[Zhitnikov and Indelman, 2022] A. Zhitnikov and V. Indelman. Simplified risk aware decision making with belief dependent rewards in partially observable domains. *Artificial Intelligence, Special Issue on "Risk-Aware Autonomous Systems: Theory and Practice"*, 2022.

Artificial Intelligence 312 (2022) 103775

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint



Simplified Risk-aware Decision Making with Belief-dependent Rewards in Partially Observable Domains ☆,☆☆

Andrey Zhitnikov [a,*], Vadim Indelman [b]

[a] Technion Autonomous Systems Program (TASP), Haifa, 3200003, Israel
[b] Department of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa, 32000, Israel

ARTICLE INFO

Article history:
Received 14 October 2021
Received in revised form 21 August 2022
Accepted 21 August 2022
Available online 27 August 2022

Keywords:
Artificial intelligence
Decision making under uncertainty
Belief space planning
POMDP

ABSTRACT

With the recent advent of risk awareness, decision-making algorithms' complexity increases, posing a severe difficulty to solve such formulations of the problem online. Our approach is centered on the distribution of the return in the challenging continuous domain under partial observability. This paper proposes a simplification framework to ease the computational burden while providing guarantees on the simplification impact. On top of this framework, we present novel stochastic bounds on the return that apply to any reward function. Further, we consider simplification's impact on decision making with risk averse objectives, which, to the best of our knowledge, has not been investigated thus far. In particular, we prove that stochastic bounds on the return yield deterministic bounds on Value at Risk. The second part of the paper focuses on the joint distribution of a pair of returns given a pair of candidate policies, thereby, for the first time, accounting for the correlation between these returns. Here, we propose a novel risk averse objective and apply our simplification paradigm. Moreover, we present a novel tool called the probabilistic loss (`PLoss`) to completely characterize the simplification impact for *any* objective operator in this setting. We provably bound the cumulative and tail distribution function of `PLoss` using `PbLoss` to provide such a characterization online using only the simplified problem. In addition, we utilize this tool to offer deterministic guarantees to the simplification in the context of our novel risk averse objective. We employ our proposed framework on a particular simplification technique - reducing the number of samples for reward calculation or belief representation within planning. Finally, we verify the advantages of our approach through extensive simulations.

1. Introduction

Autonomous online decision-making is a fundamental aspect of intelligence. In a partially observable setting, which is common in real world scenarios, there is no direct access to the state. Instead, the robot has to maintain a belief over the state and reason about its evolution while accounting for different sources of uncertainty within the decision-making stage. The renowned framework to do so is the Partially Observable Markov Decision Process (POMDP) [18]. Crucial elements

☆ This research was partially supported by the Israel Science Foundation (ISF), grant No. 371/20 and by a donation from the Zuckerman Fund to the Technion Center for Machine Learning and Intelligent Systems (MLIS).
☆☆ This paper is part of the Special Issue: "Risk-aware Autonomous Systems: Theory and Practice".
* Corresponding author.
E-mail addresses: andreyz@campus.technion.ac.il (A. Zhitnikov), vadim.indelman@technion.ac.il (V. Indelman).

https://doi.org/10.1016/j.artint.2022.103775
0004-3702/

defining the robot's behavior are the random reward and the objective operator applied to the reward distribution. The random nature of the reward arises from the uncertainties in the system.

Solving a POMDP, i.e., calculating the "right decision" in terms of an optimal action sequence or policy, involves anticipating every imaginable turn of future events and computing the *returns* based on the corresponding rewards. One typical example of the return is the future cumulative reward.

There is a large body of algorithms, formulated on top of POMDP, to approximate decision-making under uncertainty. Classical offline methods [21] are trying to find offline a policy that is optimal for all possible beliefs. These methods are based on α-vectors and point based value iteration [24,26,34]. Since the α-vector is the vector of the utility function values, starting from the state realizations or samples from the belief distribution, under the conditional plan, the set of α-vectors, each annotated with an action, can represent the policy for all beliefs. The application of such policy is to find α-vector maximizing the inner product with the belief. Unfortunately, these methods are suitable solely for the discrete state, action, and observation spaces. Some work on extending the α-vectors to continuous spaces has been done by [28]. More recent formulations suitable for continuous spaces are operating on the belief tree.

A necessary factor for planning to be successful is the number of future steps ahead (horizon) that an agent considers in the decision-making process. The belief tree grows exponentially with the horizon. However, the exponential growth with the horizon is not the only problem of the belief tree based approaches. Additionally, the number of possible states grows exponentially with the state space dimension, and consequently, an adequate representation of the belief requires more particles in the setting of non-parametric beliefs. Those last two problems are known as the *curse of history* and the *curse of dimensionality* respectively.

More recently, online methods became successful. Some of them are suitable for continuous state and observation spaces. The output of these methods is an action recommended for the current belief. The algorithm itself is a policy which maps from beliefs to actions online. Prominent examples are POMCP [33] and its various extensions (e.g., [37]), an algorithm designed for large POMDP and based on Monte Carlo tree search. Another popular algorithm, DESPOT [35] [44], focuses on the set of randomly sampled scenarios over the belief tree, avoiding drawbacks of the UCT [22] algorithm used in POMCP.

Standard POMDP formulations consider state-dependent rewards and assume that the belief-dependent reward is *nothing but expectation over the state reward*. POMDP with *belief-dependent rewards* received much less attention, although these rewards are essential in numerous problems, such as information gathering, autonomous navigation, and active sensing. Information theoretic rewards are especially significant for belief space planning (BSP) [17], [10]. Araya et al. [1] introduced ρ-POMDP and extended the exact α-vectors method and a family of point based approximation algorithms to consider convex belief-dependent reward functions. Later Fehr et al. [9] extended their work further to Lipschitz-continuous reward functions. Spaan et al. [36] proposed to augment action space with information-reward actions. Dressel and Kochenderfer [7] proposed an extension of SARSOP [24] to specific forms of belief-dependent rewards. However, these extensions are limited either to a discrete setting or to specific forms of belief dependent rewards. In a general setting, belief-dependent rewards are computationally demanding and prohibitively expensive.

Further, the most popular and widespread objective operator is the expected value of the return. However, the expected value as the objective has inherent flaws. It is oblivious to the distribution of the reward. Meaning it is unable to account for the risk that the selected action or policy is suboptimal and to prevent rare undesirable events. One way to introduce the notion of risk to decision making is to augment the expected return value with chance constraints. This augmentation, however, introduces additional complications which are out of the scope of this paper [31]. With this motivation in mind, we focus on an alternative to the expected value objectives in the context of BSP in continuous domains.

Replacing expected value by other objectives in the context of MDP has been discussed in [6]. Importantly, Defourny et al. [6] discuss risk measures and applicability of Bellman form. Attractive risk averse objectives include Value at Risk (VaR) and Conditional VaR (CVaR) [5]. VaR and CVaR were extensively studied in the context of MDP, whereas in the POMDP planning community, they started to emerge only recently [12,13]. So far, we did not find work considering belief-dependent rewards in the context of decision making under uncertainty with risk averse objectives.

The computational burden incurred by the complexity of POMDP planning inspired many research works to focus on approximations of the problem, e.g., [14]. Typically, approximation based planners show asymptotical guarantees, e.g., the convergence of the algorithms. We take a different path, which is to simplify the original decision-making problem. In other words, instead of approximating the problem, we substitute it with a simpler one. If the order of policies with respect to the original and simplified problems' objective is preserved, such substitution does not affect the decision-making quality. Moreover, suppose we can find online bounds over the original problems' returns/rewards or objective function, utilizing the simplified problem. In that case, it is possible to account for the simplification loss.

Replacement of various parts of the decision making problem to ease the computation burden while preserving the precedence of objectives for potential action plans recently appeared in the literature under the names *simplification* paradigm [39,41,8,16,19] [32,2], *action consistency* [8,19] and *tree consistency* [41]. Yet, these works have limitations. A common assumption is a specific objective operator - expectation. Moreover, Elimelech and Indelman [8], Kitanov and Indelman [19] assume Gaussian distributions and maximum likelihood observations while working in the highly challenging setting of a high dimensional state. Sztyglic et al. [41] consider non-parametric beliefs; however, they build upon a specific belief dependent reward operator.

The general simplification paradigm is concerned with carefully replacing the nonessential elements of the decision making problem and quantifying the impact of this relaxation. Specifically, simplification methods are accompanied by

stringent guarantees while alleviating the computational burden of the decision making problem. Therefore, previous works formulated the *simplification paradigm* on top of analytical bounds and a conventional expectation operator as the objective. Existing works consider a deterministic belief update; however this is problematic for non-parametric beliefs which cannot be updated in a deterministic way. In the setting of general beliefs, we shall resort to a particle filter [42] which is a stochastic belief update method since it is based on sample approximations.

1.1. Contributions

We study the simplification impact on decision making under uncertainty considering a general objective operator and non-parametric beliefs, which therefore involves the distribution over returns. This distribution conveys all the information about the decision making problem. Our overall goal is to examine how the simplification method influences the performance of the decision-maker while accelerating the decision making process.

To account for the impact of simplification on the distribution over returns, we first relax typical assumptions regarding the belief update along with the operator reward and introduce probabilistic ρ-POMDP, which we denote as $\mathbb{P}\rho$-POMDP. Given a simplification and an objective operator, we utilize bounds over the return to provide performance guarantees in terms of quality of solution with respect to the original (un-simplified) decision making problem. These bounds can be analytical, and thus hold with probability one. Crucially, we also introduce stochastic bounds that are applicable to any reward function, in contrast to analytical bounds that must be derived for each reward function separately.

Further, we consider specifically simplification impact on decision making with risk averse objectives. To the best of our knowledge, this is the first work that investigates simplification in this context. Our key result is the derivation of *deterministic* bounds on the risk averse objective (Value at Risk) using stochastic bounds on the return/reward. Consequently, we obtain solving speedup and provide guarantees.

Moreover, we examine how simplification impacts the joint distribution over the returns for two candidate policies. We believe this distribution conveys previously unaccounted information, as generally the returns for different policies, conditioned on the current belief, are coupled. To the best of our knowledge, this joint distribution has not been studied yet. Towards this end, we propose a novel risk aware objective operator on top of the joint distribution over returns for two candidate policies. This is as opposed to conventional objectives that are based on the marginal distribution of the return given a policy. Furthermore, we develop a method to provide guarantees for the simplification of such an objective. Specifically, we introduce probabilistic loss (`PLoss`) and the corresponding online bound on probabilistic loss (`PbLoss`) to completely characterize the simplification impact on the joint distribution of the rewards given two candidate policies, meaning for any objective operator. We then utilize the latter to provide performance guarantees in terms of deterministic bounds considering the mentioned risk aware objective operator.

Finally, we apply our general formulation considering a specific simplification: reducing the number of samples of the belief for the reward calculation. To be precise, in the setting of an explicitly given belief surface (e.g. Gaussian mixture model), we endow the stochastic bounds with an adaptivity property and show how to take the lowest possible number of samples while remaining action consistent. In the setting of general beliefs represented by particles we lower the number of particles of the belief and provide performance guarantees.

To summarize, our key contributions are as follows. (a) We extend ρ-POMDP to probabilistic ρ-POMDP ($\mathbb{P}\rho$-POMDP) by relaxing the assumption that the reward operator and the belief update are deterministic; (b) We introduce novel stochastic bounds on the return/reward and rigorously formulate the simplification framework on top of general objective operators and returns/rewards; (c) Using our formulations we present simplification of risk averse decision making under uncertainty; (d) We present a novel objective utilizing joint distribution of the rewards corresponding to two candidate policies and describe a method to simplify such decision making while preserving *action consistency*; (e) We introduce the general concept of `PLoss` and provide its online description with `PbLoss` and utilize it to provide guarantees in terms of deterministic bounds; (f) Finally, we exemplify our framework on a particular simplification technique, which is reducing the number of samples within planning.

1.2. Paper structure

This paper is organized as follows. In section 2 we introduce the notations and formulate the problem. In section 3 we provide mathematical foundations for our approach. We then focus on the marginal distribution of the return in section 4 and on the joint distribution of a pair of the returns corresponding to two candidate policies in section 5. We present a specific simplification in section 6. In section 7 we exemplify our findings on the problem of autonomous navigation with light beacons.

2. Notations and problem formulation

Let us denote by $\mathbb{P}$ the probability density function and by P the probability. By lowercase letter we denote a random vector or its realization. For two random variables x and y, we say that they are equal $x = y$ if they are equal as functions on their measurable space. Further, to shorten notations, we shall often use $\square_{k+}$ to denote $\square_{k+1:k+L}$, where L is the planning horizon. By $\equiv$ we denote identity. We summarize important notations used throughout the paper in Table 1.

Table 1
List of important notation.

Nomenclature	
$\mathcal{X}, \mathcal{A}, \mathcal{Z}$	State, Action, and Observation spaces
$x_k \in \mathcal{X}, a_k \in \mathcal{A}, z_k \in \mathcal{Z}$	Momentary state, action, and observation, respectively.
$\mathbf{1}\{\cdot\}$	Indicator function defined on set {}.
$a \wedge b$	$\min\{a, b\}$ where $a, b \in \mathbb{R}$.
$a \vee b$	$\max\{a, b\}$ where $a, b \in \mathbb{R}$.
ρ_{k+}, $\rho_{k+1:k+L}$	Reward vector from time index $k+1$ until $k+L$ including the ends.
$\check{\rho}_{k+}$, $\check{\rho}_{k+1:k+L}$	Simplified reward vector.
g_k	Return calculated from the reward vector, such that $g_k \triangleq f_{g_k}(\rho_{k+1:k+L})$ and f_{g_k} is some deterministic function.
$\check{g}_k$	Simplified return calculated from simplified reward vector.
z_{k+}, $z_{k+1:k+L}$	Observation vector from time index $k+1$ until $k+L$ including the ends.
ψ	A general method for updating the belief.
π, $\pi_{k:k+L-1}$	Policy sequence from time index k up until $k+L-1$ including the edges.
ν, $\nu_{k:k+L}$	The sequence of simplification operators from time index k up until $k+L$ including the edges.
$\mathcal{H}_{k+L} \triangleq \{b_k, \pi, z_{k+}\}$	The future history at the time index $k+L$.

2.1. POMDP with belief dependent rewards

Let k be an arbitrary time step. ρ-POMDP [1] is an eight tuple

$$\langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, \rho, \gamma, b_0 \rangle, \tag{1}$$

where $\mathcal{X}, \mathcal{A}, \mathcal{Z}$ are state, action, and observation spaces with $x_k \in \mathcal{X}, a_k \in \mathcal{A}, z_k \in \mathcal{Z}$ the momentary state, action, and observation, respectively, $T(x_k, a_k, x_{k+1}) = \mathbb{P}_T(x_{k+1}|x_k, a_k)$ is the stochastic transition model from the past momentary state x_k to the next x_{k+1} through action a_k, $O(z_k, x_k) = \mathbb{P}_Z(z_k|x_k)$ is the stochastic observation model, $\rho\left(b_{k+1}, z_{k+1}, a_k, b_k\right)$ is a scalar reward operator, $\gamma \in [0, 1]$ is the discount factor, and b_0 is the belief about the initial state (prior). Notably, the infinite horizon planning case necessitates $\gamma < 1$, whereas $\gamma = 1$ is permitted in a finite horizon. In this paper, we focus on the finite horizon setting. Moreover, the reward can be dependent on consecutive beliefs and the elements relating them (e.g., information gain [10]).

2.2. Belief space planning

The posterior belief at time instant k is given by

$$b_k(x_k) \approx \mathbb{P}\left(x_k | b_0, a_{0:k-1}, z_{1:k}\right). \tag{2}$$

The belief is an efficient way of storing all relevant information that has been obtained so far. The usual assumption is that the belief is a sufficient statistic for decision making objective [3]. However, in practice, the belief requires some representation. In general, this representation is not perfect, e.g., parametric or sampled form; thus, in (2), we used the $\approx$ sign. In a real life scenario

$$b_k = \psi(\psi(\dots\psi(b_0, a_0, z_1), a_{k-2}, z_{k-1}), a_{k-1}, z_k), \tag{3}$$

where ψ is a method for updating the belief. Denote by π_ℓ policy at time step ℓ such that $\pi_\ell(b_\ell) = a_\ell$ maps belief to the action. It is noteworthy that policy $\pi(b)$ is a random function of the belief in general. For simplicity we assume that policy is deterministic. However, our development is not constrained to deterministic policies. By $\pi \triangleq \pi_{k:k+L-1}$ we denote a vector of policies for L time steps starting from time step k. Let us focus on the finite horizon setting. The general decision making under uncertainty objective function is of the following form

$$\begin{aligned} V^L(b_k, \pi) &= \varphi\left(\mathbb{P}\left(\rho_{k+1:k+L} | b_k, \pi_{k:k+L-1}\right), g_k\right) \\ &\text{s.t. } b_\ell = \psi(b_{\ell-1}, \pi_{\ell-1}(b_{\ell-1}), z_\ell), \end{aligned} \tag{4}$$

where L is the planning horizon, ρ_ℓ is a random immediate reward, φ is an objective operator, and $g_k \triangleq f_{g_k}(\rho_{k+1:k+L})$ is the return [38]. The return is a deterministic function of the realization of $\rho_{k+1:k+L}$. A common choice for φ is expectation over the distribution of future rewards given all data available [6]. The return is some known function of the realization

of $\rho_{k+1:k+L}$; as discussed in [6], e.g., it could correspond to the cumulative reward $g_k = \sum_{\ell=1}^{L} \rho_{k+\ell}$. Finally, ψ is a general method for propagating the belief with action and updating it with the received observation.

The objective (4) is ultimately based on the *distribution of the return* given all information available for planning and some selected policy $\mathbb{P}(g_k|b_k, \pi_{k:k+L-1})$, which decomposes via marginalization over future observations $z_{k+} \equiv z_{k+1:k+L}$ as

$$\mathbb{P}(g_k|b_k, \pi) = \int_{z_{k+}} \mathbb{P}(g_k|b_k, \pi_{k:k+L-1}, z_{k+1:k+L}) \cdot \mathbb{P}(z_{k+1:k+L}|b_k, \pi_{k:k+L-1}) dz_{k+1:k+L}. \tag{5}$$

A conventional assumption is that $\mathbb{P}(g_k|b_k, \pi, z_{k+},)$ is a Dirac delta function.

3. Foundations

In this section we extend POMDP with belief-dependent rewards to probabilistic POMDP and rigorously define the *simplification* paradigm. We further continue to the formulation of the general bounds on the reward/return which can be analytical or stochastic. We conclude this section with our key insight.

3.1. Extended setting, probabilistic POMDP with belief dependent reward

Sometimes the belief $b_{\ell-1}$ has a simple parametric form, where $\theta_{\ell-1}$ is a vector of parameters, e.g., a Gaussian belief. In this case, belief update ψ can be deterministic, and is denoted by $\psi_{\text{dt}}(\theta_{\ell-1}, \pi_{\ell-1}(\theta_{\ell-1}), z_\ell)$, where the subscript dt stands for deterministic. In more general and challenging scenarios the belief $b_{\ell-1}$ is given by a set of weighted samples $\{(w^i_{\ell-1}, x^i_{\ell-1})\}_{i=1}^N$. Therefore, ψ is a stochastic method, e.g., a particle filter [42]. Applying multiple times ψ on the same input will yield different sets of samples approximating the same distribution of the posterior belief. We denote the stochastic ψ by $\psi_{\text{st}}(b_{\ell-1}, \pi_{\ell-1}(b_{\ell-1}), z_\ell)$. Thus, ψ_{st} is a random function of the previous belief, an action and the observation. Note also another common situation where $b_{\ell-1}$ is parameterized, but there is no closed form update. In this case, ψ is also a stochastic method. Another form to formulate the above is that the distribution

$$B(b_{\ell-1}, \pi_{\ell-1}(b_{\ell-1}), z_\ell, b_\ell) \triangleq \mathbb{P}_B\left(b_\ell|b_{\ell-1}, \pi_{\ell-1}, z_\ell\right), \tag{6}$$

is not a Dirac delta function. This aspect was disregarded so far, to the best of our knowledge. Note that in a Belief MDP (BMDP) formulation, the assumption is that B is a Dirac delta function.

Similar arguments also hold for the momentary reward operator of the belief and the previous action. In its pure theoretical form, the momentary reward is a deterministic operator of the posterior belief and possibly an action. For example, a common immediate reward is of the form

$$\rho_{\text{dt}}(b) = \mathbb{E}_{x\sim b}\left[f(b(x), x)\right] = \int_x b(x) f(b(x), x) dx, \tag{7}$$

where usually $f(b(x), x) = -\log b(x)$ or some reward on the state $f(b(x), x) = r(x)$, producing differential entropy or mean distance to goal. Unfortunately, an analytical expression for the reward operator $\rho_{\text{dt}}(\cdot)$ is available in only limited scenarios, e.g., if the belief is modeled as Gaussian and the reward is differential entropy. The representation of the beliefs in (6) dictates adequate practical reward operators. Sometimes the deterministic operator can be constructed on top of a particular belief representation. E.g., (6) outputs a set of weighted samples and (7) is adapted to be a deterministic operator of this output [4]. However, it is not always possible. In extremely challenging situations the reward includes modification of the representation of the belief. This could introduce an additional source of stochasticity. We extend (7) to

$$R(b_{\ell-1}, \pi_{\ell-1}(b_{\ell-1}), z_\ell, b_\ell, \rho_\ell) \triangleq \mathbb{P}_R\left(\rho_\ell|b_\ell, z_\ell, \pi_{\ell-1}(b_{\ell-1}), b_{\ell-1}\right), \tag{8}$$

embracing these possibilities. To our knowledge, we are the first who treat these aspects as random.

Before introducing simplification formally and analyzing its impact, we shall account for all potential sources of variability. We remove conventional approximations by extending (1) to a probabilistic reward model R (8) and probabilistic belief update B (6), and introduce

$$M = \langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, R, \gamma, b_k, B \rangle, \tag{9}$$

which we name probabilistic ρ-POMDP ($\mathbb{P}\rho$-POMDP). The rationale behind these conditional distributions (R and B) is to capture additional sources of stochasticity, such as stochastic belief update, stochastic calculation of a given reward operator or simply not knowing the operator reward in an explicit analytic form.

As discussed earlier, the value function (4) is based on (5). These previously overlooked sources of stochasticity impact the likelihood of the observations

$$\mathbb{P}\left(z_{k+1:k+L}|b_k, \pi\right), \tag{10}$$

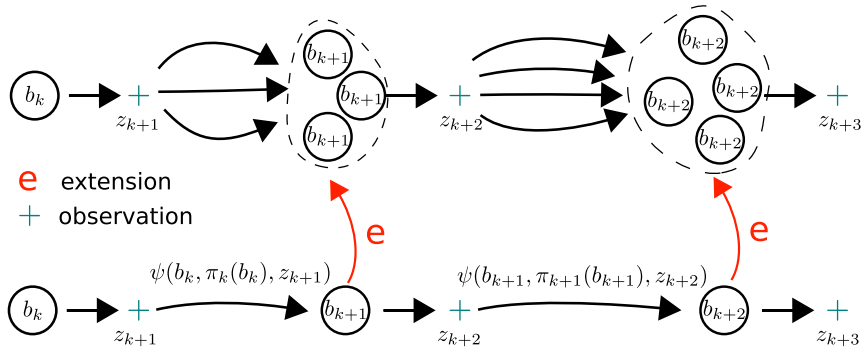


Fig. 1. Illustration of one branch of the extended belief tree. In a conventional setting (bottom), under the policy π, a specific realization of observations $z_{k+1:k+3}$ defines the beliefs along the way. In our extended setting (top), that is not the case, as discussed in text. It is customary to choose the same beliefs used to build the tree to obtain reward distribution or samples from the reward. We decoupled beliefs from the tree and beliefs from the reward calculation. By the red arrow, we denote our extension (red e). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

as well as the joint reward distribution $\mathbb{P}\left(\rho_{k+}|b_k, \pi, z_{k+}\right) \equiv \mathbb{P}\left(\rho_{k+1:k+L}|b_k, \pi_{k:k+L-1}, z_{k+1:k+L}\right)$ given a realization of future observations. The latter can be factorized as

$$\begin{aligned}&\mathbb{P}\left(\rho_{k+}|b_k, \pi, z_{k+}\right) = \\ &\int_{b_{k+1}} \mathbb{P}_R\left(\rho_{k+1}|b_{k+1}, z_{k+1}, \pi_k, b_k\right) \mathbb{P}_B\left(b_{k+1}|b_k, \pi_k, z_{k+1}\right) \\ &\int_{b_{k+2}} \dots \int_{b_{k+L}} \mathbb{P}_R\left(\rho_{k+L}|b_{k+L}, z_{k+L}, \pi_{k+L-1}, b_{k+L-1}\right) \mathbb{P}_B\left(b_{k+L}|b_{k+L-1}, \pi_{k+L-1}, z_{k+L}\right) \mathrm{d}b_{k+L} \dots \mathrm{d}b_{k+2} \mathrm{d}b_{k+1}. \end{aligned} \tag{11}$$

In contrast, in the regular setting of POMDP and ρ-POMDP $\mathbb{P}\left(\rho_{k+}|b_k, \pi, z_{k+}\right)$ is Dirac's delta function. If B is a Dirac function, a sample from (10) uniquely defines the corresponding posterior beliefs $b_{k+1:k+L}$. This, therefore, corresponds to the classical belief tree (R could still be non a Dirac function). In contrast, our $\mathbb{P}\rho$-POMDP (9), corresponds to an *extended* belief tree, which, due to (6), allows many samples of the beliefs $b_{k+1:k+L}$ for each sample of $z_{k+1:k+L}$ from (10). We illustrate this in Fig. 1.

3.2. Simplification formulation

To formally define the simplification procedure, we augment the $\mathbb{P}\rho$-POMDP tuple (9) with a simplification operator ν,

$$M_\nu = \langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, R, \gamma, b_k, B, \nu \rangle, \quad \nu \triangleq \nu_k, \dots, \nu_{k+L}. \tag{12}$$

This general operator defines any possible modification of the original problem defined by (9) alongside with (4) to a new, simpler to solve, problem. The definition (12) allows us to retain the connection to the original nonsimplified problem (9) and examine the impact of the simplification on (9). Further, we also define a novel decision making problem, undergoing simplification to ease the computational burden. The operator ν can be for example, sparsification of the initial belief b_k [8], substitution of the operator differential entropy by a simpler operator, e.g., trace of covariance matrix, discarding the normalizer in the differential entropy operator [30], replacing the reward by its topological signature [19], direct calculation of lightweight reward bounds [40], selecting a subset of hypotheses in a hybrid or mixture belief [32]. In Section 6, we consider a specific simplification of taking less samples for reward calculation considering parametric and non-parametric beliefs.

Generally, M and M_ν are different decision making problems. We shall be interested in working online with the latter while providing the guarantees with respect to the former.

To distinguish a simplified reward from the original reward, we denote the former by $\breve{\rho}$ instead of ρ; similarly, we denote the simplified belief by $\breve{b}$ instead of b. Note the operator ν can be stochastic, as discussed below.

Specifically, belief simplification is described by the distribution

$$\mathbb{P}(\breve{b}_\ell | b_\ell; \nu_\ell^b). \tag{13}$$

In general, the distribution (13) over the simplified belief $\breve{b}_\ell$ corresponds to a stochastic simplification operator ν_ℓ^b. This is the case, for example, when b_ℓ is represented by a set of N weighted samples and ν_ℓ^b is the operation of subsampling n samples according to weights; i.e., applying this operation on b_ℓ multiple times leads to different sets of n samples, each

representing another realization of $\check{b}_\ell$ from (13). Overall there are $\binom{N}{n}$ such combinations. For a deterministic operator ν^b_ℓ, (13) is a Dirac function.

Further, there are several cases of how a simplification affects belief update (6) from time $\ell-1$ to ℓ.

1. Without any simplification we have $\mathbb{P}_B(b_\ell|b_{\ell-1},\pi_{\ell-1},z_\ell)$ from (6).
2. Given a simplified belief $\check{b}_{\ell-1}$, while keeping the original stochastic belief update ψ_{st}, we have

$$\mathbb{P}_B(\check{b}_\ell|\check{b}_{\ell-1},\pi_{\ell-1},z_\ell),$$

where each realization of $\check{b}_\ell$ is obtained via ψ_{st}. Thus, given $\check{b}_{\ell-1}$, this distribution is not a function of ν.
3. We can also simplify the belief update operator, ψ_{st}, to $\check{\psi}_{\text{st}}$. Denoting the corresponding simplification operator ν^ψ_ℓ, this yields

$$\mathbb{P}_{\check{B}}(\check{b}_\ell|\check{b}_{\ell-1},\pi_{\ell-1},z_\ell;\nu^\psi_\ell).$$

4. Finally, one can decide at time ℓ to apply simplification on the belief (determined by ν^b_ℓ) via (13). The corresponding belief update can be written as

$$\mathbb{P}_{\check{B}}(\check{b}_\ell|\check{b}_{\ell-1},\pi_{\ell-1},z_\ell;\nu^b_\ell,\nu^\psi_\ell)=\int_{\tilde{b}_\ell}\mathbb{P}(\check{b}_\ell|\tilde{b}_\ell;\nu^b_\ell)\mathbb{P}_{\check{B}}(\tilde{b}_\ell|\check{b}_{\ell-1},\pi_{\ell-1},z_\ell;\nu^\psi_\ell)\mathrm{d}\tilde{b}_\ell,$$

where $\tilde{b}_\ell$ is the integration variable.

We combine these cases and write

$$\check{B}\left(\check{b}_{\ell-1},\pi_{\ell-1},z_\ell,\check{b}_\ell;\nu\right)\triangleq\mathbb{P}_{\check{B}}(\check{b}_\ell|\check{b}_{\ell-1},\pi_{\ell-1},z_\ell;\nu^b_\ell,\nu^\psi_\ell). \tag{14}$$

Similarly, reward simplification could be, in general, stochastic, leading to the distribution

$$\mathbb{P}(\check{\rho}_\ell|\rho_\ell;\nu^\rho_\ell). \tag{15}$$

Thus, given a simplified belief $\check{b}_\ell$ and $\check{b}_{\ell-1}$, and recalling (8), the distribution over $\check{\rho}_\ell$ is

$$\mathbb{P}_{\check{R}}(\check{\rho}_\ell|\check{b}_\ell,z_\ell,\pi_{\ell-1}(\check{b}_{\ell-1}),\check{b}_{\ell-1};\nu)=\int_{\tilde{\rho}_\ell}\mathbb{P}(\check{\rho}_\ell|\tilde{\rho}_\ell;\nu^\rho_\ell)\mathbb{P}_R(\tilde{\rho}_\ell|\check{b}_\ell,z_\ell,\pi_{\ell-1}(\check{b}_{\ell-1}),\check{b}_{\ell-1})\mathrm{d}\tilde{\rho}_\ell,$$

which we denote as the simplified reward model,

$$\check{R}(\check{b}_\ell,z_\ell,\pi_{\ell-1}(\check{b}_{\ell-1}),\check{\rho}_\ell;\nu)\triangleq\mathbb{P}_{\check{R}}\left(\check{\rho}_\ell|\check{b}_\ell,z_\ell,\pi_{\ell-1}(\check{b}_{\ell-1}),\check{b}_{\ell-1};\nu\right). \tag{16}$$

Throughout the document we assume that operator ν does not affect the observations likelihood. In other words, the measurements are sampled as in the original problem as in (10).

3.2.1. Joint distribution of simplified and the original reward given the candidate policy and the observations

Consequently, the models (14) and (16) impact (11), and lead to several alternatives for the original and the simplified joint reward distribution given a realization of the future observations. The first alternative is to simplify the initial belief b_k to $\check{b}_k$ and apply the update method ψ_{st} on the simplified belief

$$\begin{aligned}
&\mathbb{P}\left(\rho_{k+},\check{\rho}_{k+}|b_k,\pi,z_{k+},\nu\right)=\int_{\check{b}_k}\mathbb{P}(\check{b}_k|b_k;\nu^b_k)\int_{b_{k+1}}\int_{\check{b}_{k+1}}\mathbb{P}_{\check{B}}\left(\check{b}_{k+1}|\check{b}_k,\pi_k,z_{k+1};\nu\right)\mathbb{P}_B\left(b_{k+1}|b_k,\pi_k,z_{k+1}\right)\\
&\cdot\mathbb{P}_{\check{R}}\left(\check{\rho}_{k+1}|\check{b}_{k+1},z_{k+1},\pi_k,\check{b}_k;\nu\right)\mathbb{P}_R\left(\rho_{k+1}|b_{k+1},z_{k+1},\pi_k,b_k\right)\int_{b_{k+2}}\int_{\check{b}_{k+2}}\dots\\
&\int_{b_{k+L}}\int_{\check{b}_{k+L}}\mathbb{P}_{\check{B}}\left(\check{b}_{k+L}|\check{b}_{k+L-1},\pi_{k+L-1},z_{k+L};\nu\right)\mathbb{P}_B\left(b_{k+L}|b_{k+L-1},\pi_{k+L-1},z_{k+L}\right)\\
&\mathbb{P}_{\check{R}}\left(\check{\rho}_{k+L}|\check{b}_{k+L},z_{k+L},\pi_{k+L-1},\check{b}_{k+L-1};\nu\right)\mathbb{P}_R\left(\rho_{k+L}|b_{k+L},z_{k+L},\pi_{k+L-1},b_{k+L-1}\right)\mathrm{d}b_{k+L}\mathrm{d}\check{b}_{k+L}\dots\\
&\mathrm{d}b_{k+2}\mathrm{d}\check{b}_{k+2}\mathrm{d}b_{k+1}\mathrm{d}\check{b}_{k+1}\mathrm{d}\check{b}_k.
\end{aligned} \tag{17}$$

The second alternative is to maintain/update the original belief. In this situation, we maintain and update the original belief and then use it to determine a simplified belief to calculate the simplified reward. This is in contrast to updating based on simplified beliefs from previous times as in (17). Thus,

$$\begin{aligned}
&\mathbb{P}\left(\rho_{k+}, \check{\rho}_{k+} | b_k, \pi, z_{k+}, \nu\right) = \\
&\int_{\check{b}_k} \mathbb{P}(\check{b}_k | b_k; \nu_k^b) \int_{b_{k+1}} \int_{\check{b}_{k+1}} \mathbb{P}_{\check{R}}(\check{\rho}_{k+1} | \check{b}_{k+1}, z_{k+1}, \pi_k(\check{b}_k), \check{b}_k; \nu) \mathbb{P}_R\left(\rho_{k+1} | b_{k+1}, z_{k+1}, \pi_k, b_k\right) \\
&\mathbb{P}(\check{b}_{k+1} | b_{k+1}; \nu_{k+1}^b) \mathbb{P}_B\left(b_{k+1} | b_k, \pi_k, z_{k+1}\right) \int_{b_{k+2}} \int_{\check{b}_{k+2}} \dots \\
&\int_{b_{k+L}} \int_{\check{b}_{k+L}} \mathbb{P}_{\check{R}}(\check{\rho}_{k+L} | \check{b}_{k+L}, z_{k+L}, \pi_{k+L-1}(\check{b}_{k+L-1}), \check{b}_{k+L-1}; \nu) \mathbb{P}_R\left(\rho_{k+L} | b_{k+L}, z_{k+L}, \pi_{k+L-1}, b_{k+L-1}\right) \\
&\mathbb{P}(\check{b}_{k+L} | b_{k+L}; \nu_{k+L}^b) \mathbb{P}_B\left(b_{k+L} | b_{k+L-1}, \pi_{k+L-1}, z_{k+L}\right) \mathrm{d}b_{k+L} \mathrm{d}\check{b}_{k+L} \dots \mathrm{d}b_{k+2} \mathrm{d}\check{b}_{k+2} \mathrm{d}b_{k+1} \mathrm{d}\check{b}_{k+1} \mathrm{d}\check{b}_k.
\end{aligned} \tag{18}$$

Having introduced the two alternatives above, we are ready to go through their differences. The simplification approach defined by (17) uses only the observations from the belief tree. In the sequel, we explain why it is advantageous. In this setting, in addition to maintaining/updating b_ℓ for constructing the extended belief tree, we also have to maintain/update the simplified version $\check{b}_\ell$. Nevertheless, the advantage is that by definition of (17), we nullify covariance between simplified and the original reward/return as opposed to the equation (18). In other words, considering (17) one can write,

$$\mathbb{P}\left(\rho_{k+}, \check{\rho}_{k+} | b_k, \pi, z_{k+}, \nu\right) = \mathbb{P}\left(\rho_{k+} | b_k, \pi, z_{k+}\right) \mathbb{P}\left(\check{\rho}_{k+} | b_k, \pi, z_{k+}, \nu\right) \tag{19}$$

Alternatively, if maintaining/updating a simplified belief is not desirable, e.g., belief is given as a surface but no closed form solution for reward exists, the update of a simplified belief $\check{b}_\ell$ can be avoided. Towards this end we use (18). This is in contrast to updating based on simplified beliefs from previous times as in (17). Such simplification does not necessarily require to sample the original beliefs again. One could utilize original beliefs already present in the belief tree for simulating the observations. In the next section we delve into the subject of the bounds. Importantly, from structure of (18) we see that this distribution cannot be broken down to the multiplication of the marginals as in (19). In particular, the correlation is present through the component $\mathbb{P}(\check{b}_\ell | b_\ell; \nu_\ell^b)$.

Note, sometimes estimators of the reward, e.g., [4] require a specific connection between two consecutive beliefs.

3.3. Online stochastic and analytical bounds

While thus far we considered the joint distribution over original and simplified rewards, $\mathbb{P}\left(\rho_{k+}, \check{\rho}_{k+} | b_k, \pi, z_{k+}, \nu\right)$, in an online setting we do not have access to the original rewards as calculating them explicitly defeats the purpose of simplification. Instead, we shall now utilize simplification to provide bounds over the original rewards. These bounds can be used to provide performance guarantees, and should be cheaper to calculate than the original rewards.

Further, the bounds can be analytical as in previous simplification approaches, e.g., [8]. The bounds can be obtained via a simplified reward or directly as in [39]. For example, the authors of [39] proposed lightweight analytical adaptive bounds, calculated from the original belief, such that $l_\ell \leq \rho_\ell \leq u_\ell$ is always true by definition. If the bounds are calculated directly, we skip (18) and have instead the following.

$$\begin{aligned}
&\mathbb{P}\left(\rho_{k+}, l_{k+}, u_{k+} | b_k, \pi, z_{k+}, \nu\right) = \int_{\check{b}_k} \mathbb{P}(\check{b}_k | b_k; \nu_k^b) \int_{b_{k+1}} \int_{\check{b}_{k+1}} \\
&\mathbb{P}(\rho_{k+1}, l_{k+1}, u_{k+1} | b_{k+1}, \check{b}_{k+1}, z_{k+1}, \pi_k(b_k), b_k, \check{b}_k; \nu) \mathbb{P}(\check{b}_{k+1} | b_{k+1}; \nu_{k+1}^b) \mathbb{P}_B\left(b_{k+1} | b_k, \pi_k, z_{k+1}\right) \\
&\int_{b_{k+2}} \int_{\check{b}_{k+2}} \dots \int_{b_{k+L}} \int_{\check{b}_{k+L}} \mathbb{P}(\rho_{k+L}, l_{k+L}, u_{k+L} | b_{k+L}, \check{b}_{k+L}, z_{k+L}, \pi_{k+L-1}(b_{k+L-1}), b_{k+L-1}, \check{b}_{k+L-1}; \nu) \\
&\mathbb{P}(\check{b}_{k+L} | b_{k+L}; \nu_{k+L}^b) \mathbb{P}_B\left(b_{k+L} | b_{k+L-1}, \pi_{k+L-1}, z_{k+L-1}\right) \mathrm{d}\check{b}_{k+L} \mathrm{d}b_{k+L} \dots \mathrm{d}\check{b}_{k+2} \mathrm{d}b_{k+2} \mathrm{d}\check{b}_{k+1} \mathrm{d}b_{k+1} \mathrm{d}\check{b}_k.
\end{aligned} \tag{20}$$

The simplification type depicted by (20) is an extension of *in-place simplification* described in [39] to an extended setting. Ultimately for each realization of the return we are interested in the following relation

$$l \leq g_k \leq u. \tag{21}$$

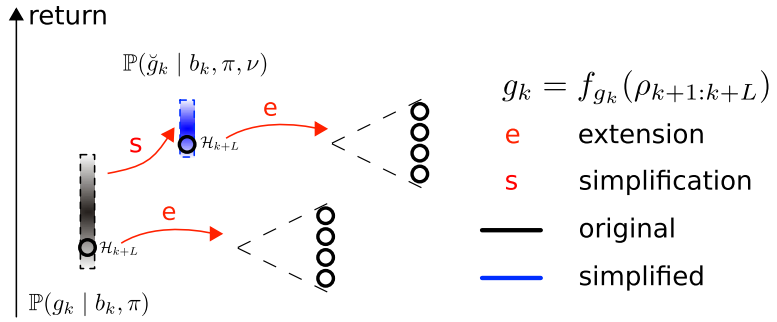


Fig. 2. Our extended setting permits variability of the reward given the present and a realization of the future. On the contrary, in a conventional setting, (23) is always a Dirac delta function. Our extension reflects on the original distribution of the return as well as the simplified.

One way to do that is to develop analytical bounds, which will hold for any possible observation $z_{k+1:k+L}$ received and any realization of return, e.g., as in [39].

In this section we show that there is another way to find more lenient bounds. Let $\mathcal{H}_{k+L} \triangleq \{b_k, \pi, z_{k+}\}$ be future history at the time index $k+L$. Our extension allows R and B, as well as $\breve{R}$ and $\breve{B}$ to be any distributions.

They can remain Dirac functions, e.g., if belief update and the reward calculation have a closed form

$$\mathbb{P}\left(\rho_\ell | b_{\ell-1}, a_{\ell-1}, z_\ell\right) = \delta(\rho_\ell - \rho_{\mathrm{dt}}(\psi_{\mathrm{dt}}(\theta_{\ell-1}, a_{\ell-1}, z_\ell))). \tag{22}$$

Successively, $\mathbb{P}(g_k | b_k, \pi, z_{k+},)$ remains Dirac delta. However, in the more general case, following our extension, there is a joint distribution of original and simplified returns given a realization of the future and the present

$$\mathbb{P}(g_k, \breve{g}_k | \mathcal{H}_{k+L}, \nu), \tag{23}$$

as illustrated in Fig. 2. As we observe in Fig. 2, given the history $\mathcal{H}_{k+L}$, the return g_k as well as the simplified return $\breve{g}_k$ has variability, in contrast to the conventional approach. Ordinarily, the belief update is commenced once and treated as deterministic. So as the rewards and return do not have variance given the history of the actions and the observations. Since (23) is no longer a Dirac function, we can use knowledge about this distribution to design bounds, which will hold with *some* probability. In Section 6, we show that it is possible to harness the structure of (23) to design the mentioned more lenient online bounds.

Our framework permits to detach the process of estimation of the bounds from the realization of the reward and truly use all accessible information in a simplified problem. For example, one way to design probabilistic bounds is to find online a random variable or deterministic scalar ϵ such that the probability

$$P(|g_k - \breve{g}_k| \leq \epsilon | \mathcal{H}_{k+L}, \nu) \tag{24}$$

is bounded from below. The corresponding probabilistic lower and upper bounds will be $l = \breve{g}_k - \epsilon$ and $u = \breve{g}_k + \epsilon$, respectively. We, therefore, refer to l and u as random variables. In our setting, even if the bounds actually bound with very low probability, it is still possible to analyze the quality of the simplification. Moreover, analytical bounds, designed in a conventional setting, can be used in our extended setting without any revision. In our extended setting, they will bound with probability one.

Having introduced the novel stochastic bounds, we proceed to the formulation of the constraints, that these bounds shall fulfill to be meaningful. The following conditional $\mathbb{P}(g_k, \breve{g}_k, l, u | \mathcal{H}_{k+L}, \nu)$ encloses all the variables situated in the problem. Let the parameter controlling the confidence level be $\alpha \in [0, 1)$. For every possible sample $\breve{g}_k$ we do not know which sample g_k one could obtain in the original problem. However, if the bounds are designed such that $\mathbb{P}(g_k, l, u | \mathcal{H}_{k+L}, \nu)$ render

$$1 - \alpha \leq P\left(\mathbf{1}\{l \leq g_k \leq u\} = 1 | \mathcal{H}_{k+L}, \nu\right) \tag{25}$$

these bounds can be useful. Notably, the above equation does not involve simplified return, so is applicable also in the case bounds are directly formulated (and not via a simplified return). However, in this case the bounds are analytical and $\alpha = 0$. To summarize, there are three types of online reward/return bounds:

1. Deterministic bounds. These analytical bounds exist in case of a closed form belief update ψ_{dt} and a deterministic operator reward $\rho_{\mathrm{dt}}(b)$ from (7), e.g., belief is a Gaussian and the reward is differential entropy. In this case, even in our extended setting R and B remain Dirac functions.
2. Stochastic bounds that hold with probability one, namely $\alpha = 0$. These are also analytical bounds. In our extended setting R and B are no longer Dirac functions. However, these bounds hold for any realization of sample approximation, as stated around (21).
3. Stochastic bounds that hold at least with probability $1 - \alpha$. They exist only in our extended setting when R and B are not Dirac functions.

3.4. Key insight - characterization of the return using stochastic bounds

Let us recite that our goal is to accelerate the decision making. We recall the notion of "usual stochastic order" and interpret the definition within our context.

Usual stochastic order implies, that if for three random variables l, g_k, u given b_k, π holds $l \leq g_k$ and $g_k \leq u$ for $\forall \omega \in \Omega$, so $\forall \xi \in (-\infty, \infty)$

$$P(l > \xi | b_k, \pi, \nu) \leq P(g_k > \xi | b_k, \pi) \leq P(u > \xi | b_k, \pi, \nu). \tag{26}$$

Let us present out main theorem which we will extensively use further.

Theorem 1 *(Characterization of the return using stochastic bounds). Fix $\alpha \in [0, 1)$. Assume that (25) holds. This implies that $\forall \xi \in (-\infty, \infty)$*

$$(P(l > \xi | b_k, \pi, \nu) - \alpha)(1 - \alpha) \leq P(g_k > \xi | b_k, \pi) \leq \frac{P(u > \xi | b_k, \pi, \nu)}{1 - \alpha} + \alpha. \tag{27}$$

For the detailed proof please refer to Appendix A.1. Let us further improve the bounds as such

$$\mathcal{LB}_\alpha(\xi) = 0 \vee \big(P(l > \xi | b_k, \pi, \nu) - \alpha\big)(1 - \alpha) \leq P(g_k > \xi | b_k, \pi), \tag{28}$$

where $\vee$ is a maximum operator.

$$P(g_k > \xi | b_k, \pi) \leq 1 \wedge \left(\frac{P(u > \xi | b_k, \pi, \nu)}{1 - \alpha} + \alpha \right) = \mathcal{UB}_\alpha(\xi), \tag{29}$$

where $\wedge$ is the minimum operator.

4. Simplification impact on a marginal distribution of the return

Previously, we defined a simplification procedure that results in a corresponding new decision making problem that should be easier to solve. From $\mathbb{P}\left(\rho_{k+} | b_k, \pi, z_{k+}, \nu\right)$ and $\mathbb{P}\left(\breve{\rho}_{k+} | b_k, \pi, z_{k+}, \nu\right)$ we arrive at the distribution of the original as well as simplified returns $\mathbb{P}(g_k | b_k, \pi)$ and $\mathbb{P}(\breve{g}_k | b_k, \pi, \nu)$ for the evaluated candidate policy. In this section, we show how the stochastic bounds can be utilized in the context of known risk aware objectives such as VaR. Notably, this section presents a discussion concerning the marginal distribution of the pair of the returns - original and simplified given a candidate policy.

4.1. Distributions affected by the simplification

In this section we decompose the distribution of interest. To grasp the simplification impact we shall assess the relation between simplified and original returns portrayed by the following distribution

$$\mathbb{P}(g_k, \breve{g}_k | b_k, \pi_{k:k+L-1}, \nu) = \int_{z_{k+}} \mathbb{P}(g_k, \breve{g}_k | b_k, \pi, z_{k+}, \nu) \cdot \mathbb{P}(z_{k+} | b_k, \pi) dz_{k+}. \tag{30}$$

Recall that $z_{k+1:k+L} \equiv z_{k+}$. In general the simplification operator ν can affect also the observation likelihood (10). We leave it to future research.

4.2. Decision making

Algorithm 1 Generic *simplified with performance guarantees* sampling based decision making algorithm with challenging rewards and objectives (note that the theory is formulated for policies but here we discuss discrete space of action sequences $\mathcal{A}$).

Input: belief b_k, action space $\mathcal{A}$.
for action sequence a^i from all possible action sequences $i \in 1 : |\mathcal{A}|$ **do**
  Sample returns and calculate interval $\mathcal{LB}^i, \mathcal{UB}^i$.
end for
Set optimal action sequence a^{i^*} by $i^* = \arg\max_{i=1:|\mathcal{A}|} \mathcal{LB}^i$
Find $j^\star = \arg\max_{j=1:|\mathcal{A}| \setminus i^*} \mathcal{UB}^j$. Define loss incurred by the simplification as follows $\max\{0, \mathcal{UB}^{j^\star} - \mathcal{LB}^{i^*}\}$;
In case that absolute loss doesn't have meaning, define relative loss as follows $\frac{\max\{0, \mathcal{UB}^{j^\star} - \mathcal{LB}^{i^*}\}}{\min\{|\mathcal{LB}^{i^*}|, |\mathcal{UB}^{j^\star}|\}} \geq \frac{V^* - V(b_k, a^{i^*})}{|V^*|}$, where V^* is true optimal value.

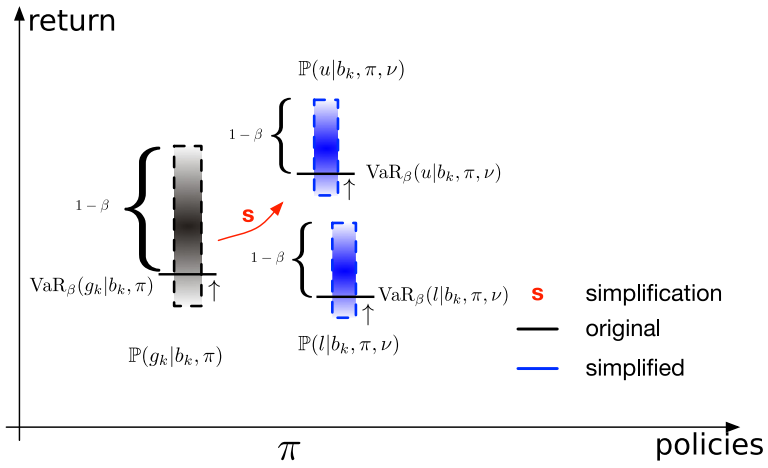


Fig. 3. Illustration of Value at Risk. Probability that the return will be under VaR is β. In other words VaR is β-quantile of the return.

In this section, we apply our findings in order to bound popular objectives with lightweight bounds to accelerate the decision making mechanism. We skip the most common objective operator, the expectation, since it is *oblivious* to the distribution of the returns. Therefore the expectation is not a risk averse objective. Motivated by this assertion, we consider more versatile objectives. Since it is not clear whether or not these objectives conform to Bellman form we incorporate simplification with generic decision making (Algorithm 1). In this algorithm, we traverse the loop over the possible action sequences. We calculate the interval defined by the upper and lower bound on the value function in each loop iteration. Finally, we select the action sequence with the highest lower bound and compare it with the highest upper bound corresponding to all other action sequences. We report a relative loss. Using analytical or stochastic bounds on the return our goal is to bound the value function V as such

$$\mathcal{LB} \leq V \leq \mathcal{UB} \tag{31}$$

to accelerate the decision making. To our knowledge there are no attempts to simplify risk aware decision making under uncertainty through maintaining guarantees on the *simplification* impact.

Risk averse objectives One possible risk averse probabilistic objective for POMDP is

$$\mathbb{E}[\mathbf{1}\{g_k > a\}|b_k,\pi] = P(g_k > a|b_k,\pi), \tag{32}$$

where $a \in \mathbb{R}$. Maximizing this objective can be thought as maximizing the probability of achieving the target a. If we choose belief dependent reward to be the negative entropy $I(b_k) = -\mathcal{H}(b_k)$, set $a = I(b_k)$ and the return to be terminal reward $g_k = \rho_L$, or $g_k = \frac{1}{L}\sum_{\ell=k+1}^{k+L} \rho_\ell$ [21], such objective quantifies the probability that information gain is positive; and such decision making prefers the action which maximizes probability of positive information gain. We can further control amount of most probable information gain by setting $a = c \cdot I(b_k)$, where c is a factor larger than one. Once optimal action is obtained we are confident that $g_k > a$ with probability $P(g_k > a|b_k,\pi)$. Substituting ξ by a the bounds from (28) and (29) hold for this objective.

Another objective is reward variant of Value at Risk (VaR) (Fig. 3)

$$\mathrm{VaR}_\beta(g_k|b_k,\pi) = \sup\{\xi \text{ s.t } P(g_k > \xi|b_k,\pi) \geq 1-\beta\}. \tag{33}$$

This objective articulates that we are interested in the maximal worst case return. Meaning maximal return such that probability mass to be above this return is larger than $1-\beta$. Notably, if $g_k|b_k,\pi$ has a strictly increasing Cumulative Distribution Function (CDF), the VaR is its β-quantile $\mathrm{VaR}_\beta(g_k|b_k,\pi) = P^{-1}(g_k \leq \beta|b_k,\pi)$. The CDF of a continuous random variable is strictly increasing if it does not have intervals on a real line happening with probability zero. In the case of symmetrical distributions, the expected value overlaps with median, which is VaR with confidence level $\beta = 0.5$. Using again usual stochastic order, we can bound this objective with analytical return bounds as well as with stochastic return bounds. We start from analytical bounds. Let us focus on the lower bound. We want to show that

$$\mathrm{VaR}_\beta(l|b_k,\pi,\nu) \leq \mathrm{VaR}_\beta(g_k|b_k,\pi). \tag{34}$$

Since the bounds are analytical we use (26) to behold

$$\{\xi \text{ s.t. } P(l > \xi|b_k,\pi,\nu) \geq 1-\beta\} \subseteq \{\xi \text{ s.t. } P(g_k > \xi|b_k,\pi) \geq 1-\beta\}. \tag{35}$$

We know that maximum on the containing set is above or equal to maximum on the contained. This argument yields

$$\sup\{\xi \text{ s.t. } P(l>\xi|b_k,\pi)\geq 1-\beta\}\leq \sup\{\xi \text{ s.t. } P(g_k>\xi|b_k,\pi)\geq 1-\beta\}. \tag{36}$$

Switching roles of l to g_k and g_k to u we have that

$$\mathrm{VaR}_\beta(g_k|b_k,\pi,\nu)\leq \mathrm{VaR}_\beta(u|b_k,\pi,\nu). \tag{37}$$

Now let us bound the objective (33) using stochastic bounds.

Theorem 2 *(Deterministic bound of Value at Risk using stochastic bounds on the return). Assume that (25) holds. Let* $0\leq\alpha<1$, $0\leq\beta<1$. *Assume that* $\alpha(2-\alpha)\leq\beta\leq 1-\alpha$.

$$\mathrm{VaR}_{\frac{\beta-\alpha(2-\alpha)}{1-\alpha}}(l|b_k,\pi,\nu)\leq \mathrm{VaR}_\beta(g_k|b_k,\pi)\leq \mathrm{VaR}_{\beta+\alpha(2-\beta-\alpha)}(u|b_k,\pi,\nu). \tag{38}$$

The reader can find the detailed proof in Appendix A.2. Let us mention that the above bounds hold for theoretical objectives. In practice, however, the sample approximations are sufficiently close to the theoretical values.

5. Simplification impact on the joint distribution of the returns given two policies

So far, we analyzed marginal distributions over the returns/rewards corresponding to a candidate policy in the context of known risk aware objectives. Interestingly, if we consider the joint distribution over the returns corresponding to two candidate policies, as we further show, we can define novel objectives and harness the information encoded in the joint distribution.

We start by showing that $\mathbb{P}(g_k,g'_k|b_k,\pi,\pi')\neq\mathbb{P}(g_k|b_k,\pi)\cdot\mathbb{P}(g'_k|b_k,\pi')$. The source for correlation is the mutual likelihood of observations:

$$\mathbb{P}(g_k,g'_k|b_k,\pi,\pi')=\int\limits_{\substack{z_{k+}\\ z'_{k+}}}\mathbb{P}(g_k,g'_k|b_k,\pi,\pi',z_{k+},z'_{k+})\cdot\mathbb{P}(z_{k+},z'_{k+}|b_k,\pi,\pi')dz_{k+}dz'_{k+}= \tag{39}$$

$$\int\limits_{\substack{z_{k+}\\ z'_{k+}}}\mathbb{P}(g_k|b_k,\pi,z_{k+})\mathbb{P}(g'_k|b_k,\pi' z'_{k+})\cdot\mathbb{P}(z_{k+},z'_{k+}|b_k,\pi,\pi')dz_{k+}dz'_{k+} \tag{40}$$

Let us observe the joint likelihood of observations given the belief at present time and two candidate policies $\mathbb{P}\left(z_{k+1:k+L},z'_{k+1:k+L}|b_k,\pi,\pi'\right)$ breaks down using chain rule as follows

$$\begin{aligned}&\mathbb{P}\left(z_{k+1:k+L},z'_{k+1:k+L}|b_k,\pi,\pi'\right)=\mathbb{P}\left(z_{k+1},z'_{k+1}|b_k,\pi_k,\pi'_k\right)\\&\int\limits_{b_{k+1}}\int\limits_{b'_{k+1}}\mathbb{P}\left(z_{k+2}|b_{k+1},\pi_{k+1}\right)\mathbb{P}\left(z'_{k+2}|b'_{k+1},\pi'_{k+1}\right)\mathbb{P}_B\left(b_{k+1}|b_k,\pi_k,z_{k+1}\right)\mathbb{P}_B\left(b'_{k+1}|b'_k,\pi'_k,z'_{k+1}\right)\\&\int\limits_{b_{k+2}}\int\limits_{b'_{k+2}}\dots\int\limits_{b_{k+L-1}}\int\limits_{b'_{k+L-1}}\mathbb{P}\left(z_{k+L}|b_{k+L-1},\pi_{k+L-1}\right)\mathbb{P}\left(z'_{k+L}|b'_{k+L-1},\pi'_{k+L-1}\right)\cdot\\&\mathbb{P}_B\left(b_{k+L}|b_{k+L-1},\pi_{k+L-1},z_{k+L}\right)\mathbb{P}_B\left(b'_{k+L}|b'_{k+L-1},\pi'_{k+L-1},z'_{k+L}\right)\mathrm{d}b_{k+L-1}\mathrm{d}b'_{k+L-1}\dots\mathrm{d}b_{k+2}\mathrm{d}b'_{k+2}\mathrm{d}b_{k+1}\mathrm{d}b'_{k+1}\end{aligned} \tag{41}$$

The myopic observations $\mathbb{P}\left(z_{k+1},z'_{k+1}|b_k,\pi_k,\pi'_k\right)$ are correlated through the present time belief b_k. To see this explicitly we marginalize over the propagated states and employ the observation and motion models

$$\begin{aligned}&\mathbb{P}\left(z_{k+1},z'_{k+1}|b_k,\pi_k,\pi'_k\right)=\\&\int\limits_{\substack{x_{k+1}\\ x'_{k+1}\\ x_k}}\mathbb{P}_Z(z_{k+1}|x_{k+1})\mathbb{P}_Z(z'_{k+1}|x'_{k+1})\mathbb{P}_T(x_{k+1}|\pi_k(b_k),x_k)\mathbb{P}_T(x'_{k+1}|\pi'_k(b_k),x_k)b_k(x_k)\mathrm{d}x_{k+1}\mathrm{d}x'_{k+1}\mathrm{d}x_k.\end{aligned} \tag{42}$$

This insinuates that to base decision making on marginal means to lose this correlation which we aim to exploit. Prompted by this insight we suggest an objective uncovered in the next section.

Remark: Note that when the belief is parametric, we do not have a way to jointly parametrically propagate the belief with a pair of actions as in (42). However, we can always sample the parametric belief. So even if the belief b_k is parametric we still can account for correlation in (42) by switching to samples.

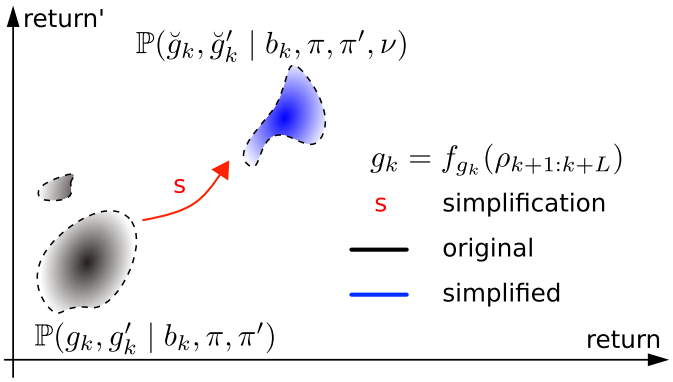


Fig. 4. This figure shows alteration of the distribution of joint returns g_k and g'_k of two candidate policies π and π' as a result of simplification. Color intensity denotes distribution values. This is a conceptual illustration, i.e., we do not imply higher/lower rewards or change of support due to simplification.

Extension of the decision making objective Let us define the objective to be maximized involving two candidate policies.

$$J^L(b_k, \pi, \pi') = \varphi\left(\mathbb{P}\left(\rho_{k+1:k+L}, \rho'_{k+1:k+L} | b_k, \pi, \pi'\right), (g_k, g'_k)\right) \quad \text{s.t. } b_\ell = \psi(b_{\ell-1}, \pi_{\ell-1}(b_{\ell-1}), z_\ell). \tag{43}$$

5.1. Distributions affected by simplification

Now we stipulate on the quality of the simplification for two candidate policies π and π'. To quantify the impact of the simplification procedure, we shall concentrate on the *joint* distribution of the pair of simplified and original returns appropriate for two candidate polices $\mathbb{P}(g_k, g'_k, \breve{g}_k, \breve{g}'_k | b_k, \pi, \pi', \nu)$. Our goal is to examine how the simplification procedure alters the joint distribution $\mathbb{P}(g_k, g'_k | b_k, \pi, \pi')$ towards $\mathbb{P}(\breve{g}_k, \breve{g}'_k | b_k, \pi, \pi', \nu)$. These two distributions are illustrated in Fig. 4. In the context of (43) we focus on

$$\mathbb{P}(g_k, g'_k, \breve{g}_k, \breve{g}'_k | b_k, \pi, \pi', \nu), \tag{44}$$

i.e., the joint distribution over original and simplified returns of both policies. This distribution decomposes via marginalization over future observations $z_{k+} \equiv z_{k+1:k+L}$ and $z'_{k+} \equiv z'_{k+1:k+L}$ as

$$\mathbb{P}(g_k, g'_k, \breve{g}_k, \breve{g}'_k | b_k, \pi, \pi', \nu) = \int_{\substack{z_{k+} \\ z'_{k+}}} \mathbb{P}(g_k, g'_k, \breve{g}_k, \breve{g}'_k | b_k, \pi, \pi', \nu, z_{k+}, z'_{k+}) \cdot \mathbb{P}(z_{k+}, z'_{k+} | b_k, \pi, \pi') dz_{k+} dz'_{k+}, \tag{45}$$

which, according to (6), (8) and (14)-(16), decomposes to

$$\int_{\substack{z_{k+} \\ z'_{k+}}} \mathbb{P}(g_k, \breve{g}_k | \mathcal{H}_{k+L}, \nu) \mathbb{P}(g'_k, \breve{g}'_k | \mathcal{H}'_{k+L}, \nu) \cdot \mathbb{P}(z_{k+}, z'_{k+} | b_k, \pi, \pi') dz_{k+} dz'_{k+}. \tag{46}$$

Note that the pair of histories is defined as follows $\mathcal{H}_{k+L} \triangleq \{b_k, \pi, z_{k+}\}$ and $\mathcal{H}'_{k+L} \triangleq \{b_k, \pi', z'_{k+}\}$; where the belief b_k is shared by both histories.

In other words, the simplification operator ν independently affects each realization of the future. Given two such realizations $(\mathcal{H}_{k+L}, \mathcal{H}'_{k+L}, \nu)$, the pairs of original and simplified returns are statistically independent of all other rewards. This crucial observation will be significant in the sequel.

5.2. Decision making

This section outlines a generic algorithm for decision making favoring pairwise joint distribution of the returns (Algorithm 2). This algorithm starts by taking the first action sequence as the best. We again traverse the loop over the possible action sequences. We compare two action sequences in each loop iteration and select the current optimal sequence. In the end, the optimal action sequence is optimal with respect to all possible action sequences. In particular, we propose a novel method to perform decision making using the joint distribution of the returns corresponding to the two candidate policies. The authors from [27] proposed to perform decision making with maximum likelihood observations. However, when the belief distribution is general and sophisticated, a generalization of [27] is to compare number of samples which fulfill the

Algorithm 2 Generic *simplified with performance guarantees* sampling based decision making based on pairwise joint distributions (note that the theory is formulated for policies but here we discuss given discrete space $\mathcal{A}$ of action sequences).

Input: belief b_k, $\mathcal{A}$.
$a^* \leftarrow a^1$
for action sequence a^i from all possible action sequences **do**
  Make simplified decision making using two actions a^* and a^i
  $a^* \leftarrow$ action defined as optimal in the line above
end for
return a^*

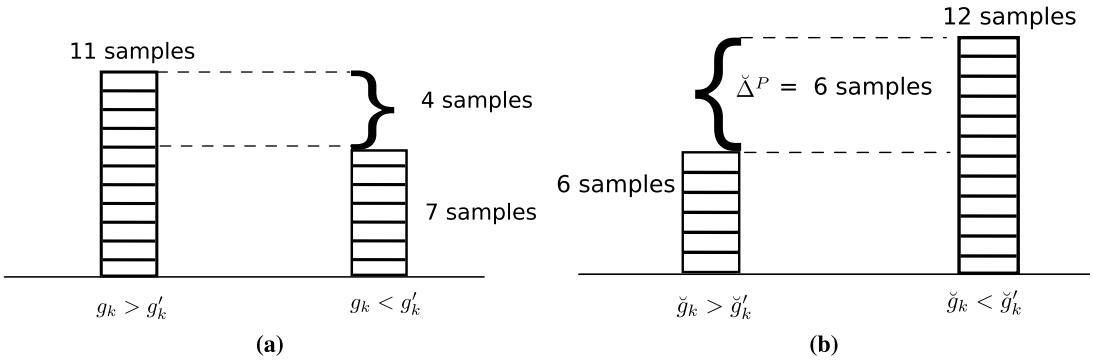


Fig. 5. (a) Hypothesis based decision making; **(b)** The outcome of decision making is wrong with margin 6 samples due to simplification.

hypothesis that $g_k > g'_k$ with number of samples satisfying $g_k < g'_k$. Such a decision making process can be thought as risk aware, since we are concerned with choosing an action which will be optimal with higher probability. Namely, if

$$\sum_{i=1}^{s} \mathbf{1}\{g_k^i > g_k'^i\} \geq \sum_{i=1}^{s} \mathbf{1}\{g_k^i < g_k'^i\}, \tag{47}$$

where the summation is over s samples of the pairs of the returns, we declare that π is better, else the π' is better. Note we assume that the event $g_k = g'_k$ happens with probability zero. This assumption is fulfilled with continuous distributions.

Simplified hypothesis based decision making Assume for the moment that bounds in (21) are analytical (hold with probability one), e.g., bounds from [39]. We can then define simplified returns as follows $\breve{g}_k = \frac{l+u}{2}$ and $\breve{g}'_k = \frac{l'+u'}{2}$. Simplification of the decision making portrayed by (47) is as follows. We take a simplified return instead of the original and ask if the following inequality is fulfilled

$$\sum_{i=1}^{s} \mathbf{1}\{\breve{g}_k^i > \breve{g}_k'^i\} \geq \sum_{i=1}^{s} \mathbf{1}\{\breve{g}_k^i < \breve{g}_k'^i\}. \tag{48}$$

If the answer is yes, we declare that π is better, else π' is better. Similar to not simplified decision making (47) we assume that the event $\breve{g}_k = \breve{g}'_k$ happens with probability zero.

The question is can we make a wrong conclusion with respect to the original problem due to the simplification, see Fig. 5. To provide guarantees on such a simplified decision making we first develop a novel mathematical tool we call Probabilistic Loss, which we believe has much bigger potential since it is able to describe the simplification impact for *any* operator objective φ. We then, in section 5.4, show how to provide guarantees for the specific objective operator (47).

5.3. Probabilistic loss (PLoss)

Let us define the following random variable, which we shall refer to as "loss"

$$\mathcal{L} \triangleq f_{\mathcal{L}}(g_k, g'_k, \breve{g}_k, \breve{g}'_k) = \begin{cases} \max\{g'_k - g_k, 0\} & \text{if } \breve{g}_k > \breve{g}'_k, \\ \max\{g_k - g'_k, 0\} & \text{if } \breve{g}_k < \breve{g}'_k. \end{cases} \tag{49}$$

With (49) we aim to capture a complete impact of a simplification onto the decision making problem (43). Specifically, this definition captures for each possible realization of $g_k, g'_k, \breve{g}_k, \breve{g}'_k$ the absolute difference between the original returns $\Delta = |g'_k - g_k|$ in case action trend was not preserved on this realization. Meaning, at this realization, the optimal actions of original and simplified problems would differ. Given a sample $(g_k, g'_k, \breve{g}_k, \breve{g}'_k)$, the simplification is action consistent at this sample if the sign of the difference of the returns is preserved. In other words, the same action would be identified

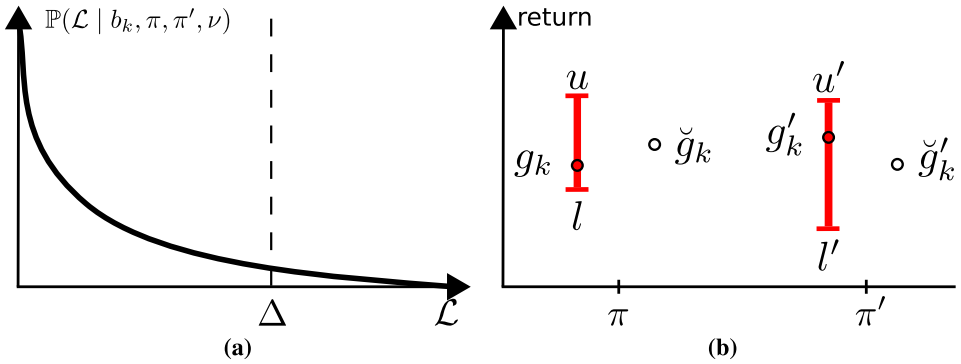


Fig. 6. Illustration of **(a)** the distribution of loss, and **(b)** the online bounds of the return.

as optimal with the original and simplified returns; else we must account for the loss (49). Our object of interest is the distribution density of $\mathcal{L}$ given all the information available at our disposal,

$$\mathbb{P}\left(\mathcal{L}|b_k, \pi, \pi', \nu\right). \tag{50}$$

We denote this distribution by Probabilistic Loss (`PLoss`). See illustration in Fig. 6a. E.g., if (50) is the Dirac delta function $\delta(\mathcal{L})$, the simplification method is absolute action consistent for every possible objective operator φ. Moreover, for any Δ, its CDF $\mathbb{P}\left(\mathcal{L} \leq \Delta|b_k, \pi, \pi', \nu\right)$ provides probability to suffer loss at most Δ. Similarly, the Tail Distribution Function (TDF) $\mathbb{P}\left(\mathcal{L} > \Delta|b_k, \pi, \pi', \nu\right)$ provides probability to suffer loss greater than Δ. The source of distribution (50) is (44).

5.3.1. Online bound on probabilistic loss (`PbLoss`)

The distribution defined by (50) requires access to (44) which we do not have in an online setting. To circumvent the requirement of accessing g_k and g'_k, we propose to substitute them by online lower and upper bounds l, u and l', u', respectively. These bounds should be accessible without knowledge of original returns. Similar to section 3.3 we aim to bound each original return corresponding to its candidate policy.

Let us consider a sampled return realization $(g_k, g'_k, \breve{g}_k, \breve{g}'_k)$ from (44). As in an online setting we do not actually have access to the original returns (g_k, g'_k), we strive to bound the latter,

$$l \leq g_k \leq u,\ l' \leq g'_k \leq u', \tag{51}$$

where, for now, we assume (51) holds for any sample of $(g_k, g'_k, \breve{g}_k, \breve{g}'_k)$; for example, these could be analytically-derived bounds. This setting is illustrated in Fig. 6b. However, further we also discuss a more general setting where we allow (21) to be violated with probability larger than zero.

Using these bounds we are able to define online a bound on loss (49) *without* accessing the original problem (R and B),

$$\bar{\mathcal{L}} \triangleq f_{\bar{\mathcal{L}}}(\breve{g}_k, l, u, \breve{g}'_k, l', u') = \begin{cases} \max\{u'-l, 0\} & \text{if } \breve{g}_k > \breve{g}'_k, \\ \max\{u-l', 0\} & \text{if } \breve{g}_k < \breve{g}'_k. \end{cases} \tag{52}$$

Note that sometimes we can find bounds over the returns by applying the same function f_{g_k} on the bounds on the momentary rewards (returns when $L = 1$), e.g., in case of cumulative reward $u = \sum_{\ell=k+1}^{k+L} u_\ell$ and $l = \sum_{\ell=k+1}^{k+L} l_\ell$. However, this is not always possible, e.g., if g_k deviates from the sum of momentary rewards or in the case of Bellman form of the objective. Sometimes it is, therefore, better to work with momentary bounds.

In an online setting, we are interested in the distribution density of $\bar{\mathcal{L}}$,

$$\mathbb{P}\left(\bar{\mathcal{L}}|b_k, \pi, \pi', \nu\right), \tag{53}$$

which we denote by Probabilistic Bound on Loss (`PbLoss`).

As we discuss in Section 5.3.2, `PbLoss` characterizes the impact of a simplification in an online setting; thus, it enables to determine online if a candidate simplification is acceptable given a user-specified criteria. The decision to either accept or decline a (candidate) simplification is guided by probabilistic guarantees, as provided by our approach.

5.3.2. Description of `PLoss` online

In this section, we show how `PbLoss` can be used in an online setting to characterize `PLoss` (which is unavailable online). In turn, this enables us to provide online probabilistic performance guarantees for a considered simplification (represented by operator ν), or to decide if it is adequate given a user-specified criteria.

Specifically, recall `PLoss` CDF and TDF, i.e., probability to suffer loss at most, or greater, than $\Delta \in \mathbb{R}$, respectively,

$$\texttt{PLoss CDF: } P\left(\mathcal{L} \leq \Delta|b_k, \pi, \pi', \nu\right) \tag{54}$$

$$\texttt{PLoss TDF: } P\left(\mathcal{L} > \Delta|b_k, \pi, \pi', \nu\right). \tag{55}$$

We now aim to bound PLoss CDF (54) from below, and PLoss TDF (55) from above by utilizing PbLoss.

We now consider PLoss TDF and express $P\left(\mathcal{L} > \Delta | b_k, \pi, \pi', \nu\right)$ as

$$P\left(\mathcal{L} > \Delta, \bar{\mathcal{L}} \geq \mathcal{L} | b_k, \pi, \pi', \nu\right) + P\left(\mathcal{L} > \Delta, \bar{\mathcal{L}} < \mathcal{L} | b_k, \pi, \pi', \nu\right).$$

The first term can be written via chain rule as

$$P\left(\mathcal{L} > \Delta | \bar{\mathcal{L}} \geq \mathcal{L}, b_k, \pi, \pi', \nu\right) P\left(\bar{\mathcal{L}} \geq \mathcal{L} | b_k, \pi, \pi', \nu\right). \tag{56}$$

Performing chain rule similarly also on the second term and recalling that $P(\bar{\mathcal{L}} \geq \mathcal{L}|\cdot) + P(\bar{\mathcal{L}} < \mathcal{L}|\cdot) = 1$, allows to express PLoss TDF as

$$P\left(\mathcal{L} > \Delta | \bar{\mathcal{L}} \geq \mathcal{L}, b_k, \pi, \pi', \nu\right) \lambda + P\left(\mathcal{L} > \Delta | \bar{\mathcal{L}} < \mathcal{L}, b_k, \pi, \pi', \nu\right) (1 - \lambda), \tag{57}$$

where

$$\lambda \triangleq P\left(\bar{\mathcal{L}} \geq \mathcal{L} | b_k, \pi, \pi', \nu\right) \equiv P\left(\mathbf{1}_{\{\bar{\mathcal{L}} \geq \mathcal{L}\}} = 1 | b_k, \pi, \pi', \nu\right). \tag{58}$$

While λ from (58) is unavailable, we can bound it from below using

$$1 - \alpha \leq P\left(\mathbf{1}_{\{l \leq g_k \leq u\}} = 1 | \mathcal{H}_{k+L}, \nu\right) \tag{59}$$

and

$$1 - \alpha \leq P\left(\mathbf{1}_{\{l' \leq g'_k \leq u'\}} = 1 | \mathcal{H}'_{k+L}, \nu\right) \tag{60}$$

and

Theorem 3 *(Probability that bound bounds). Fix $\alpha \in \mathbb{R}$. Assume that (59) and (60) hold. Then:*

$$P\left(\mathbf{1}_{\{\bar{\mathcal{L}} \geq \mathcal{L}\}} = 1 | b_k, \pi, \pi', \nu\right) \geq (1 - \alpha)^2. \tag{61}$$

We provide the detailed proof in Appendix A.3. Now we show that given the event $\{\bar{\mathcal{L}} \geq \mathcal{L}\}$, PLoss TDF is bounded from above by PbLoss TDF. It is clear that $\forall \Delta \in \mathbb{R}$,

$$P\left(\mathcal{L} > \Delta | \mathbf{1}_{\{\bar{\mathcal{L}} \geq \mathcal{L}\}} = 1, b_k, \pi, \pi', \nu\right) \leq P\left(\bar{\mathcal{L}} > \Delta | \mathbf{1}_{\{\bar{\mathcal{L}} \geq \mathcal{L}\}} = 1, b_k, \pi, \pi', \nu\right). \tag{62}$$

Finally, we characterize PLoss as follows.

Theorem 4 *(Upper and Lower bounds). Denote*

$$\theta_\alpha(\Delta) \triangleq \min\left\{1, \frac{P\left(\bar{\mathcal{L}} > \Delta | b_k, \pi, \pi', \nu\right)}{(1 - \alpha)^2} + 2\alpha - \alpha^2\right\},$$

so

$$P\left(\mathcal{L} > \Delta | b_k, \pi, \pi', \nu\right) \leq \theta_\alpha(\Delta) \quad \forall \Delta \in \mathbb{R}_{\geq 0} \tag{63}$$

and

$$P\left(\mathcal{L} \leq \Delta | b_k, \pi, \pi', \nu\right) \geq 1 - \theta_\alpha(\Delta) \quad \forall \Delta \in \mathbb{R}_{\geq 0}. \tag{64}$$

The full proof can be found in Appendix A.4. With accessible online $\theta_\alpha(\Delta)$ we are able to obtain a complete characterization of the simplification. Moreover, since $0 \leq \mathcal{L}$, setting $\Delta = 0$ in Algorithm 3 we can assess the probability to be absolute action consistent for any φ.

Algorithm 3 Online empirical characterization of the PLoss with PbLoss.

Input: Two candidate policies π, π'. Initial belief b_k. Samplers from $\mathbb{P}(\breve{g}_k, l, u|\mathcal{H}_{k+L}, \nu)$ and $\mathbb{P}(\breve{g}'_k, l', u'|\mathcal{H}'_{k+L}, \nu)$.

Sample b_k or take the initial samples from inference. Obtain s samples from $\mathbb{P}\left(z_{k+1:k+L}, z'_{k+1:k+L}|b_k, \pi, \pi'\right)$ and create two belief policy trees.

for sample pairs $(z_{k+1:k+L}, z'_{k+1:k+L})$ **do**
  Obtain sample $(\breve{g}_k, l, u, \breve{g}'_k, l', u')$.
  Calculate $f_{\bar{\mathcal{L}}}(\breve{g}_k, l, u, \breve{g}'_k, l', u')$ according to (52).
end for

$\{f_{\bar{\mathcal{L}}}(\breve{g}_k, l, u, \breve{g}'_k, l', u')\}$ represents the set of samples of $\bar{\mathcal{L}}$.

Output: $\forall\Delta$ empirically calculated $P\left(\bar{\mathcal{L}} > \Delta|b_k, \pi, \pi', \nu\right)$ as $\frac{\sum_{i=1}^{s}[\bar{\mathcal{L}}^i > \Delta]}{s}$.

5.3.3. Calculating PLoss offline and PbLoss online

One approach to obtain PLoss *offline* is to sample $(g_k, g'_k, \breve{g}_k, \breve{g}'_k)$ from (44) using decomposition (46). PLoss is then represented by $\{f_{\mathcal{L}}(g_k, g'_k, \breve{g}_k, \breve{g}'_k)\}$.

We take samples of $\mathbb{P}(z_{k+}, z'_{k+}|b_k, \pi, \pi')$ from the corresponding extended belief policy trees. To sample

$$\mathbb{P}(g_k, \breve{g}_k|\mathcal{H}_{k+L}, \nu) \quad , \quad \mathbb{P}(g'_k, \breve{g}'_k|\mathcal{H}'_{k+L}, \nu), \tag{65}$$

we use the original (not simplified) rewards calculated from the beliefs present at the belief tree (belief tree does not undergo simplification) and their simplified counterparts.

So far, we did not explain how to calculate PbLoss (53). One approach is to sample $(\breve{g}_k, l, u, \breve{g}'_k, l', u')$ from

$$\mathbb{P}(\breve{g}_k, l, u, \breve{g}'_k, l', u'|b_k, \pi, \pi', \nu) \tag{66}$$

and evaluate $\bar{\mathcal{L}}$ for each such sample via (52). Then, PbLoss is represented by $\{f_{\bar{\mathcal{L}}}(\breve{g}_k, l, u, \breve{g}'_k, l', u')\}$.

Generating samples from (66) involves marginalizing over future measurements $z_{k+} \equiv z_{k+1:k+L}$ and $z'_{k+} \equiv z'_{k+1:k+L}$. Similar to (46), the (66) decomposes to

$$\int_{\substack{z_{k+}\\ z'_{k+}}} \mathbb{P}(\breve{g}_k, l, u|\mathcal{H}_{k+L}, \nu)\mathbb{P}(\breve{g}'_k, l', u'|\mathcal{H}'_{k+L}, \nu)\mathbb{P}(z_{k+}, z'_{k+}|b_k, \pi, \pi')dz_{k+}dz'_{k+} \tag{67}$$

In practice, $\mathbb{P}(z_{k+}, z'_{k+}|b_k, \pi, \pi')$ corresponds to two extended belief policy trees, starting from the same root (b_k) and having the same rule for choosing rollouts. The specific way of obtaining samples from

$$\mathbb{P}(\breve{g}_k, l, u|\mathcal{H}_{k+L}, \nu) \quad , \quad \mathbb{P}(\breve{g}'_k, l', u'|\mathcal{H}'_{k+L}, \nu) \tag{68}$$

depends on the operator ν. We summarized the proposed approach in Algorithm 3. In the next section, we elaborate on these aspects, considering a specific simplification operator.

5.4. Guarantees on simplified hypothesis based decision making

In this section we provide guarantees on the simplification portrayed by (48). We recite that the concept of PLoss and PbLoss is valid for any objective operator φ. In this section we describe a specific usage for the objective operator presented in section 5.2. Let us make a following definition

$$\breve{\Delta}^P \triangleq \left|\sum_{i=1}^{s} \mathbf{1}\{\breve{g}_k^i > \breve{g}_k'^i\} - \sum_{i=1}^{s} \mathbf{1}\{\breve{g}_k^i < \breve{g}_k'^i\}\right|. \tag{69}$$

Each not-action consistent sample decreases this margin by 2, so in order to be not-action consistent we need to satisfy

$$2 \cdot \sum_{i=1}^{s} \mathbf{1}\{\mathcal{L}^i > 0\} \geq \breve{\Delta}^P. \tag{70}$$

The following relation permits us to answer the question either or not it is possible that the order is switched due to simplification.

$$\sum_{i=1}^{s} \mathbf{1}\{\mathcal{L}^i > 0\} = s \cdot P(\mathcal{L} > 0|b_k, \pi, \pi', \nu). \tag{71}$$

The offline condition that simplification is action consistent will be

$$2sP(\mathcal{L} > 0|b_k, \pi, \pi', \nu) < \breve{\Delta}^P. \tag{72}$$

From here we can define the *online* accessible condition that simplification is action consistent as

$$2 \cdot \sum_{i=1}^{s} \mathbf{1}\{\bar{\mathcal{L}}^i > 0\} < \breve{\Delta}^P. \tag{73}$$

We observe that the above relation involves `PbLoss` from Algorithm 3. Remarkably, for analytical or stochastic bounds for sufficiently large s so (59) and (60) hold, we obtain that the condition to be action consistent is

$$2s\theta_\alpha(0) < \breve{\Delta}^P. \tag{74}$$

Noteworthy, similar to VaR we provide deterministic guarantees using stochastic bounds.

6. Specific simplification

In this section, we exemplify our technique on a specific simplification method. Let us recite that if the simplification regime acts according to (17), it results in uncorrelated $\breve{g}_k|b_k, \pi, z_{k+}, \nu$ and $g_k|b_k, \pi, z_{k+}$. Conversely, if the simplification strategy complies to (18), the correlation is present. We start from the setting of a given belief surface and continue to the general nonparametric setting. In the following section we describe adaptive stochastic bounds in the setting of an explicitly given belief surface.

6.1. Online adaptive bounds on sample based return with a given belief surface

Let us start from the scenario in which the belief surface is explicitly given and we are interested in negative differential entropy as a belief dependent reward. Assume the belief is represented in closed form as a Gaussian mixture such that we have a deterministic update ψ_{dt} (see e.g. [25]). Since the differential entropy doesn't have a closed form solution, we are obliged to sample from the corresponding posterior belief. Assume we have n i.i.d. samples. One way to approximate the desired reward [10] is

$$I = \mathbb{E}\left[\ln(b(x))\right] \approx -\hat{\mathcal{H}} = \frac{1}{n}\sum_{i=1}^{n} \ln(b(x^i)). \tag{75}$$

We refer to (75) as a simplified reward. This estimator is unbiased as

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\ln(b(x^i))\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\ln(b(x^i))\right] \underbrace{=}_{\forall i\ x_i \sim b} \mathbb{E}\left[\ln(b(x))\right] = I. \tag{76}$$

Assume that b_k is a sampleable surface and

$$g_k|\mathcal{H}_{k+L} = f_{g_k}((\mathbb{E}\left[\ln(b_\ell(x_\ell))\right])_{\ell=k+1}^{k+L}) \tag{77}$$

$$\breve{g}_k|\mathcal{H}_{k+L}, \nu = f_{g_k}\left(\left(\frac{1}{n}\sum_{i=1}^{n}\ln(b_\ell(x_\ell^i))\right)_{\ell=k+1}^{k+L}\right) \tag{78}$$

Note that $g_k|\mathcal{H}_{k+L}$ is theoretical at this point. Its distribution is Dirac delta and its variance is zero. In case we have a belief surface represented as a Gaussian mixture with M components the complexity of calculating such simplified return will be $O(n \cdot M)$, where n is the number of samples from the surface.

Using a standard Gaussian confidence interval [29] we obtain

$$P\left(|g_k - \breve{g}_k| \le z_{\alpha/2}\text{se}(n)|\mathcal{H}_{k+L}, \nu\right) \approx 1 - \alpha. \tag{79}$$

Adaptive stochastic bounds Let us focus on variance of the reward. Assume that f_{g_k} is of the following form.

$$f_{g_k}(\rho_{k+1:k+L}) = \frac{1}{L}\sum_{\ell=k+1}^{k+L}\rho_\ell. \tag{80}$$

Denote by 1 column vector of ones. The variance $\mathbb{V}\left(\frac{1}{L}\sum_{\ell=k+1}^{k+L}\breve{\rho}_\ell \middle| b_k, \mathcal{H}_{k+L}, \nu\right)$ can be written as

$$\mathbb{V}\left(\frac{1}{L}1^T\breve{\rho}_{k+1:k+L}\middle| b_k, \mathcal{H}_{k+L}, \nu\right) = \frac{1^T\Sigma 1}{L^2} \le \max_i \sigma_{ii}^2. \tag{81}$$

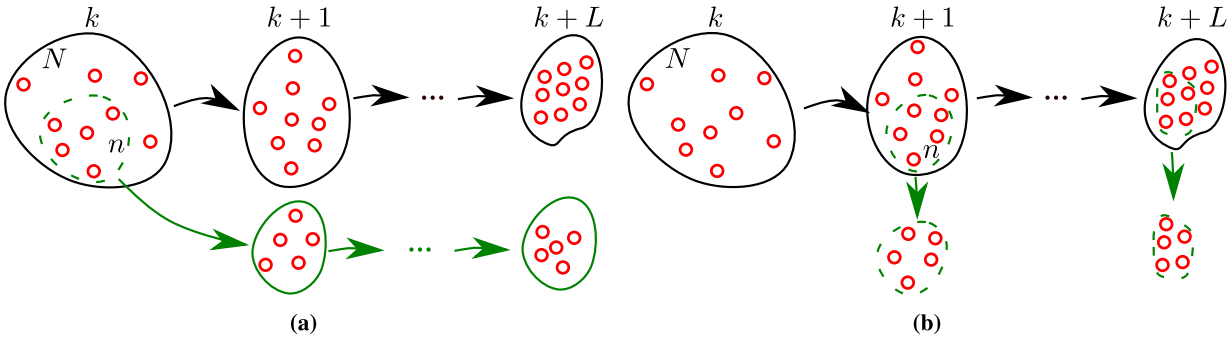


Fig. 7. Potential simplification techniques: **(a)** Choosing a subset of samples only at time k and updating the simplified belief. Such a simplification corresponds to (17).; **(b)** Choosing a subset of samples at each time ℓ and updating the original belief. Such a simplification corresponds to (18).

From now let us focus on the variance of $\check{\rho}_\ell|b_k, \mathcal{H}_{k+L}, \nu$, which is (the samples from the belief surface are i.i.d.)

$$\mathbb{V}\left[\frac{1}{n}\sum_{i=1}^{n}\ln(b_\ell(x^i))|b_k, \mathcal{H}_{k+\ell}, \nu\right] = \frac{1}{n}\mathbb{V}_{x\sim b_\ell}\left[\ln(b_\ell(x))|b_k, \mathcal{H}_{k+\ell}, \nu\right]. \tag{82}$$

If we knew how to update incrementally $\mathbb{V}_{x\sim b}[\ln(b(x))]$, this would yield adaptive stochastic bounds. We have samples from b, so we can calculate sample variance using n samples of the belief as

$$\hat{\mathbb{V}}_{x\sim b_\ell}\left[\ln(b_\ell(x))|b_k, \mathcal{H}_{k+\ell}, \nu\right] = \frac{1}{n-1}\left(\sum_{i=1}^{n}\ln^2(b_\ell(x^i)) - \left(\frac{1}{n}\sum_{i=1}^{n}\ln(b_\ell(x^i))\right)^2\right). \tag{83}$$

Alternatively, using Taylor expansion similar to [15] we obtain an approximation for desired variance

$$\mathbb{V}_{x\sim b_\ell}\left[\ln(b_\ell(x))\right] = \mathbb{E}[\ln^2(b_\ell(x))] - \mathbb{E}^2[\ln(b_\ell(x))]. \tag{84}$$

Suppose we use sample variance. It has readily available incremental update using Welford's online algorithm. Suppose we have calculated $\mathrm{se}^2_\ell(n)$ and $\check{g}^n_k = \frac{1}{L}\sum_{\ell=k+1}^{k+L}\mu^n_\ell$, now we want to tighten the bounds. We sample point $x_\ell \sim b_\ell$ from each surface $\ell = k+1 : k+L$. Firstly we update the simplified return incrementally

$$\check{g}^{n+1}_k = \frac{1}{L}\sum_{\ell=k+1}^{k+L}\left(\mu^n_\ell + \frac{\ln b(x_\ell) - \mu^n_\ell}{n}\right) = \check{g}^n_k + \frac{1}{n}\left(\frac{1}{L}\sum_{\ell=k+1}^{k+L}\ln b(x_\ell) - \check{g}^n_k\right). \tag{85}$$

We then update the $\mathrm{se}^2_\ell(n)$ towards $\mathrm{se}^2_\ell(n+1)$. Again the incremental update is readily available

$$(n+1)\cdot\mathrm{se}^2_\ell(n+1) = n\cdot\mathrm{se}^2_\ell(n) + (\ln b(x_\ell) - \mu^n_\ell)(\ln b(x_\ell) - \mu^{n+1}_\ell). \tag{86}$$

We will have to bookkeep μ^n_ℓ.

Unfortunately, the belief surface is not always obtainable. Moreover, not always not-simplified reward has zero variance. To these aspects we devote the next section.

6.2. Online bounds on sample based return - general setting

Suppose we are given from the inference stage a belief represented by a set of N weighted particles $b_k = \{w^i_k, x^i_k\}_{i=1}^N$. We would like to simplify planning by taking substantially less particles $\check{b}_k = \{w^j_k, x^j_k\}_{j=1}^n$. Alternatively we subsample the original belief b_ℓ at each time index to obtain $\check{b}_\ell = \{w^j_\ell, x^j_\ell\}_{j=1}^n$. Our simplification operator ν provides a way to choose a subset of n samples from the original N samples. For example, subsampling according to weights. We take ψ_{st} as an off-the-shelf particle filter, which produces the same number of samples as the input. The two ways of updating the belief are illustrated in Fig. 7. To present development for (25), we continue with unbiasedness assumption and take an inspiration from confidence intervals. Let us introduce the following model

$$\begin{pmatrix} g_k \\ \check{g}_k \end{pmatrix}|\mathcal{H}_{k+L}, \nu \sim \mathcal{N}\left(\begin{pmatrix}\mu \\ \mu\end{pmatrix}; \begin{pmatrix}\mathrm{se}^2(N) & \mathrm{cov} \\ \mathrm{cov} & \mathrm{se}^2(n)\end{pmatrix}\right), \tag{87}$$

where se is the standard error and cov is the covariance. Online we do not have access to these quantities. The standard error depends on the number of samples N and n respectively, dwindling as the number of samples increases. We assume that each marginal is distributed around the same mean value μ (no bias).

Denote $y = g_k - \breve{g}_k$. It is known that y is a zero mean Gaussian with the following variance

$$\text{var}(y) = \text{se}^2(N) + \text{se}^2(n) - 2\text{cov}. \tag{88}$$

Let $z = \frac{y}{\sqrt{\text{var}(y)}} \sim \mathcal{N}(0,1)$ and $z_{\alpha/2} = \Phi^{-1}(1-\alpha/2)$, where Φ is a CDF of a standard normal variable so $\mathbb{P}(z > z_{\alpha/2}) = \alpha/2$ and

$$P\left(-z_{\alpha/2} \leq z \leq z_{\alpha/2}\right) = 1-\alpha. \tag{89}$$

In other words

$$P\left(|y| \leq z_{\alpha/2}\sqrt{\text{var}(y)} | \mathcal{H}_{k+L}, \nu\right) = 1-\alpha. \tag{90}$$

Using the facts $\text{se}(N) \leq \text{se}(n)$ and $\text{cov} \leq \text{se}(N)\text{se}(n)$ we arrive at two cases. The first case is the zero covariance ($\text{cov} = 0$).

$$\text{var}(y) = \text{se}^2(N) + \text{se}^2(n) \leq 2\text{se}^2(n). \tag{91}$$

The second case is more general.

$$\text{var}(y) = \text{se}^2(N) + \text{se}^2(n) - 2\text{cov} \leq 4\text{se}^2(n). \tag{92}$$

Thus, from (90) we obtain for both cases

$$P\left(|g_k - \breve{g}_k| \leq z_{\alpha/2}\sqrt{2}\text{se}(n) | \mathcal{H}_{k+L}, \nu\right) \geq 1-\alpha. \tag{93}$$

$$P\left(|g_k - \breve{g}_k| \leq z_{\alpha/2}2\text{se}(n) | \mathcal{H}_{k+L}, \nu\right) \geq 1-\alpha. \tag{94}$$

6.2.1. Comparison to baseline methods

As a baseline we take conventional methods applying ψ_{st} once and treat the sample obtained as representative.

Sample mean Let us assume that the objective is the sample mean of the return with one sample of return per observation and $\{z_{k+1:k+L}\}_{i=1}^{s}$ samples of observations. Suppose that samples of observations are i.i.d. Let us recall that the variance of this sample mean is as follows

$$\mathbb{V}\left(\frac{1}{s}\sum_{i=1}^{s} g_k^i\right) = \frac{1}{s}\left(\mathbb{E}_{z+}[\text{se}^2(z_{k+}, N)] + \mathbb{V}(\mu(z_{k+}))\right). \tag{95}$$

With analytical bounds we bound deterministically every sample g_k^i so we bound sample mean. However in case of stochastic bounds we can not bound expected value but we can use samples of simplified return instead original. Under the model (87) the expected value of the same sample mean will stay the same however the variance of the estimator will grow. In this case we will accelerate decision making but we will pay with increasing variance

$$\mathbb{V}\left(\frac{1}{s}\sum_{i=1}^{s} \breve{g}_k^i\right) = \frac{1}{s}\left(\mathbb{E}_{z+}[\text{se}^2(z_{k+}, n)] + \mathbb{V}(\mu(z_{k+}))\right). \tag{96}$$

We believe that this is an interesting relation.

Sample Value at Risk When the objective is sample approximation of VaR with analytical bounds as well with stochastic bounds, we can bound only the theoretical VaR.

6.2.2. Estimation of the variance

As we do not have access to $\text{se}(n)$ in (93) and (94), it has to be estimated. The simplest way to do that is to repeatedly sample simplified returns m times from one of (17), (18), (20) depending on the simplification type. Note that a possible bias of the particle filter and the estimation of standard error make (90) only asymptotically correct. However, when dealing with a sufficient amount of samples N and n, these deviations from (87) are negligible. Even with repeated re-sampling we will reduce computational complexity, as we analyze in Section 7. The bounds for both simplification methods are

$$(\text{no cov})\ u = \breve{g}_k + z_{\alpha/2}\sqrt{2}\hat{\text{se}}_m \quad l = \breve{g}_k - z_{\alpha/2}\sqrt{2}\hat{\text{se}}_m, \tag{97}$$

$$(\text{cov})\ u = \breve{g}_k + z_{\alpha/2}2\hat{\text{se}}_m \quad l = \breve{g}_k - z_{\alpha/2}2\hat{\text{se}}_m. \tag{98}$$

Moreover, since we recalculate the simplified reward m times, we could improve the final simplified return. In this case, we take the average of the samples of the simplified return given the history (prior belief, candidate policy, and the realization of the observations) as a final simplified return for this history

$$\breve{g}_k = \frac{1}{m}\sum_{j=1}^{m} \breve{g}_k^j \tag{99}$$

and the model becomes

$$g_k | \mathcal{H}_{k+L}, \nu \sim \mathcal{N}(\breve{g}_k, \mathrm{se}^2(n)). \tag{100}$$

The bounds in this case are

$$u = \breve{g}_k + z_{\alpha/2}\hat{\mathrm{se}}_m \quad l = \breve{g}_k - z_{\alpha/2}\hat{\mathrm{se}}_m. \tag{101}$$

These bounds asymptotically hold with probability at least $1-\alpha$.

Using $\breve{g}_k^i = \frac{1}{m}\sum_{j=1}^{m} \breve{g}_k^j$, where $i = 1:s$ we will obtain

$$\mathbb{V}\left(\frac{1}{s}\sum_{i=1}^{s} \breve{g}_k^i\right) = \frac{1}{s}\left(\frac{1}{m}\mathbb{E}_{z+}[\mathrm{se}^2(z_{k+}, n)] + \mathbb{V}(\mu(z_{k+}))\right). \tag{102}$$

6.3. Implementation details and computational complexity

Now we describe steps in building an extended belief tree which is common for all our simulations. First, we need to construct an extended belief tree appropriate to a given candidate policy (see Fig. 1); alternatively, if the objective operator is mounted on the joint distribution of a pair of returns given a pair of policies, as in (43), we shall construct a pair of coupled belief trees. Second, we shall apply the simplification and calculate simplified returns and bounds. In all simulations with nonparametric beliefs we choose ψ_{st} to be an off-the-shelf particle filter with low-variance re-sampling [42]. The entire belief update process complexity is $\mathcal{O}(N)$. Since the extended belief tree does not undergo simplification, it is common to the original and simplified problems.

In practice, the marginal likelihoods $\mathbb{P}(z_{k+}|b_k, \pi)$ and $\mathbb{P}(z'_{k+}|b_k, \pi')$ as in section 4 or the mutual likelihood of the observations $\mathbb{P}(z_{k+}, z'_{k+}|b_k, \pi, \pi')$ as in section 5 (see (41)) correspond to two extended belief policy trees, starting from the same root (b_k) and having the same rule for choosing rollouts.

Below we discuss the construction of the extended belief tree. Let N be a number of samples of the posterior belief. We choose the samples of the belief for creating the observations heuristically according to the following scheme. Let $n_z^{(\ell)}$ be number of observations generated by each belief at level ℓ of the tree. We specify $n_z^{(1)}$ (the number of observations generated by b_k) and the dwindle factor c. Starting from $\ell = 2$ the number of observations generated by each belief on level ℓ in the tree is calculated as $n_z^{(\ell)} = \max\{1, \lfloor \frac{n_z^{(1)}}{(\ell-1)\cdot c} \rfloor\}$. In the setting of nonparametric beliefs, we sample states for the observations from resampled posterior with Fisher-Yates shuffling (with early termination) [20]. This algorithm is $\mathcal{O}(N)$ for initialization, plus $\mathcal{O}(n_z^{(\ell)})$ for random shuffling.

In our extended belief policy tree, there may be many beliefs stemming from an observation. Denote this number by n_b. In the setting of nonparametric beliefs represented by the particles, the complexity of constructing the tree is

$$\mathcal{O}(N)\sum_{\ell=1}^{L-1}\prod_{i=1}^{\ell} n_b n_z^{(i)}. \tag{103}$$

At each level of the tree beside the bottom, we must apply a particle filter number of times equal to the total number of the beliefs at the next level, which is $\prod_{i=1}^{\ell} n_b n_z^{(i)}$ at level ℓ. Also, we need to subsample observations at the current level. Since the number of beliefs at the next level is not smaller than at the current level, and the subsampler and particle filter complexity is linear in N, we are left with (103). Let us mention that sampling from the belief and application of particle filters on each level can be done in parallel.

Now we analyze the speedup in running time as a result of simplification in the setting of nonparametric beliefs. As a momentary reward, we take the differential entropy estimator from [4]. This selection makes the complexity of calculating the momentary reward to be $\mathcal{O}(N^2)$. For the bounds calculation depending on the simplification method we need to apply particle filter with n samples (17) or with N (18), (20)) samples, L times for each return. Since its complexity is linear in the number of samples, the expected speedup is governed only by the immediate reward and bounds calculation. The speedup is approximately

$$\frac{N^2}{n^2 \cdot m}. \tag{104}$$

This acceleration has a rather intuitive explanation. Since we are comparing running time of exactly the same function the ratio gives approximately exact speedup. The empirical behavior of estimator is as $\Theta(N^2)$. We obtained this speedup in all our simulations.

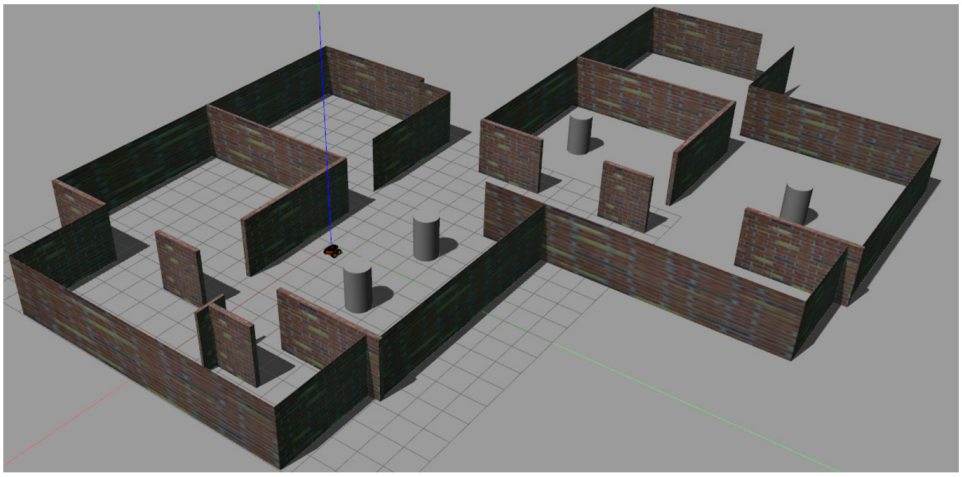


Fig. 8. Gazebo simulated environment. Each square in the map corresponds to 1×1 meters square.

7. Simulations and results - autonomous navigation with light beacons

In this section, we demonstrate our findings. In the center of our focus are the risk-averse operators, and in all cases, simplification yields a significant speedup without sacrificing the quality of the solution. We consider the setting of marginal distributions over returns per candidate policy, as in Section 4, and the joint distribution over the returns given a pair of policies, as in Section 5. In both settings, we consider the problem of autonomous navigation to a pre-defined goal in an environment with known beacons.

We start from marginals of the return in the setting of a given belief surface and then proceed to the general domain of nonparametric beliefs and an inaccessible belief surface. We then continue to the joint distribution of a pair of the returns given two policies and `PLoss` and `PbLoss` simulations. Finally, we report the technical characteristics of computers used in simulations in Appendix B.

7.1. Marginal return distributions corresponding to candidate polices

For our simulations, we utilize a localization problem with a known map created in the Gazebo simulator [23]. We used a Pioneer 3-AT robot to perform navigation to the goal as illustrated in Fig. 8.

7.1.1. A given belief surface

We start by exemplifying the adaptive bounds from Section 6.1 in the setting described in [25]. We do not assume that we know which beacon generated an observation in this setting. Instead, we maintain a hypothesis about each possible configuration of the beacons creating the observation. Such an approach is realistic since, in the planning phase, the robot considers the future observations identically as in the inference phase when the real observation is obtained. It results in belief being a Gaussian Mixture Model (GMM), where each component corresponds to a possible configuration of data association. The weights of the components are probabilities of the hypothesis that the Gaussian component is an actual configuration. No analytical expression exists for differential entropy when the belief surface is GMM, so we are obliged to sample and want as few samples as possible. For simplicity, we consider only two possible paths to the goal, as shown in Fig. 9. We take the belief b_k over of the robot's 2D location as a Gaussian $\mathcal{N}(\mu, \Sigma)$, where $\mu = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$. The belief update is deterministic ψ_{dt}. Each beacon is visible on the maximum radius of 3 meters. Leave out the recursive setting instead of smoothing, we strictly follow the theory presented at [25]. Let us restate that, in planning, when considering possible observations, we do not assume that we know from which beacon they arrived; instead, we maintain hypotheses regarding each possible configuration of the beacons to yield an observation. We denote each such data association by $\beta_{k+\ell}$.

In this study we utilize the following motion model T.

$$x_{k+1} = x_k + a_k + \|a_k\| \cdot w_k \quad w_k \sim \mathcal{N}(0, \Sigma_w), \tag{105}$$

where $x \in \mathbb{R}^2$, $a \in \mathbb{R}^2$, $\Sigma_w = w \cdot I$ (w is a given parameter) and action $a_k \in \mathcal{A}$. The $\mathcal{A}$ is the set of action sequences with actions of variable length. Each visible beacon b produces the observation according to the following model $z_i \sim \mathcal{N}(\|x - x^b\|_2^2, \Sigma_v)$, where $\Sigma_v = v \cdot I$. We selected the following parameters $w = 0.5$ and $v = 0.005$. Overall observation is the concatenation of the observations received from all seen beacons. Let us denote by M the given map with its beacons. For simulating an observation for planning we sample state $x_{k+\ell}$ from the belief propagated with an action. We use the simplest

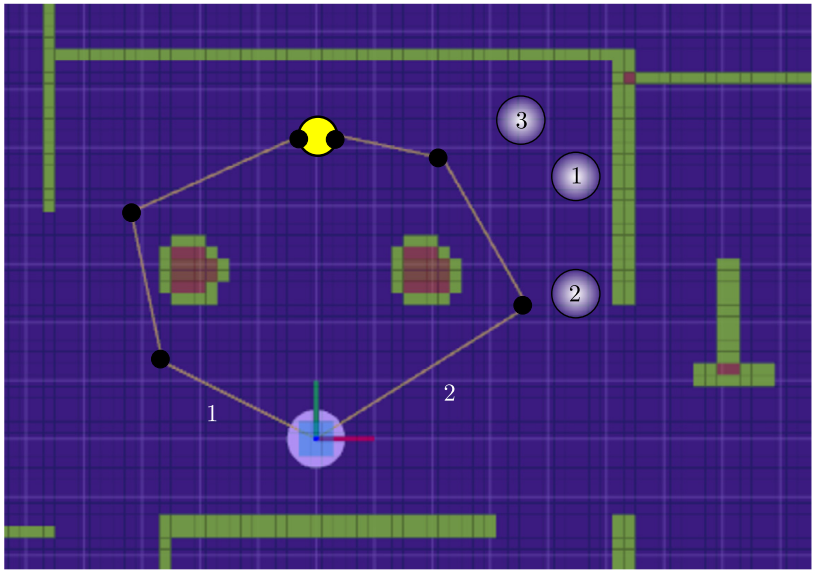


Fig. 9. Gazebo simulated scenario where the belief surface is explicitly given. The white numbers enumerate the paths, the black numbers enumerate the light beacons, the yellow circle is the goal, the black dots are the spots there the observations are taken by the robot, the purple circle is the initial belief b_k.

model for $\beta_{k+\ell}$ being $P_\beta(\beta_{k+\ell}(i)=1|x_{k+\ell}) = \mathbf{1}\{\|x_{k+\ell} - x_i^b\| \leq r\}$, where the index i goes from 1 until the number of beacons in the map. According to this model, given the state and the map, each beacon is deterministically seen or not. Once we obtained the configuration of the seen beacons $\beta_{k+\ell}$; we sample the observation from the following model as follows. We define subsequence $\beta_{k+\ell}(i_j)$ such that the index j pull in ascending order the indexes of i where $\beta_{k+\ell}(i_j)$ equal 1.

$$\mathbb{P}_Z(z_{k+\ell}|x_{k+\ell}, M) = \prod_{j=1}^{n_{k+\ell}(x_{k+\ell})} \mathbb{P}(z_{k+\ell}^j|x_{k+\ell}, M) = \prod_{j=1}^{n_{k+\ell}(x_{k+\ell})} \mathbb{P}(z_{k+\ell}^j|x_{k+\ell}, x_{i_j}^b), \tag{106}$$

where $n_{k+\ell}(x_{k+\ell})$ is the number of beacons seen from the state $x_{k+\ell}$. If no beacon is seen there is no observation received ($n_{k+\ell}(x_{k+\ell}) = 0$).

For belief update, however, we do not assume that we know this configuration $\beta_{k+\ell}$. The belief in each time instant is a Gaussian Mixture. To update the belief, we propagate each gaussian with an action using standard Kalman filter [42]. To update propagated Gaussian with an observation, we do not assume we know from which beacons this observation is received. This result to Gaussian mixture obtained from each propagated Gaussian, where each Gaussian in the mixture corresponds to the beacons configuration, which can render such an observation. We utilize unscented Kalman filter [42] to update each propagated Gaussian with the observation and a realization of the $\beta_{k+\ell}$. The weight of Gaussian corresponds to the probability that such beacons configuration resulted in the obtained observation. The above requires to model visibility of the beacon given the state from propagated Gaussian. Since $\beta_{k+\ell}$ is a discrete random variable, we normalize when all the above probabilities are computed. For an in-depth discussion, please refer to [25].

We sample 500 samples from propagated belief and set the parameter visibility radius as follows $r = 3$.

Let us recall that in this setting the original return is the theoretical differential entropy over the belief surface which is out of the reach. We want to set the number of samples from the belief surface n as small as possible to decide which path out of two brings less uncertainty. We aim to choose the path maximizing uncertainty criterion, which we define as follows

$$\varphi\left(\mathbb{P}\left(\breve{\rho}_{k+1:k+L}|b_k, \pi_{k:k+L-1}, \nu\right), \breve{g}_k\right) = \mathrm{VaR}_\beta(\breve{g}_k^I|b_k, \pi, \nu), \tag{107}$$

where

$$\breve{g}_k^I = I(b_{k+1:k+L}|b_k, \pi) = \frac{1}{L}\sum_{\ell=k+1}^{k+L}\left(\frac{1}{n}\sum_{i=1}^{n}\ln(b_\ell(x_\ell^i))\right). \tag{108}$$

We set $\alpha = 0.05$ such that $z_{\alpha/2} = 1.96$, and $\beta = 0.3$, overall number of observations in the belief tree is 500. Note that the condition $\alpha \cdot (2-\alpha) \leq \beta \leq 1-\alpha$ is fulfilled. We start from initial $n = 5$ and add one sample for each immediate reward in the belief tree until there is no overlap between the intervals. In this simulation the adaptation yielded not overlapping deterministic bounds on VaR when $n = 28$ and the second path was chosen as optimal. The interval for the first action sequence was $[-3.77, -2.76]$ and for the second sequence was $[-2.74, -1.52]$.

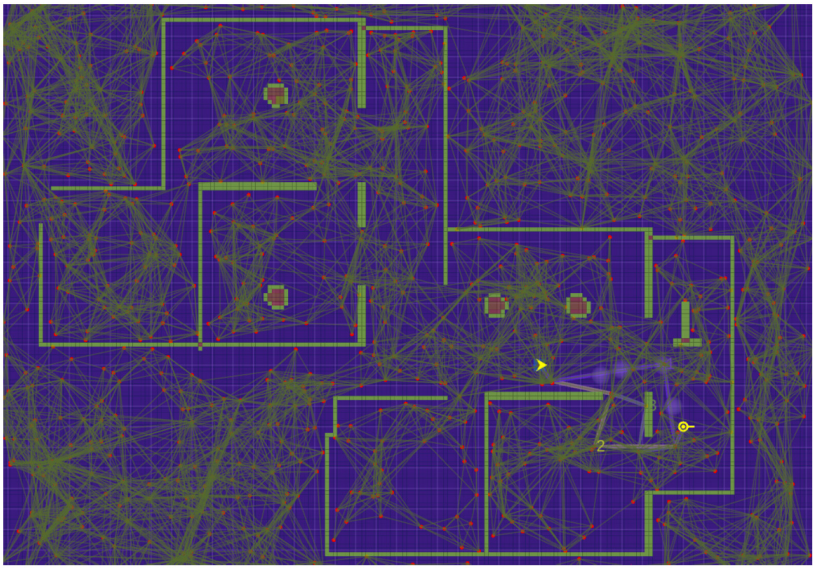


Fig. 10. Diverse short paths. The current robot position denoted yellow arrow-head and the goal marked by yellow circle. Candidate paths are enumerated. Transparent silver spheres are the light beacons.

Table 2
Running times for each of the three action sequences for $N = 2000$ and $n = 100$.

	a_1	a_2	a_3
g_k time [sec]	30178	23858	20664
$\check{g}_k$ time [sec]	85	64	57
l, u time [sec]	4084	3255	2805
speedup	7.24	7.18	7.22

7.1.2. Not accessible belief surface in the setting of nonparametric belief

In this section we consider non parametric beliefs represented by particles. The belief surface in this setting is out of the reach. We remain in recursive formulation of the two dimensional continuous state space. For updating the belief we use particle filter with low variance resampler [42]. We begin by building the Probabilistic Road Map (PRM) using OMPL library. After the map is built we apply the Diverse Short Path algorithm [43]. The resulting paths from robot to goal are visualized at Fig. 10. These paths constitute our action space. We normalize by path length L to obtain fair comparison. To accelerate the calculations we apply Algorithm 1.

We use same motion model as in the previous section. However, the observation model varies. In this scenario there are four beacons, but each beacon is always seen and produces the observation according to the following observation model O. $z^i_{k+\ell} \sim \mathcal{N}(x_{k+\ell}, \Sigma_v(x_{k+\ell}))$ for $i = 1, \dots 4$, where the spatially-varying covariance matrix is

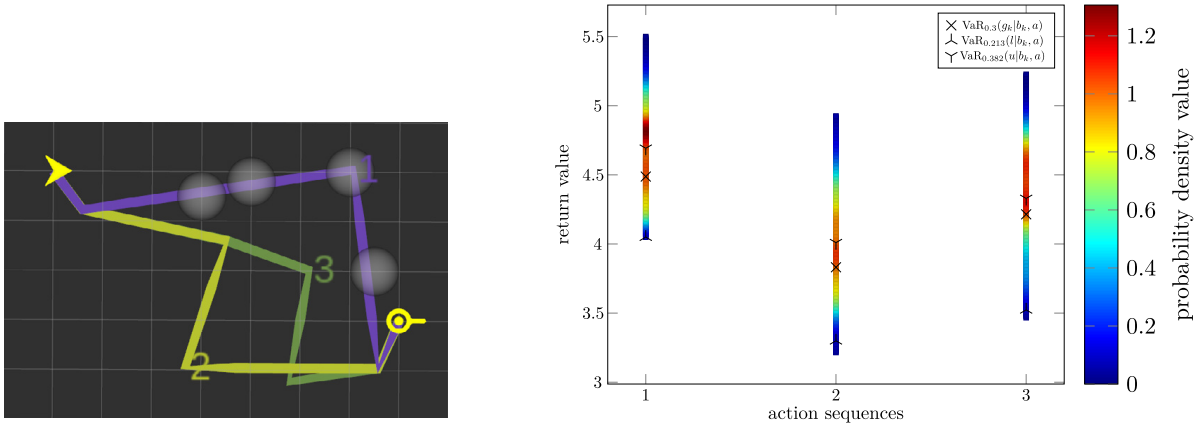
$$\Sigma_v(x) = v(x) \cdot I, \quad v(x) = w \cdot \|x - x^b_i\|^2_2, \tag{109}$$

where x^b_i is the location of the light beacon number i. The noise parameter w is taken from the motion model. In contrast to the previous section, we assume that the data association is solved. Overall observation received from all the beacons has the following probability density function

$$\mathbb{P}_Z(z|x) = \prod_{i=1}^{4} \mathbb{P}(z^i_{k+\ell}|x_{k+\ell}, x^b_i). \tag{110}$$

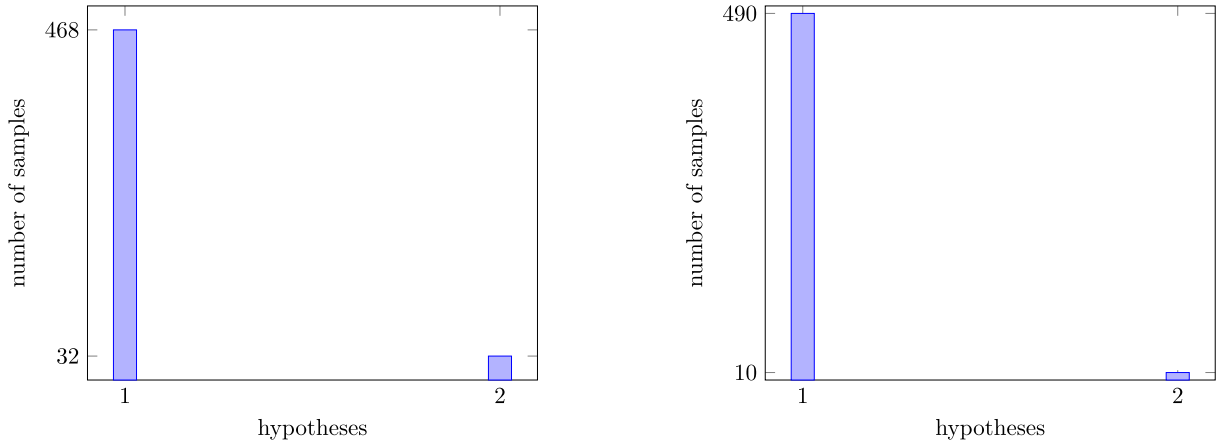
Without losing generality, we assume b_k at planning time is uniformly distributed in a unit square, such that the differential entropy is zero. In the naive approach to evaluate motion and observation models $\mathbb{P}_T$ and $\mathbb{P}_Z$ we need to inverse covariance matrix. Of course we can speedup this calculation by caching values of distribution of parameters or even value of evaluated motion model. However, this is out of scope of current discussion. We utilize an off-the-shelf Julia language implementation of Gaussian distribution. In general evaluation of the model can be extremely costly as described in [14].

We present results in Fig. 11b. Note that we chose $N = 2000$ guided by the works such as [4] and [11]. In this setting we obtain 8 times speedup according to (104), corroborated by Table 2, while the same optimal action is chosen as without simplification.



(a) **(b)**

Fig. 11. Simplified risk aware decision making using VaR. **(a)** Three candidate paths and four light beacons. **(b)** Results of simplified planning under uncertainty with $\beta = 0.3$. The first path is true optimal path due to its proximity to the beacons. In this scenario $w = 0.01$ and $v = 0.001$. The optimal path selected by solving the simplified problem is first and the relative error is zero whereas the online bound on the relative error is 0.07. In this simulation $N = 2000$ and $n = 100$. For calculation of the standard error for the bounds we recalculate the simplified reward $m = 50$ times.



(a) Original hypothesis based decision making

(b) Simplified hypothesis based decision making

Fig. 12. Comparison of the hypotheses for action sequence one and two. The total number of samples is 500.

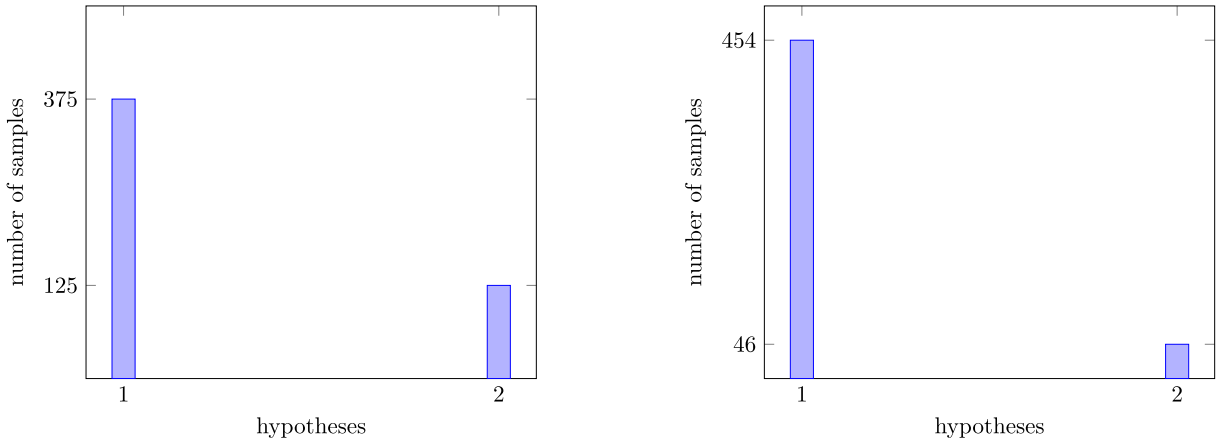
7.2. Joint distribution of the rewards corresponding to two candidate policies

In this section we exemplify simplified hypothesis based decision making outlined in section 5.2. We utilize the concept of PLoss to provide guarantees considering the specific objective from (48). Further we delve into PLoss to show the complete characterization of the simplification for any objective operator φ.

7.2.1. Simplified hypothesis based decision making

Let us focus on the previous scenario shown in Fig. 10. Our setting is as in previous section $N = 2000$, $n = 100$, $m = 50$. We start by comparing the first path to the second and show the results in Fig. 12. The first hypothesis is that the first action sequence is better and the second hypothesis is that the second action sequence is better. Remarkably, we observe that the simplification actually improves the decision making since more samples fall into the first hypothesis, and as we below show the first hypothesis is indeed optimal. We now utilize PbLoss at $\Delta = 0$ to provide deterministic guarantees. Continuing the discussion on Fig. 12 we obtain $\breve{\Delta}^P = 480$. The sample approximation $P(\mathcal{L} > 0|b_k, \pi, \pi', \nu) = 0.166$, such that the offline condition (72) is met $44 < 480$. The online bound on PLoss TDF at $\Delta = 0$ is $\theta_\alpha(0) = 0.47$ such that the online condition (74) is also met as $472 < 480$. Therefore, we can guarantee deterministically that the actions trend is preserved as a result of the simplification.

According to Algorithm 2 we shall also compare the first and the third paths. We present the comparison in Fig. 13. In this experiment we obtain $\breve{\Delta}^P = 408$, $P(\mathcal{L} > 0|b_k, \pi, \pi', \nu) = 0.166$, $\theta_\alpha(0) = 0.83$, such that the offline condition is fulfilled while the online condition is violated. To conclude, we were able to provide online guarantees that the simplification was action consistent when we compared the first and the second path. In some cases, as the comparison of the first and the third path, we cannot guarantee action consistency. Meaning, in this case, there is room to take more samples. It is even more interesting to utilize an incremental approach and develop adaptive stochastic bounds in the setting of nonparametric



(a) Original hypothesis based decision making

(b) Simplified hypothesis based decision making

Fig. 13. Comparison of the hypotheses for action sequence one and three. The total number of samples is 500.

beliefs. This is, however, out of the scope of this paper, and we leave it to further research. Alternatively, the analytical adaptive bounds from [39] can be used.

7.2.2. Probabilistic loss

We believe that the concept of PLoss and PbLoss can provide much more than showed in the previous section. PLoss characterizes the simplification in a complete manner such that it is possible that one can define probabilistic more lenient action consistency on top of PLoss. Thus, we devote this section to experiments with PLoss and PbLoss. In addition we show time acceleration speedup of the calculation of belief dependent rewards in the belief tree as a result of simplification. Further we discuss empirical action consistency for *any* objective operator φ.

We exemplify our method on the problem of autonomous navigation to a goal with light beacons, which can be used for localization. In all our simulations in this section, the return g_k is a cumulative reward. In this study, the simplification conforms to (17). As the action space we take the space of motion primitives. Moreover, let us emphasize we do not average the simplified rewards taken for the approximation of standard error since we aim to examine general behavior and standard error possibly can be estimated without resampling of the returns. The bounds are calculated according to (97).

For simplicity, assume we have a linear motion model T, where $x \in \mathbb{R}^2$ as well as $a \in \mathbb{R}^2$

$$x_{k+1} = x_k + a_k + w_k \quad w_k \sim \mathcal{N}(0, \Sigma_w), \tag{111}$$

where $\Sigma_w = w \cdot I$ (w is a given parameter) and action $a_k \in \mathcal{A}$, and where the action space $\mathcal{A}$ is the space of motion primitives of unit length.

We consider next probabilistic and absolute action consistency description using PLoss offline and PbLoss online. We say that the action consistency is probabilistic if the probability that a pair of samples of the return will not preserve the trend with respect to a pair of actions due to simplification is larger than zero. Remarkably, the analysis below is valid for any objective operator φ.

Characterizing probabilistic action consistency The observation model O is as follows, $z \sim \mathcal{N}(x, \Sigma_v(x))$, where the spatially-varying covariance matrix is

$$\Sigma_v(x) = v(x) \cdot I, \quad v(x) = w \cdot \min\{1, \|x - x^*\|_2^2\}, \tag{112}$$

where x^* is the location of the light beacon closest to x. The noise has a constant variance w. Without losing generality, we assume b_k at planning time is uniformly distributed in a unit square. We set $L = 12$ and compare two action sequences: $a_{k+1:k+12}$ is six times $(1, 0)^T$ and after that six times $(0, 1)^T$. In the action sequence $a'_{k+1:k+12}$ we switched the order of actions such that the robot performs six times $(0, 1)^T$ and after that six times $(1, 0)^T$.

One realization of a possible future in terms of measurements and corresponding posterior beliefs is illustrated in Fig. 14. It is clearly seen that proximity to a beacon improves localization. Note, the robot is always able to avoid a dead reckoning scenario as it always gets an observation from the closest beacon. We hope that this setting conveys a real world scenario where an ambulating robot is equipped with long and short range sensors. The close range sensors are activated when the robot is inside a unit circle around the beacon. When the robot is outside a unit circle from the closest beacon, the beacon is detectable only by the long range sensors, which are less sensitive. We present results of the simplification for $w = 0.1$, $N = 1500$, $m = 50$, $\alpha = 0.01$, $z_{\alpha/2} = 2.56$, and the total number of observations is 500. For each sample of $z_{k+1:k+L}$, we sampled $b_{k+1:k+L}$ once. As we see in the left part of Fig. 15 we gained speedup as expected (104) for $n = 175$. We show measurements of all running times in our simulations in Table 4.

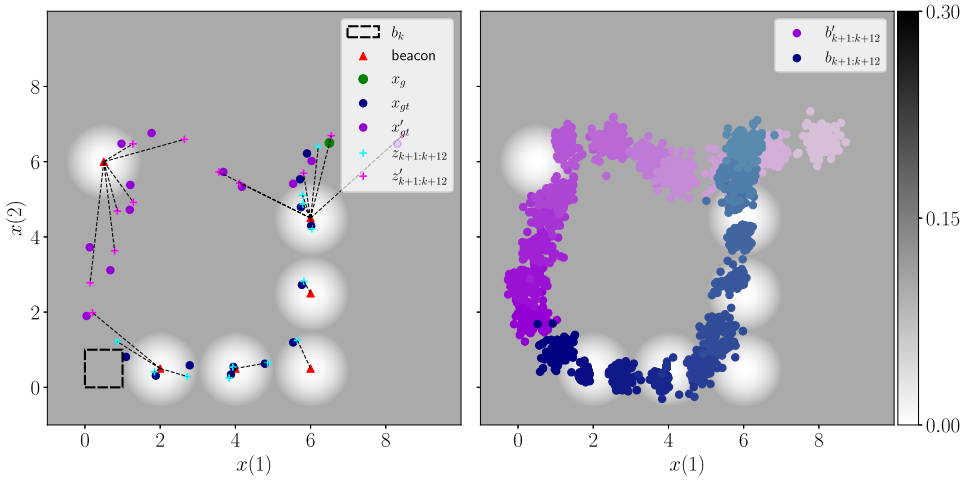


Fig. 14. Results for scenario 1 - probabilistic action consistency: Illustration of one realization of the future in a simulated scenario considering two possible action sequences. We start from b_k represented by samples uniformly distributed on a unit square. We demonstrated two sequences of observations alongside ground truth state samples, and the closest beacons produced these observations from the left. From the right, we plotted two sequences of the beliefs produced by these two histories. We show 100 most probable samples of each belief.

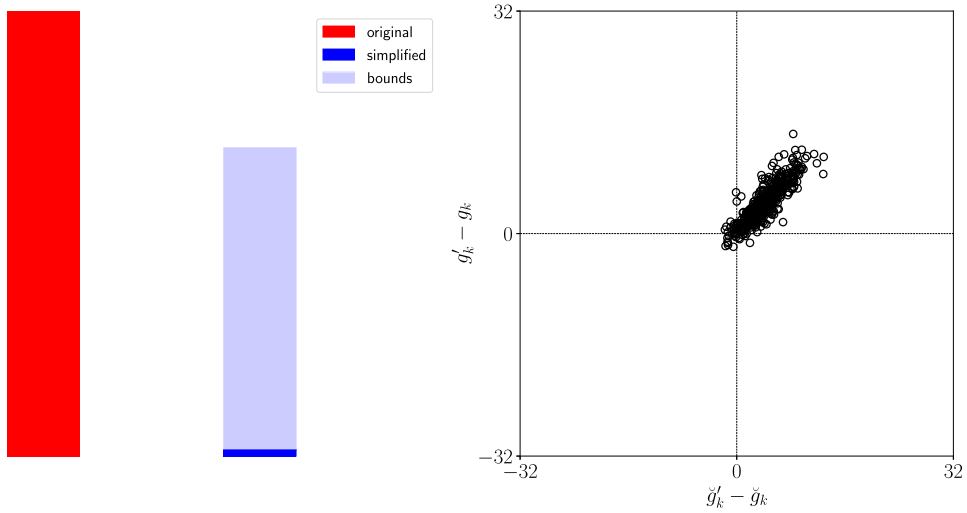


Fig. 15. Results for scenario 1 - probabilistic action consistency: (left) Demonstration of runtimes of the total number of the returns for a given extended belief tree where $N = 1500$ and $n = 175$. Note that this illustration agrees with (104); (right) action consistency of the samples of the return.

From these samples of the returns and bounds, we build PLoss and PbLoss in Fig. 19. In the right part of Fig. 15 quadrants II and IV, we observe samples that are not action consistent. To assess performance we need to choose some representative Δ. Since online we have access exclusively to the simplified problem, let us choose $\breve{\Delta}^* = |\mathbb{E}[\breve{g}_k|b_k, \pi, \nu] - \mathbb{E}[\breve{g}'_k|b_k, \pi', \nu]|$ and $\Delta = 0.5\breve{\Delta}^*$. Note that under our model in average the sample mean is not influenced by the lowering the number of samples of the reward. Only the variance of sample mean is increased. Moreover we assume that the distributions are without gaps such that the expected value of the return is some sample with probability density function larger than zero. Table 3 quantifies online characterization against offline PLoss TDF.

We showed an illustration of this scenario in Fig. 14. In Fig. 16, we demonstrated scatter plots that show samples of the simplified and original returns' differences. We identify that with decreasing n, more samples are not action consistent. This phenomenon is corroborated by the histograms of $\mathcal{L}$ in Fig. 17.

Let us focus on $n = 175$ in Fig. 18; *online* we can conclude that probability that loss incurred by this simplification will be greater than $\breve{\Delta}^*$ is at most 0.11, while actual $P(\mathcal{L} > \breve{\Delta}^*|\cdot)$ is 0.0. Similarly, the probability for loss incurred by this simplification to be greater than $0.5\breve{\Delta}^*$ is at most 0.33, while actual $P(\mathcal{L} > 0.5\breve{\Delta}^*|\cdot)$ is 0.0. In this scenario, the simplification is not absolute action consistent; it means variability described by (87) is sufficient to switch the order of the returns and incur loss Δ at some sampled realization.

Furthermore, our bounds depend on variance ($\text{se}^2(n)$) of the sample approximation of the reward (97), which, according to (87) does not depend on Δ. Hence, as Δ decreases towards zero, the contribution of variance versus the difference between simplified returns grows for any realization of $\bar{\mathcal{L}}$. Therefore, PbLoss departs from PLoss as Δ decreases. We observe this behavior in Fig. 18. Moreover, with the diminishing number of samples, this effect is amplified, as demonstrated

Table 3
Results for scenario 1 - probabilistic action consistency: Online characterization for $N = 1500$, $\alpha = 0.01$, $z_{\alpha/2} = 2.56$.

n	$P(\mathcal{L} > 0.5\breve{\Delta}^* \mid \cdot)$	$\theta_\alpha(0.5\breve{\Delta}^*)$	$\breve{\Delta}^*$	$P(\mathcal{L} > \breve{\Delta}^* \mid \cdot)$	$\theta_\alpha(\breve{\Delta}^*)$
175	0.0	0.33	4.14	0.0	0.11
150	0.01	0.43	4.04	0.0	0.17
125	0.01	0.43	4.21	0.0	0.2
100	0.0	0.56	4.08	0.0	0.29
75	0.01	0.64	4.01	0.0	0.39
50	0.02	0.83	3.72	0.01	0.63
25	0.07	1.0	3.34	0.03	0.94

Table 4
Results for scenario 1 - probabilistic action consistency: run times for $N = 1500$.

	$n = 175$	$n = 150$	$n = 125$	$n = 100$	$n = 75$	$n = 50$	$n = 25$
g_k time [sec]	104957	69658	95651	69713	68584	96354	66513
$\breve{g}_k$ and l, u time [sec]	72694	34842	33759	15498	8293	5589	969
$\breve{g}_k$ time [sec]	1454	661	669	298	172	119	14
l, u time [sec]	71240	34181	33090	15200	8121	5469	955

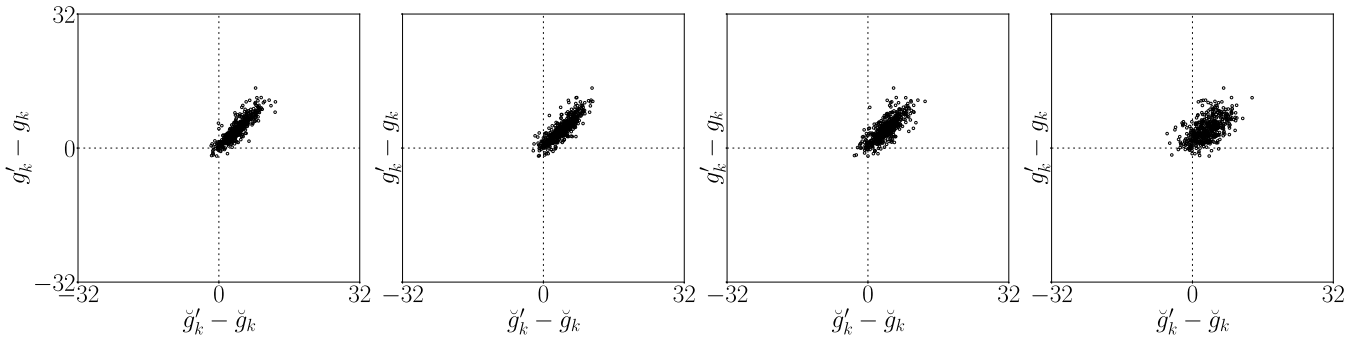


Fig. 16. Results for scenario 1 - probabilistic action consistency: We demonstrate from the left to the right action consistency of the samples of the returns for $n = 175, n = 125, n = 75, n = 25$, whereas $N = 1500$. As we see, samples violating action consistency are present at all graphs.

in Fig. 18, due to growing variance (87). Remarkably, when samples of original returns are more distinct, the effect of variance is nullified. In such a setting, our characterization is incredibly precise, see Fig. 24.

Thus, the behavior of the PbLoss is more conservative in more delicate scenarios, where two candidate policies are close to each other in terms of returns. Importantly, for significantly different policies, PbLoss becomes tighter to PLoss. This brings us to the next section.

Revealing empirical absolute action consistency In this scenario we modified the noise in the observation model as such $v(x) = w \cdot \|x - x^*\|_2^2$. In addition we removed one beacon on the way of the second action sequence. We remain with $w = 0.1$, $m = 50$, $\alpha = 0.01$, $z_{\alpha/2} = 2.56$ and set $N = 1000$. In this scenario the returns of two action sequences are much more distant. The samples in the right segment of Fig. 21 are more distant from the origin than in Fig. 15. The characterization is shown in Table 5. Therefore, the simplification is empirically absolute action consistent. As we see from the Table 5, observing $\theta_\alpha(\Delta = 0.0)$ we are able to identify online that for $n = 100$ and $n = 75$, probability to receive samples of the returns violating action consistency is at most 0.03, while $P(\mathcal{L} > 0.0 \mid \cdot)$ is 0.0.

Here the covariance matrix of the observation model is

$$\Sigma_v(x) = v(x) \cdot I, \quad v(x) = w \cdot \|x - x^*\|_2^2. \tag{113}$$

We demonstrated this scenario in Fig. 20. As we can see in Fig. 22, the clouds of samples are farther from the origin than in the previous scenario. Therefore, two action sequences are more distant. In this case, the simplification is empirically absolute action consistent, as we observe in the histograms of $\mathcal{L}$ in Fig. 23 and empirical characterization shown in Fig. 24. We report run times for two scenarios in Table 4 and Table 6, respectively.

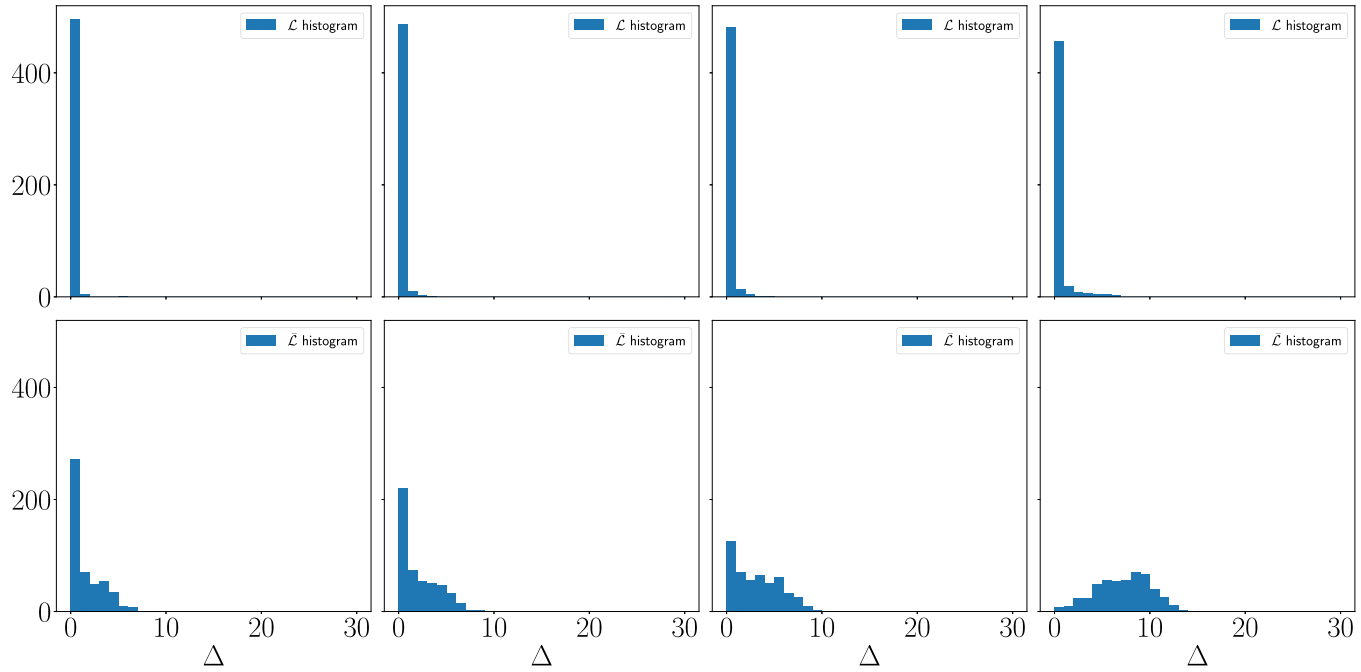


Fig. 17. Results for scenario 1 - probabilistic action consistency: Histograms of `PLoss` and `PbLoss` for $N = 1500$, $\alpha = 0.01$, $z_{\alpha/2} = 2.56$, bin width is 1.0; from the left to the right $n = 175$, $n = 125$, $n = 75$, $n = 25$.

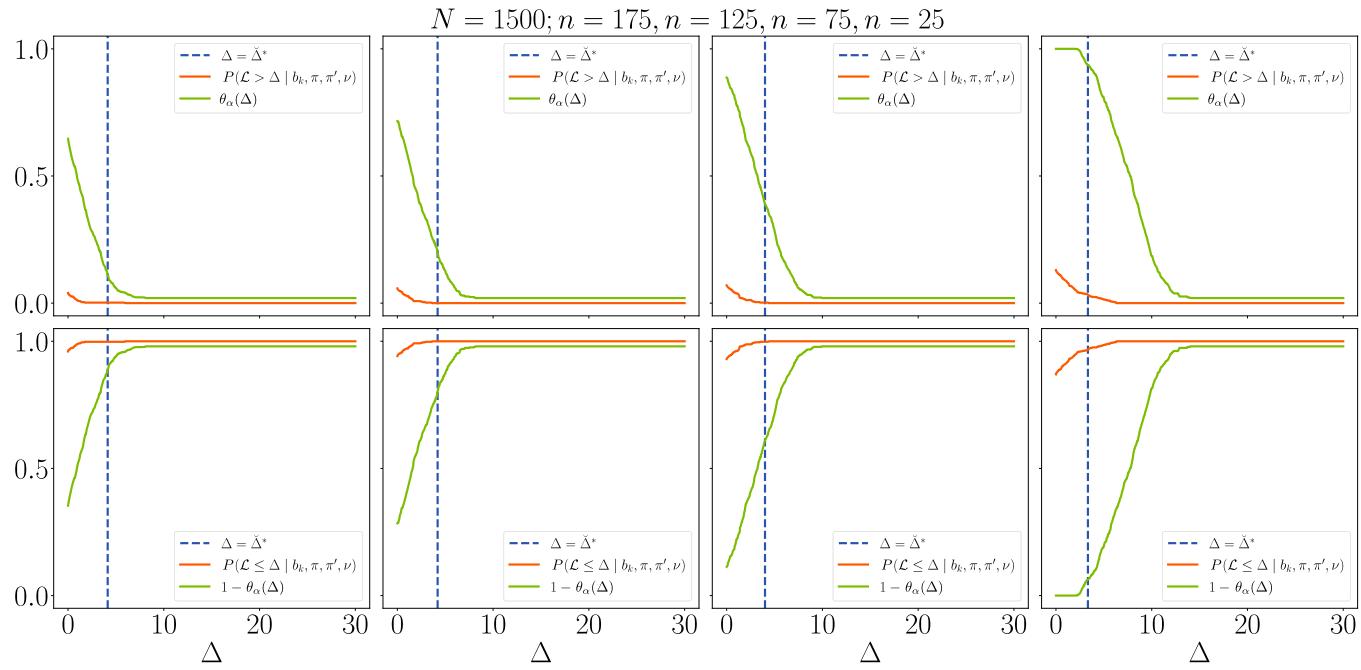


Fig. 18. Results for scenario 1 - probabilistic action consistency: Empirical characterization for $N = 1500$, $\alpha = 0.01$, $z_{\alpha/2} = 2.56$, evaluated in a grid with intervals 0.001; from the left to the right $n = 175$, $n = 125$, $n = 75$, $n = 25$.

Table 5
Results for scenario 2 - empirical absolute action consistency: Online characterization for $N = 1000$, $\alpha = 0.01$, $z_{\alpha/2} = 2.56$.

n	$P(\mathcal{L} > 0.0 \mid \cdot)$	$\theta_\alpha(\Delta = 0.0)$	$\breve{\Delta}^*$	$P(\mathcal{L} > \breve{\Delta}^* \mid \cdot)$	$\theta_\alpha(\breve{\Delta}^*)$
100	0.0	0.03	17.54	0.0	0.02
75	0.0	0.03	17.14	0.0	0.02
50	0.0	0.06	16.65	0.0	0.02
25	0.0	0.19	15.27	0.0	0.02

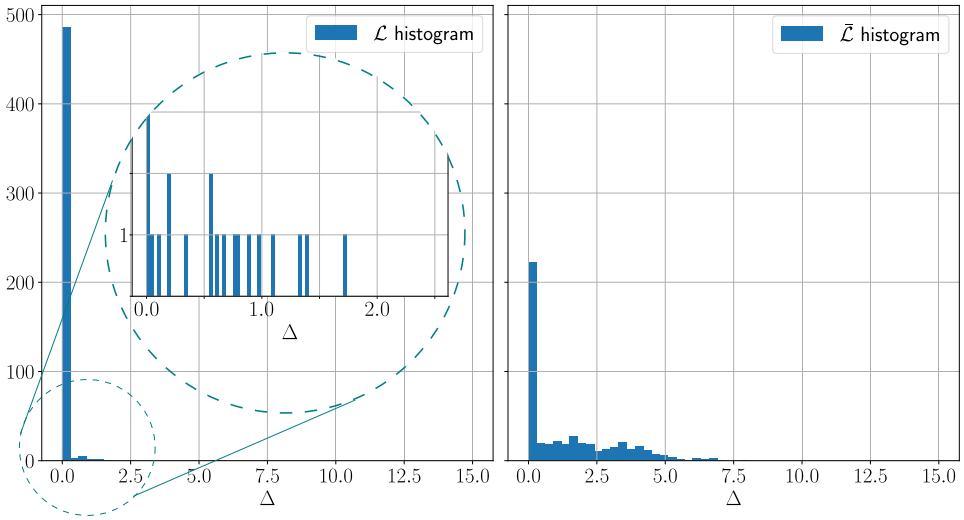


Fig. 19. Results for scenario 1 - probabilistic action consistency: Histograms of `PLoss` and `PbLoss` for $N = 1500$, $n = 175$, $\alpha = 0.01$, $z_{\alpha/2} = 2.56$ (bin width is 0.3, in zoom-in, bin width is 0.03).

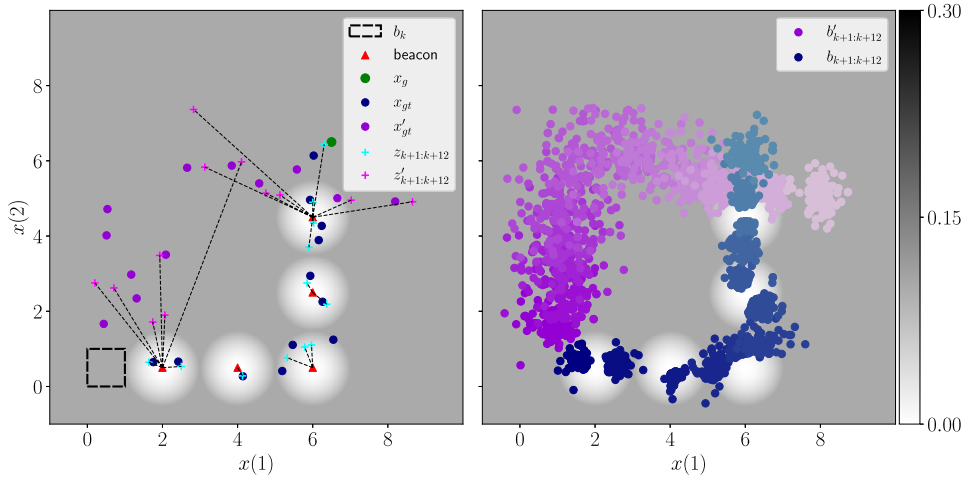


Fig. 20. Results for scenario 2 - empirical absolute action consistency: Illustration of one realization of the future in a simulated scenario considering two possible action sequences. We start from b_k represented by samples uniformly distributed on a unit square. We demonstrated two sequences of observations alongside ground truth state samples, and the closest beacons produced these observations from the left. From the right, we plotted two sequences of the beliefs produced by these two histories. We show 100 most probable samples of each belief.

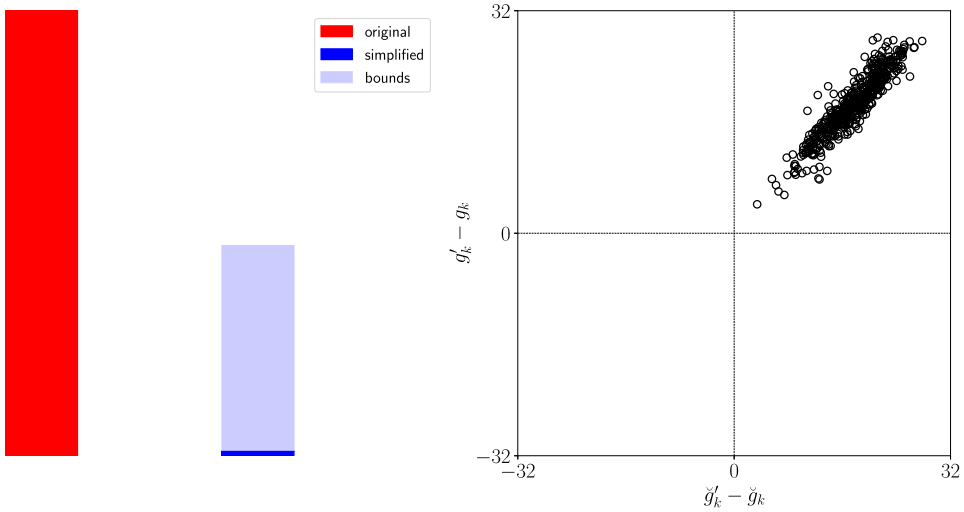


Fig. 21. Results for scenario 2 - empirical absolute action consistency: (left) Demonstration of runtimes of the total number of the returns for a given extended belief tree where $N = 1000$ and $n = 100$. Note that this illustration agrees with (104); (right) action consistency of the samples of the return.

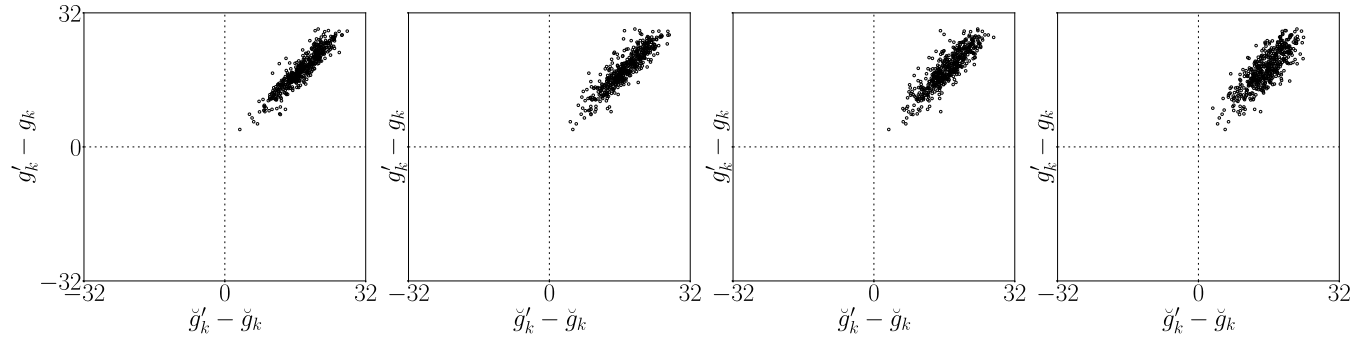


Fig. 22. Results for scenario 2 - empirical absolute action consistency: We demonstrate from the left to the right action consistency of the samples of the returns for $n = 100, n = 75, n = 50, n = 25$, whereas $N = 1000$. As we see, all the samples are action consistent.

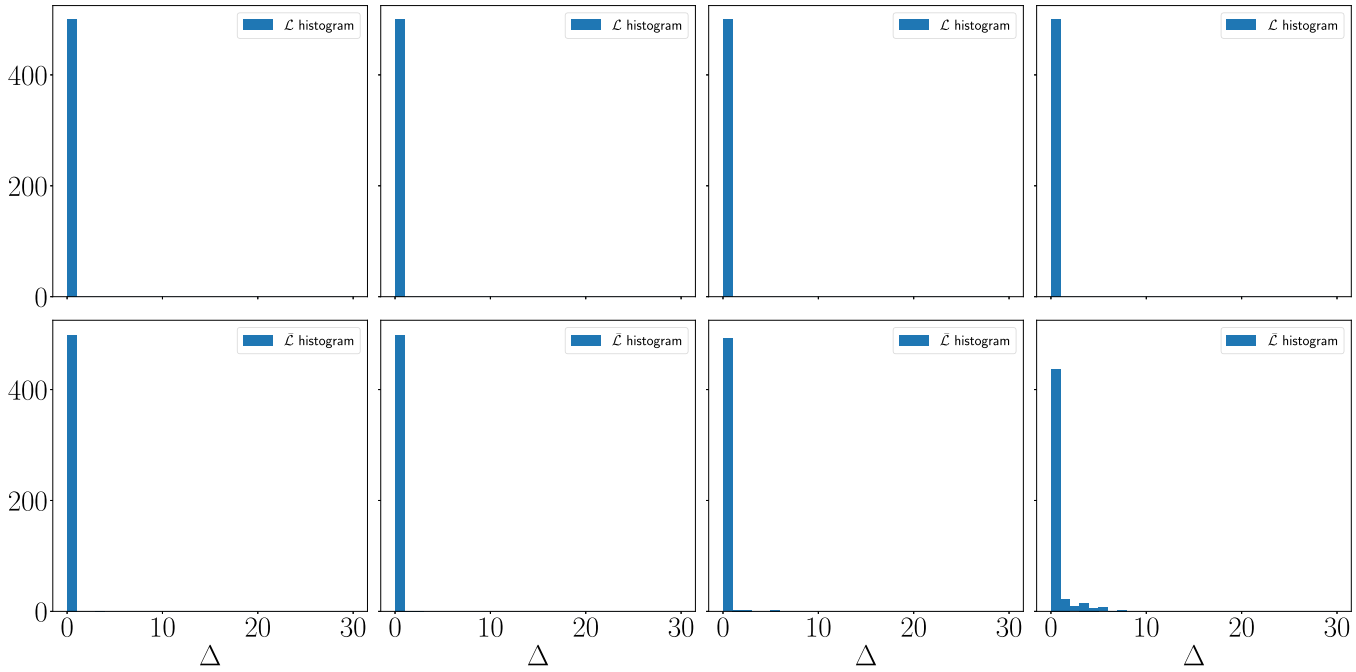


Fig. 23. Results for scenario 2 - empirical absolute action consistency: Histograms of `PLoss` and `PbLoss` for $N = 1000$, $\alpha = 0.01$, $z_{\alpha/2} = 2.56$, bin width is 1.0; from the left to the right $n = 100$, $n = 75$, $n = 50$, $n = 25$.

Table 6
Results for scenario 2 - empirical absolute action consistency: run times for $N = 1000$.

	$n = 100$	$n = 75$	$n = 50$	$n = 25$
g_k time [sec]	36745	45187	44899	30889
$\breve{g}_k$ and l, u time [sec]	17361	12546	4388	844
$\breve{g}_k$ time [sec]	363	247	65	14
l, u time [sec]	16998	12299	4323	830

8. Conclusions

We introduced a novel simplification framework in the challenging continuous domain with possibly nonparametric beliefs and general belief dependent rewards. We presented a formulation of novel stochastic bounds on the return and proved that these bounds yield deterministic bounds on VaR. We considered simplification impact also on the joint distribution of a pair of returns given two candidate policies, while accounting for the correlation between these returns. In this context, we proposed an innovative objective operator on top of the joint distribution. In addition, we presented a mathematical tool `PLoss` and its online counterpart `PbLoss` to characterize the simplification impact on the decision making entirely for any objective operator. Moreover, we utilized it to provide deterministic guarantees for our novel risk aware objective operator mounted on the joint distribution of a pair of returns given a pair of policies. We presented an instance of our framework

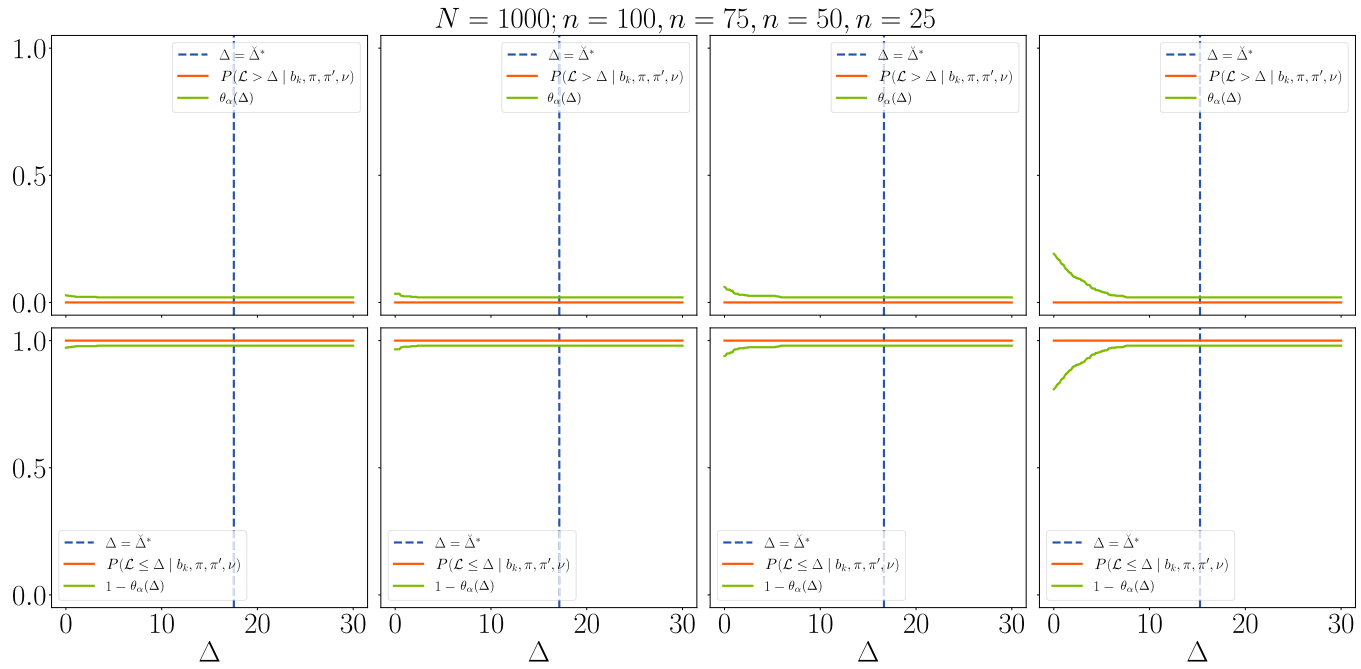


Fig. 24. Results for scenario 2 - empirical absolute action consistency: Empirical characterization for $N = 1000$, $\alpha = 0.01$, $z_{\alpha/2} = 2.56$, evaluated in a grid with intervals 0.001; from the left to the right $n = 100$, $n = 75$, $n = 50$, $n = 25$.

with a specific simplification method, which is reducing the number of samples of the return or the belief used for reward calculation. Finally, we verified the advantages of our approach through extensive simulations. For example, in section 7.1.2 we obtained approximately 8 times speedup with respect to the original problem while still identifying the optimal action.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Proofs for the theorems

A.1. Proof of the Theorem 1

Using the marginalization over future observations with Probability Density Function (PDF) being $\mathbb{P}(z_{k+}|b_k, \pi)$ we have that

$$P(\mathbf{1}_{\{l \leq g_k \leq u\}} = 1 | b_k, \pi, \nu) \geq (1-\alpha) \underbrace{\int_{z_{k+}} \mathbb{P}(z_{k+}|b_k, \pi) \mathrm{d}z_{k+}}_{=1} = 1 - \alpha. \tag{114}$$

The following holds from property of a lower bound (usual stochastic order) $\forall \xi \in (-\infty, \infty)$

$$P(l > \xi | b_k, \pi, \nu, \mathbf{1}_{\{l \leq g_k \leq u\}} = 1) \leq P(g_k > \xi | b_k, \pi, \mathbf{1}_{\{l \leq g_k \leq u\}} = 1). \tag{115}$$

Denote $\lambda = P(\mathbf{1}_{\{l \leq g_k \leq u\}} = 1 | b_k, \pi, \nu)$. This notation implies that $1 - \lambda = P(\mathbf{1}_{\{l \leq g_k \leq u\}} = 0 | b_k, \pi, \nu)$. Using marginalization over the indicator function we have that

$$P(g_k > \xi | b_k, \pi) = P(g_k > \xi | b_k, \pi, \nu, \mathbf{1}_{\{l \leq g_k \leq u\}} = 1)\lambda + P(g_k > \xi | b_k, \pi, \nu, \mathbf{1}_{\{l \leq g_k \leq u\}} = 0)(1 - \lambda). \tag{116}$$

Since each summand in the equation above is non negative and using (114) we obtain

$$P(g_k > \xi | b_k, \pi) \geq P(l > \xi | b_k, \pi, \nu, \mathbf{1}\{l \leq g_k \leq u\} = 1)(1 - \alpha). \tag{117}$$

Assume $\alpha \in [0, 1)$, exist $c \in \mathbb{R}^+$ such that

$$P(g_k > \xi | b_k, \pi, \nu) = c(1 - \alpha). \tag{118}$$

This implies

$$P(g_k > \xi \cap \mathbf{1}\{l \leq g_k \leq u\} = 1 | b_k, \pi, \nu) \leq P(g_k > \xi | b_k, \pi, \nu) = c(1-\alpha). \tag{119}$$

Applying the chain rule and rearranging the terms, we have that

$$P(g_k > \xi | \mathbf{1}\{l \leq g_k \leq u\} = 1, b_k, \pi, \nu) \leq c \underbrace{\frac{1-\alpha}{\lambda}}_{\leq 1} \leq c. \tag{120}$$

Using again marginalization over the indicator function, we represent the $P(l > \xi | b_k, \pi, \nu)$ as

$$P(l > \xi | b_k, \pi, \nu, \mathbf{1}_{\{l \leq g_k \leq u\}} = 1)\lambda + P(l > \xi | b_k, \pi, \nu, \mathbf{1}_{\{l \leq g_k \leq u\}} = 0)(1-\lambda). \tag{121}$$

Using that $\lambda \leq 1$ and $P(l > \xi | b_k, \pi, \nu, \mathbf{1}_{\{l \leq g_k \leq u\}} = 0) \leq 1$ we have that

$$P(l > \xi | b_k, \pi, \nu) \leq P(l > \xi | b_k, \pi, \nu, \mathbf{1}_{\{l \leq g_k \leq u\}} = 1) + 1 - \lambda \leq P(l > \xi | b_k, \pi, \nu, \mathbf{1}_{\{l \leq g_k \leq u\}} = 1) + 1 - (1-\alpha). \tag{122}$$

Using (115) and (118), we arrive at the desired result

$$\begin{aligned} &P(l > \xi | b_k, \pi, \nu, \mathbf{1}_{\{l \leq g_k \leq u\}} = 1) + \alpha \leq \\ &P(g_k > \xi | b_k, \pi, \nu, \mathbf{1}_{\{l \leq g_k \leq u\}} = 1) + \alpha \leq c + \alpha = \frac{P(g_k > \xi | b_k, \pi)}{1-\alpha} + \alpha. \end{aligned} \tag{123}$$

Rearranging the terms bears

$$\big(P(l > \xi | b_k, \pi, \nu) - \alpha\big)(1-\alpha) \leq P(g_k > \xi | b_k, \pi). \tag{124}$$

Switching the roles of g_k to u and l to g_k, we obtain the upper bound

$$P(g_k > \xi | b_k, \pi) \leq \frac{P(u > \xi | b_k, \pi, \nu)}{1-\alpha} + \alpha. \tag{125}$$

This completes the proof. □

A.2. Proof of the Theorem 2

Let us start from upper bound. From Theorem 1

$$\left\{\xi \text{ s.t } P(g_k > \xi | b_k, \pi, \nu) \geq 1-\beta\right\} \subseteq \left\{\xi \text{ s.t } \frac{P(u > \xi | b_k, \pi, \nu)}{1-\alpha} + \alpha \geq 1-\beta\right\}. \tag{126}$$

Equivalently

$$\left\{\xi \text{ s.t } P(g_k > \xi | b_k, \pi, \nu) \geq 1-\beta\right\} \subseteq \left\{\xi \text{ s.t } P(u > \xi | b_k, \pi, \nu) \geq (1-\beta-\alpha)(1-\alpha)\right\}. \tag{127}$$

Rearranging the terms, we have that

$$\sup\left\{\xi \text{ s.t } P(g_k > \xi | b_k, \pi, \nu) \geq 1-\beta\right\} \leq \sup\left\{\xi \text{ s.t } P(u > \xi | b_k, \pi, \nu) \geq 1 - (\beta + \alpha(2-\beta-\alpha)\right\}. \tag{128}$$

It is left to show that

$$0 \leq \beta + \alpha(2-\beta-\alpha) \leq 1. \tag{129}$$

Since $\alpha + \beta \leq 2$ and $\alpha \geq 0$, we have

$$0 \leq \beta \leq \beta + \alpha(2-\beta-\alpha). \tag{130}$$

To prove the right inequality we show that

$$\beta + \alpha(2-\beta-\alpha) - 1 \leq 0. \tag{131}$$

Multiplying by -1 we observe that the inequality reads

$$(1-\beta-\alpha)\underbrace{(1-\alpha)}_{\geq 0} \geq 0. \tag{132}$$

Requiring that $1-\beta-\alpha \geq 0$, we obtain the condition which we already assumed

$$(1-\alpha) \geq \beta. \tag{133}$$

We have that

$$\mathrm{VaR}_{\beta}(g_k|b_k,\pi,\nu) \leq \mathrm{VaR}_{\beta+\alpha(2-\beta-\alpha)}(u|b_k,\pi,\nu). \tag{134}$$

To prove the second part of the theorem we use the following

$$\left\{\xi \text{ s.t } (P(l>\xi|b_k,\pi,\nu)-\alpha)(1-\alpha) \geq 1-\beta\right\} \subseteq \left\{\xi \text{ s.t } P(g_k>\xi|b_k,\pi,\nu) \geq 1-\beta\right\}. \tag{135}$$

Equivalently,

$$\left\{\xi \text{ s.t } P(l>\xi|b_k,\pi,\nu) \geq 1-\left(1-\frac{1-\beta}{1-\alpha}-\alpha\right)\right\} \subseteq \left\{\xi \text{ s.t } P(g_k>\xi|b_k,\pi,\nu) \geq 1-\beta\right\}. \tag{136}$$

It is left to show that

$$0 \leq 1-\frac{1-\beta}{1-\alpha}-\alpha \leq 1. \tag{137}$$

We immediately see that $1-\frac{1-\beta}{1-\alpha}-\alpha \leq 1$. Rearranging the terms, we arrive at the second condition that

$$\alpha(2-\alpha) \leq \beta. \tag{138}$$

This completes the proof. □

A.3. Proof of the Theorem 3

By definition

$$P(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}}=1|\mathbf{1}_{\{l\leq g_k\leq u\}}=1,\mathbf{1}_{\{l'\leq g'_k\leq u'\}}=1,b_k,\pi,\pi',z_{k+},z'_{k+},\nu)=1. \tag{139}$$

We first apply marginalization over future observations $z_{k+}\equiv z_{k+1:k+L}$ and $z'_{k+}\equiv z'_{k+1:k+L}$, and events $\{\omega|l(\omega)\leq g_k(\omega)\leq u(\omega)\}$ and $\left\{\omega|l'(\omega)\leq g'_k(\omega)\leq u'(\omega)\right\}$. We then use the fact that given two histories $\mathcal{H}_{k+L}\triangleq\{b_k,\pi,z_{k+}\}$ and $\mathcal{H}'_{k+L}\triangleq\{b_k,\pi',z'_{k+}\}$, the events $\{\omega|l(\omega)\leq g_k(\omega)\leq u(\omega)\}$ and $\left\{\omega|l'(\omega)\leq g'_k(\omega)\leq u'(\omega)\right\}$ are independent of each other. Furthermore, each such event depends exclusively on its own history by design. We have that $P\left(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}}=1|b_k,\pi,\pi',\nu\right)$ equals to

$$\int\limits_{\substack{z_{k+}\\ z'_{k+}}} P\left(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}}=1|b_k,\pi,\pi',z_{k+},z'_{k+},\nu\right)\mathbb{P}\left(z_{k+},z'_{k+}\mid b_k,\pi,\pi'\right)\mathrm{d}z_{k+}\mathrm{d}z'_{k+}. \tag{140}$$

Moreover, the $P\left(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}}=1|b_k,\pi,\pi',z_{k+},z'_{k+},\nu\right)$ is larger or equal to

$$P(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}}=1\wedge\mathbf{1}_{\{l\leq g_k\leq u\}}=1\wedge\mathbf{1}_{\{l'\leq g'_k\leq u'\}}=1\mid b_k,\pi,\pi',z_{k+},z'_{k+},\nu). \tag{141}$$

Engaging the chain rule and using the constraints (59) and (60), and their statistical independence we face that

$$P\left(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}}=1|b_k,\pi,\pi',z_{k+},z'_{k+},\nu\right)\geq(1-\alpha)^2. \tag{142}$$

The above expression straightforwardly yields that $P\left(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}}=1|b_k,\pi,\pi',\nu\right)\geq(1-\alpha)^2$ through the marginalization over the future observations since $\int_{\substack{z_{k+}\\ z'_{k+}}}\mathbb{P}\left(z_{k+},z'_{k+}\mid b_k,\pi,\pi'\right)\mathrm{d}z_{k+}\mathrm{d}z'_{k+}=1$. This completes the proof. □

A.4. Proof of the Theorem 4

To shorten notations let us denote $|b_k,\pi,\pi',\nu$ by $|\cdot$ in the proof. Let us express `PLoss` TDF as

$$P(\mathcal{L}>\Delta|\cdot)=P\left(\mathcal{L}>\Delta|\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}}=1,\cdot\right)P\left(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}}=1|\cdot\right)+P\left(\mathcal{L}>\Delta|\mathbf{1}_{\{\bar{\mathcal{L}}<\mathcal{L}\}}=1,\cdot\right)P\left(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}}=0|\cdot\right). \tag{143}$$

Similarly, `PbLoss` TDF reads

$$P\left(\bar{\mathcal{L}} > \Delta|\cdot\right) = P\left(\bar{\mathcal{L}} > \Delta|\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}} = 1, \cdot\right) P\left(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}} = 1|\cdot\right) + P\left(\bar{\mathcal{L}} > \Delta|\mathbf{1}_{\{\bar{\mathcal{L}}<\mathcal{L}\}} = 1, \cdot\right) P\left(\mathbf{1}_{\{\bar{\mathcal{L}}<\mathcal{L}\}} = 1|\cdot\right). \tag{144}$$

Since $\alpha \in [0, 1)$ it exists $c \in \mathbb{R}_{>0}$ such that

$$P\left(\mathcal{L} > \Delta \wedge \mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}} = 1|\cdot\right) \leq P\left(\bar{\mathcal{L}} > \Delta|\cdot\right) = c(1-\alpha)^2. \tag{145}$$

This implies

$$P\left(\bar{\mathcal{L}} > \Delta|\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}} = 1, \cdot\right) P\left(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}} = 1|\cdot\right) \leq c(1-\alpha)^2, \tag{146}$$

$$P\left(\bar{\mathcal{L}} > \Delta|\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}} = 1, \cdot\right) \leq c \underbrace{\frac{(1-\alpha)^2}{P\left(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}} = 1|\cdot\right)}}_{\leq 1} \leq c. \tag{147}$$

Moreover, using that $P(\mathbf{1}_{\{\bar{\mathcal{L}}\geq\mathcal{L}\}} = 1|\cdot) + P(\mathbf{1}_{\{\bar{\mathcal{L}}<\mathcal{L}\}} = 1|\cdot) = 1$, we obtain

$$\begin{aligned} P\left(\mathcal{L} > \Delta|\cdot\right) = & P\left(\mathcal{L} > \Delta|\bar{\mathcal{L}} \geq \mathcal{L}, \cdot\right) P\left(\bar{\mathcal{L}} \geq \mathcal{L}|\cdot\right) + \\ & P\left(\mathcal{L} > \Delta|\bar{\mathcal{L}} < \mathcal{L}, \cdot\right) \left(1 - P(\bar{\mathcal{L}} \geq \mathcal{L})|\cdot\right) \leq \\ & P\left(\mathcal{L} > \Delta|\bar{\mathcal{L}} \geq \mathcal{L}, \cdot\right) + 1 - (1-\alpha)^2 \leq c + 2\alpha - \alpha^2, \end{aligned} \tag{148}$$

but $c = \frac{P(\bar{\mathcal{L}} > \Delta|b_k, \pi, \pi', \nu)}{(1-\alpha)^2}$. We showed that

$$P\left(\mathcal{L} > \Delta|\cdot\right) \leq \frac{P\left(\bar{\mathcal{L}} > \Delta|b_k, \pi, \pi', \nu\right)}{(1-\alpha)^2} + 2\alpha - \alpha^2. \tag{149}$$

Furthermore, by definition of TDF

$$P\left(\mathcal{L} > \Delta|\cdot\right) \leq 1. \tag{150}$$

We write the above two relations compactly as

$$P\left(\mathcal{L} > \Delta|\cdot\right) \leq \theta_\alpha(\Delta), \tag{151}$$

where $\theta_\alpha(\Delta) = \min\left\{1, \frac{P(\bar{\mathcal{L}} > \Delta|b_k, \pi, \pi', \nu)}{(1-\alpha)^2} + 2\alpha - \alpha^2\right\}$. Clearly

$$P\left(\mathcal{L} \leq \Delta|b_k, \pi, \pi', \nu\right) = 1 - P\left(\mathcal{L} > \Delta|b_k, \pi, \pi', \nu\right) \geq 1 - \theta_\alpha(\Delta). \tag{152}$$

This concludes the proof. □

Appendix B. Technical characteristics of computers used in simulations

Our simulations are written in Julia language with a multi-threaded calculation of immediate reward. We used 4 computers with the following characteristics:

1. 40 cores Intel(R) Xeon(R) E5-2670 v2 with 256 GB of RAM working at 2.50 GHz;
2. 24 cores Intel(R) Core(TM) i9-7920X with 64 GB of RAM working at 2.90 GHz;
3. 20 cores Intel(R) Xeon(R) E5-2630 v4 with 64 GB of RAM working at 2.20 GHz;
4. 20 cores Intel(R) Core(TM) i9-9820X with 64 GB of RAM working at 3.30 GHz.

References

[1] M. Araya, O. Buffet, V. Thomas, F. Charpillet, A pomdp extension with belief-dependent rewards, in: Advances in Neural Information Processing Systems (NIPS), 2010, pp. 64–72.
[2] M. Barenboim, V. Indelman, Adaptive information belief space planning, in: The 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI), 2022.
[3] D. Bertsekas, Dynamic Programming and Optimal Control, vol. 1, Athena Scientific, Belmont, MA, 1995.
[4] Y. Boers, H. Driessen, A. Bagchi, P. Mandal, Particle filter based entropy, in: 2010 13th International Conference on Information Fusion, 2010, pp. 1–8.
[5] Y. Chow, A. Tamar, S. Mannor, M. Pavone, Risk-sensitive and robust decision-making: a cvar optimization approach, Adv. Neural Inf. Process. Syst. 28 (2015) 1522–1530.
[6] B. Defourny, D. Ernst, L. Wehenkel, Risk-aware decision making and dynamic programming, in: NIPS Workshop on Model Uncertainty and Risk in RL, 2008.

[7] L. Dressel, M.J. Kochenderfer, Efficient decision-theoretic target localization, in: L. Barbulescu, J. Frank, Smith S.F. Mausam (Eds.), Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS 2017, Pittsburgh, Pennsylvania, USA, June 18–23, 2017, AAAI Press, 2017, pp. 70–78, https://aaai.org/ocs/index.php/ICAPS/ICAPS17/paper/view/15761.
[8] K. Elimelech, V. Indelman, Simplified decision making in the belief space using belief sparsification, Int. J. Robot. Res. 41 (2022) 470–496, https://doi.org/10.1177/02783649221076381.
[9] M. Fehr, O. Buffet, V. Thomas, J. Dibangoye, rho-pomdps have Lipschitz-continuous epsilon-optimal value functions, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 31, Curran Associates, Inc., 2018, pp. 6933–6943.
[10] J. Fischer, O.S. Tas, Information particle filter tree: an online algorithm for pomdps with belief-based rewards on continuous domains, in: Intl. Conf. on Machine Learning (ICML), Vienna, Austria, 2020.
[11] D. Fox, Adapting the sample size in particle filters through kld-sampling, Int. J. Robot. Res. 22 (2003) 985–1003.
[12] A. Hakobyan, G.C. Kim, I. Yang, Risk-aware motion planning and control using cvar-constrained optimization, IEEE Robot. Autom. Lett. 4 (2019) 3924–3931.
[13] A. Hakobyan, I. Yang, Wasserstein distributionally robust motion planning and control with safety constraints using conditional value-at-risk, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 490–496.
[14] M. Hoerger, H. Kurniawati, A. Elfes, Multilevel Monte-Carlo for solving pomdps online, in: Proc. International Symposium on Robotics Research (ISRR), 2019.
[15] M.F. Huber, T. Bailey, H. Durrant-Whyte, U.D. Hanebeck, On entropy approximation for Gaussian mixture random vectors, in: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, 2008, pp. 181–188.
[16] V. Indelman, No correlations involved: decision making under uncertainty in a conservative sparse information space, IEEE Robot. Autom. Lett. 1 (2016) 407–414.
[17] V. Indelman, L. Carlone, F. Dellaert, Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments, Int. J. Robot. Res. 34 (2015) 849–882.
[18] L.P. Kaelbling, M.L. Littman, A.R. Cassandra, Planning and acting in partially observable stochastic domains, Artif. Intell. 101 (1998) 99–134.
[19] A. Kitanov, V. Indelman, Topological information-theoretic belief space planning with optimality guarantees, arXiv preprint arXiv:1903.00927, 2019.
[20] D.E. Knuth, Art of Computer Programming, Volume 2: Seminumerical Algorithms, Addison-Wesley Professional, 2014.
[21] M. Kochenderfer, T. Wheeler, K. Wray, Algorithms for Decision Making, MIT Press, 2022.
[22] L. Kocsis, C. Szepesvári, Bandit based Monte-Carlo planning, in: European Conference on Machine Learning, Springer, 2006, pp. 282–293.
[23] N. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2004.
[24] H. Kurniawati, D. Hsu, W.S. Lee, SARSOP: efficient point-based POMDP planning by approximating optimally reachable belief spaces, in: Robotics: Science and Systems (RSS), 2008.
[25] S. Pathak, A. Thomas, V. Indelman, A unified framework for data association aware robust belief space planning and perception, Int. J. Robot. Res. 32 (2018) 287–315.
[26] J. Pineau, G.J. Gordon, S. Thrun, Anytime point-based approximations for large POMDPs, J. Artif. Intell. Res. 27 (2006) 335–380.
[27] R. Platt, R. Tedrake, L. Kaelbling, T. Lozano-Pérez, Belief space planning assuming maximum likelihood observations, in: Robotics: Science and Systems (RSS), Zaragoza, Spain, 2010, pp. 587–593.
[28] J.M. Porta, N. Vlassis, M.T. Spaan, P. Poupart, Point-based value iteration for continuous pomdps, J. Mach. Learn. Res. 7 (2006) 2329–2367.
[29] J.A. Rice, Mathematical Statistics and Data Analysis, Cengage Learning, 2006.
[30] A. Ryan, Information-theoretic tracking control based on particle filter estimate, in: AIAA Guidance, Navigation and Control Conference, 2008, pp. 1–15.
[31] P. Santana, S. Thiébaux, B. Williams, Rao*: an algorithm for chance-constrained pomdp's, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2016.
[32] M. Shienman, V. Indelman, D2a-bsp: distilled data association belief space planning with performance guarantees under budget constraints, in: IEEE Intl. Conf. on Robotics and Automation (ICRA), 2022.
[33] D. Silver, J. Veness, Monte-Carlo planning in large pomdps, in: Advances in Neural Information Processing Systems (NIPS), 2010, pp. 2164–2172.
[34] T. Smith, R. Simmons, Heuristic search value iteration for pomdps, in: Conf. on Uncertainty in Artificial Intelligence (UAI), 2004, pp. 520–527.
[35] A. Somani, N. Ye, D. Hsu, W.S. Lee, Despot: online pomdp planning with regularization, in: NIPS, 2013, pp. 1772–1780.
[36] M.T. Spaan, T.S. Veiga, P.U. Lima, Decision-theoretic planning under uncertainty with information rewards for active cooperative perception, Auton. Agents Multi-Agent Syst. 29 (2015) 1157–1185.
[37] Z. Sunberg, M. Kochenderfer, Online algorithms for pomdps with continuous state, action, and observation spaces, in: Proceedings of the International Conference on Automated Planning and Scheduling, 2018.
[38] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[39] O. Sztyglic, V. Indelman, Online pomdp planning via simplification, arXiv preprint arXiv:2105.05296, 2021.
[40] O. Sztyglic, V. Indelman, Speeding up online pomdp planning via simplification, in: IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2022.
[41] O. Sztyglic, A. Zhitnikov, V. Indelman, Simplified belief-dependent reward mcts planning with guaranteed tree consistency, arXiv preprint arXiv:2105.14239, 2021.
[42] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, The MIT Press, Cambridge, MA, 2005.
[43] C. Voss, M. Moll, L.E. Kavraki, A heuristic approach to finding diverse short paths, in: IEEE Intl. Conf. on Robotics and Automation (ICRA), 2015, pp. 4173–4179.
[44] N. Ye, A. Somani, D. Hsu, W.S. Lee, Despot: online pomdp planning with regularization, J. Artif. Intell. Res. 58 (2017) 231–266.

Simplified Continuous High-Dimensional Belief Space Planning With Adaptive Probabilistic Belief-Dependent Constraints

Andrey Zhitnikov and Vadim Indelman

***Abstract*—Online decision making under uncertainty in partially observable domains, also known as Belief Space Planning, is a fundamental problem in Robotics and Artificial Intelligence. Due to an abundance of plausible future unravelings, calculating an optimal course of action inflicts an enormous computational burden on the agent. Moreover, in many scenarios, e.g., Information gathering, it is required to introduce a belief-dependent constraint. Prompted by this demand, in this article, we consider a recently introduced probabilistic belief-dependent constrained partially observable Markov decision process (POMDP). We present a technique to adaptively accept or discard a candidate action sequence with respect to a probabilistic belief-dependent constraint, before expanding a complete set of sampled future observations episodes and without any loss in accuracy. Moreover, using our proposed framework, we contribute an adaptive method to find a maximal feasible return (e.g., Information Gain) in terms of Value at Risk and a corresponding action sequence, given a set of candidate action sequences, with substantial acceleration. On top of that, we introduce an *adaptive simplification* technique for a probabilistically constrained setting. Such an approach provably returns an identical-quality solution while dramatically accelerating the online decision making. Our universal framework applies to any belief-dependent constrained continuous POMDP with parameteric beliefs, as well as nonparameteric beliefs represented by particles. In the context of an information-theoretic constraint, our presented framework stochastically quantifies if a cumulative Information Gain along the planning horizon is sufficiently significant (for e.g., Information Gathering, active simultaneous localization and mapping (SLAM)). As a case study, we apply our method to two challenging problems of high dimensional belief space planning: active SLAM and sensor deployment. Extensive realistic simulations corroborate the superiority of our proposed ideas.**

***Index Terms*—Active simultaneous localization and mapping (SLAM), autonomous robotic exploration, belief space planning (BSP), belief-dependent probabilistic constraints, belief-dependent rewards, constrained belief-dependent partially observable Markov decision process (POMDP).**

Manuscript received 11 August 2023; revised 12 November 2023; accepted 19 November 2023. Date of publication 12 December 2023; date of current version 12 February 2024. This paper was recommended for publication by Associate Editor Maurice Fallon and Editor Sven Behnke upon evaluation of the reviewers' comments. This work was supported by the Israel Science Foundation (ISF). *(Corresponding author: Andrey Zhitnikov.)*

Andrey Zhitnikov is with the Technion Autonomous Systems Program (TASP), Haifa 32000, Israel (e-mail: andreyz@campus.technion.ac.il).

Vadim Indelman is with the Department of Aerospace Engineering Technion - Israel Institute of Technology, Haifa 32000, Israel (e-mail: vadim.indelman@technion.ac.il).

Digital Object Identifier 10.1109/TRO.2023.3341625

I. Introduction

A COMPREHENSIVE approach to craft many online decision-making problems, characterized by the agent situated in an environment and acting under uncertainty, is the partially observable Markov decision process (POMDP). For most such problems, it is sufficient to assume that the belief-dependent reward is merely the expectation of a state-dependent reward with respect to belief. This assumption is the case in classical POMDP formulations. In contrast, numerous problems in robotics, such as informative planning tasks [1], active simultaneous localization and mapping (SLAM) [2], and sensor deployment (SD) problem [3] are explicitly concerned with decreasing uncertainty, thereby raising the need for planning with general belief-dependent reward functionals.

General belief-dependent operators were examined in the context of reward but hardly so in the context of the constraint. In the robotics community, continuous POMDP with belief-dependent information-theoretic rewards is known as belief space planning (BSP) [4], [5]. Oftentimes the belief in BSP is over a high-dimensional state. In this article we focus on such a setting.

One of the embodiments of high-dimensional BSP, and also the subject of our interest, is active SLAM. Further we sometimes omit word "active." In SLAM, the environment where the robot operates is unknown and shall be revealed by the robot. Such a map can be represented, for instance, as a discrete occupancy grid [6] or continuous landmarks [5]. In the latter setting, typically the robot's state comprises the robot's pose trajectory and the map to be estimated. In the landmark-based SLAM the previous robot poses are not marginalized out but kept to preserve the sparse structure of the belief. Another related problem is SD. In this problem, a robot shall decide where to deploy sensors to measure some spatially dispersed continuous phenomenon, e.g., temperature. The map is represented by a grid, such that the number of grid cells is the dimension of the quantity of interest.

Both of these problems have a high-dimensional state. In the SD problem, the state is of the dimension of the grid alongside the robot pose. The number of grid cells can be arbitrarily large. In the SLAM problem, in the case of a binary grid map, the dimension is large since, typically, a satisfactory resolution is desired. In the case of continuous landmarks representation,

the robot gradually reveals more and more landmarks making the state increasingly large.

Since the belief is to be maintained over a high-dimensional state, it is not an easy task for an online operating robot. This computational challenge in the context of planning is known as *curse of dimensionality*. Moreover, with an increasing planning horizon, the number of possible measurements and candidate action sequences grows exponentially, assembling the computationally intractable decision making problem. This phenomenon is usually regarded as the *curse of history*. Many research efforts have targeted both *curses*.

Since typical high-dimensional BSP problems hold an enormous computational burden, many methods exist to reduce computational complexity and find an approximately optimal solution. Let us mention a few. In robotics, the abundance of possible future observations within the planning phase is often resolved by the maximum likelihood (ML) assumption. Originally suggested for low-dimensional BSP by Platt et al.[7], it was adopted to active SLAM [8], [9]. Yet, while widely used, taking into account merely the most likely measurements episode is highly unrealistic, particularly in the presence of significant uncertainty. It is possible that the largest available reward is not the most likely one, resulting in a substantial error in the objective estimate and, consequently, a suboptimal autonomous behavior. Stachniss et al. [10] sampled a single episode of possible future observations. One standing-out approach to use a number of sampled observations builds upon the reuse of calculations between successive planning sessions, alleviating the computational burden [11], [12]. Another approximation in a high-dimensional BSP setting done by [11] and [12] is to consider predefined static action sequences instead of policies. Interestingly, this approximation is also implicitly done by all methods utilizing ML observations or a single sample of the future observations episode. This is because under a single future observations episode assumption the candidate policy and predefined static action sequence are the same. One more method [3] along these lines leverages the structure of the belief over a high-dimensional state to speedup BSP and does not compromise performance at all. Notably, while the authors of [3] used ML assumption, it is not an inherent limitation of the approach. An additional example [13] is finding approximate POMDP solutions through belief compression. This approach was designed to reduce computational complexity for high-dimensional beliefs and policies, but works with expected state-dependent rewards and the extension to general belief-dependent rewards requires clarification.

The artificial intelligence (AI) community is also engaged in augmenting the classical POMDP formulation with belief-dependent rewards. The journey started from ρ-POMDP [14] and significantly advanced through time [15], [16], [17]. Commonly, these approaches seek to find an optimal policy instead of predefined static action sequence.

Recent methods, merging both worlds, build upon the *simplification* paradigm [18], [19], [20]. These simplification-based methods finally relax limiting assumptions, e.g., Gaussian belief, piecewise linearity, or Lipschitz continuity of the reward, and permitted universal belief-dependent rewards, such as differential entropy of general beliefs. Since the differential entropy operator acts over the belief, which can be parameterized in various ways, e.g., Gaussian or set of particles, questions of piecewise linearity, or Lipschitz continuity are vague and well defined only when the state is discrete and the number of possible state realizations is finite. In a continuous setting, they shall be approached individually for each belief parameterization. This fact discards many early approaches [14], [15] to include belief-dependent rewards within POMDP. Another line of *simplification* works alleviate the curse of dimensionality in the setting of multivariate Gaussian distributions utilizing sparsification [21], [22] and topological [23], [24] aspects. The simplification paradigm was also applied with Gaussian-mixture distributed beliefs [25], [26], [27].

Adaptivity is another important mechanism to identify redundancies in the decision making problem and reduce the computational effort [28].

All decision-making methods discussed above are concerned with selecting the best action and disregarding the actual amount of profit or risk entirely. However, the latter is essential, since preventing the robot from performing unnecessary or self-destructive operations is highly important. This gap can be filled by introducing constraints into the decision-making formulation. Some attempts to do so in the context of safe POMDPs include chance constraints [29].

A general belief-dependent constraint, however, has not received proper attention so far except in our previous work [30], where we focused on safety and comparison to chance constraints, and not on the Information gathering tasks. Note that chance constraints do not accommodate general belief-dependent operators such as Information Gain (IG).

In this article, we continue to investigate the facets of our proposed earlier framework [30] of belief-dependent probabilistically constrained continuous POMDP. Motivated by Information gathering, also called informative planning tasks, we focus on the cumulative form of the constraint in the realm of high-dimensional BSP. This is in contrast to the multiplicative form as in our previous article. One of the specific applications of our framework is stopping exploration. Moreover we provably extend the simplification framework to both forms of the constraints in our novel probabilistically constrained setting. The first form is cumulative and the second is multiplicative.

There are attempts to use differential entropy gain as a constraint to halt exploration in the problem of active SLAM [9], [31]. However, it was only partially investigated since algorithms solving BSP typically assume single observations episode [1], [3], [9], [10] to alleviate the computational burden. Stopping exploration is still regarded as an open problem [31]. Importantly, we did not find any works relaxing single observations episode assumption in the context of SD problem [3], [22], [32] and informative planning [1].

Our probabilistic belief-dependent constraint of cumulative form, which will become apparent later, generalizes previous approaches. The naive way to threshold a belief-dependent operator under partial observability is to do expectation with

respect to observations. However, even this has gained less attention so far and has not been done to the best of the authors' knowledge, due to the reason discussed above, single observations episode assumption. In contrast to expectation with respect to future observations, we propose a probabilistic condition. Our proposed variant is sensitive to the distribution of the belief-dependent constraint, which we call inner constraint, while averaging with respect to future observations is not.

As opposed to a threshold on expectation with respect to observations, we propose two conditions. Interior condition thresholds using δ the belief-dependent operator (return) for given sequence of possible future observations. The exterior condition verifies that the interior one is satisfied with confidence level of at least $1-\epsilon$. To rephrase it, we require that the fraction of the observation sequences samples fulfilling the interior condition will be at least $1-\epsilon$. In due course, we consider two different problem formulations. In the first problem, δ is specified externally by the user. We coin this problem as *optimality under a probabilistic constraint*. In the second problem, that we name *maximal feasible return*, δ is a free parameter to be maximized. In turn, our formulation and approach enable fast adaptive maximization of value at risk (VaR) on top of a general belief-dependent return. This problem is highly challenging due to the fact that VaR is not a coherent functional [33].

Our contributions are fourfold. Below we list them down in the same order as they are presented in the manuscript.

1) First, we utilize our probabilistically constrained belief-dependent POMDP in the context of an information-theoretic constraint. We focus on the IG, however, our theory supports any other belief-dependent operator, e.g., difference between traces of covariance matrices of two consecutive-in-time beliefs. We analyze the mutual information (MI) constraint and ML observation approach versus our novel probabilistic constraint. Notably, we did not find any works shifting the MI from the reward operator to the constraint.
2) Second, we rigorously derive a theory of *simplification* in the constrained setting. We emphasize that the simplification paradigm has not been considered in this setting before. Given a monotonically converging to the belief-dependent constraint or/and reward bounds, depending on context, our approach can be simplified, gaining substantial speedup without any loss in performance quality.
3) Third, we present an algorithm to maximize VaR adaptively utilizing the suggested theory. As we unveil in this article, this enables the decision maker to save time by adaptively expanding the lowest required number of observation episodes without compromising the quality of the solution.
4) Fourth, we apply our technique to a high-dimensional BSP. In particular, our case studies are active SLAM and SD problems.

The rest of this article is structured as follows. We start from background and notations in Section II. Section III presents our next step, that is, the in-depth discussion of the problem formulation and our approach. In Section IV, we then present an application of our methods. Section V presents the simulations and results. Finally, Section VI concludes this article.

II. Background and Notations

By the bold symbols, we denote time vector quantities; by $\square_{a:b}$, we mark series annotated by the time discrete indices running from a to b inclusive. The letter $\mathbb{P}$ symbolizes the probability density function (PDF) and P the probability. By lowercase letter we denote the random quantities or the realizations depending on the context. For brevity, we sometimes replace $\mathbb{E}_{\square}[\cdot|\cdot]$ by $\mathbb{E}_{\square|\cdot}[\cdot]$.

A. High-Dimensional BSP

Let us introduce the POMDP with belief-dependent rewards named ρ-POMDP alias to BSP. The ρ-POMDP is a tuple $\langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, \rho, \gamma, b_0 \rangle$ where $\mathcal{X}, \mathcal{A}$, and $\mathcal{Z}$ denote state, action, and observation spaces with $x \in \mathcal{X}$, $a \in \mathcal{A}$, and $z \in \mathcal{Z}$ the momentary state, action, and observation, respectively. $T(x', a, x) = \mathbb{P}_T(x'|x, a)$ is a stochastic transition model from the past state x to the subsequent x' through action a. Further, $\gamma \in (0, 1]$ is the discount factor, b_0 is the belief over the initial state (prior), and $\rho(b, b')$ is a general belief-dependent reward depending on two consecutive in time beliefs. For conciseness, let us denote interchangeably $\square_{k+}$ and $\square_{k:k+L-1}$, as well as $\square_{(k+1)+}$ and $\square_{k+1:k+L}$. This article deals with static action sequences of variable horizon L. Namely, our action space is $\mathcal{A} \triangleq \{a^i_{k:k+L^i-1}\}_{i=1}^{|\mathcal{A}|}$. Our actions along a particular action sequence are of different lengths. We also can think about such an action sequence as a path $\mathcal{P}$ comprising motion primitives. Yet, the action sequence is a much more general notion. So far, we have described the classical components of POMDP. However, in BSP, the observation model $O(\cdot)$ undergoes a customization that will be apparent later. For now, we leave it undefined.

An autonomous robot deployed in an environment (possibly unknown) repeatedly performs acting, sensing, and planning sessions, up until it reaches the required goal or *fails* to do so as we further formulate. In the planning phase, the robot relies on the entire action-observation history. Let $h_t \triangleq \{b_0, a_{0:t-1}, z_{1:t}\}$ be the history, i.e, the set comprising the performed by the agent actions $a_{0:t-1}$ and obtained observations $z_{1:t}$ in an interleaving manner up to time instant t, and the prior belief b_0. To clarify, we denote by t an arbitrary time instant and by k the time instant of the current planning session. Such that if $t \geq k$, the subscript t regards to future time. Another representation of history is the posterior belief. We define the posterior belief b_t as a shorthand for the PDF of the POMDP state, given all information up to time instant t. The state is denoted by $\boldsymbol{x}_t$ and the belief is $b_t(\boldsymbol{x}_t) \triangleq \mathbb{P}(\boldsymbol{x}_t|h_t)$. In this article the belief converts the history to a more convenient form, b_t and can be used interchangeably with h_t, as opposed to our previous work [19].

Frequently, in BSP problems, the robot's map is unknown and therefore regarded as a random quantity. This allows the robot to operate in unfamiliar environments. For the SLAM problem we opt for landmarks map representation, so the robot's state is $\boldsymbol{x}_t \triangleq (x_{0:t}, \{\ell^j\}_{j=1}^{M(k)})$, where $x_{0:t}$ are the robot poses,

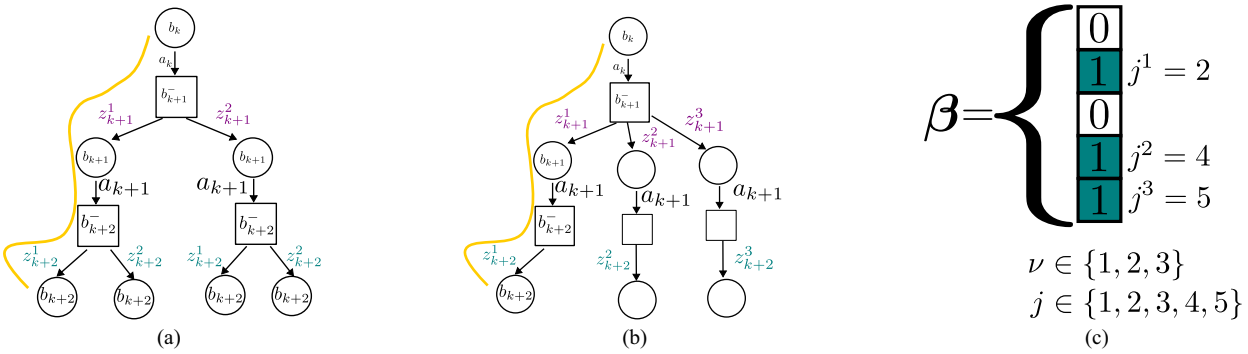


Fig. 1. Possible belief trees in continuous setting given $\boldsymbol{\beta}_{k+1:k+2}$. By **purple** and **teal** colors, we denote possibly different dimensionality of the observation as explained in Section II-B. Thick yellow lace illustrates the observation sequence $\boldsymbol{z}_{k+1:k+2}$ (Section III-E). (a) Visualization of the belief tree given the realization of $\boldsymbol{\beta}_{k+1:k+2}$ for action sequence a_{k+}. Here, we show two samples of observations per propagated belief. Superscript designates the child number. This belief tree supports policies and Bellman update. (b) In this belief tree the observation superscript designates the lace. (c) One possible realization of configuration is $\boldsymbol{\beta} = (01011)^T$.

$\{\ell^j\}_{j=1}^{M(k)}$ are the landmarks and $M(k)$ is the number of landmarks the robot has observed until time instant k inclusive. These landmarks represent the unknown robot's environment, specifically the map, to be estimated. To emphasize that j is not a time index, we denote it by a superscript instead of a subscript. Commonly, in SD problem the map is known. The robot moves over the known map divided into cells. Many works assume a deterministic transition model [1], [3], [22]. In contrast we do not make this assumption and formulate the SD problem as a complete POMDP with state comprising the robot position $x_t \in \mathbb{R}^2$ and the phenomenon of interest, vector $\boldsymbol{\xi} \in \mathbb{R}^N$. Overall, the POMDP state is $\boldsymbol{x}_t \triangleq (x_t, \boldsymbol{\xi}) = (x_t, \xi^{1:N})$. Note that for clear notation, cells in state are linearly indexed. The conversion from a Cartesian index to linear does not pose a problem. Let $\texttt{LinInd}(\cdot)$ be the function doing that.

B. Observation Configuration Random Vector and Model

In this section, we rigorously define a customized observation model in BSP. The dimension of the observation in BSP planning can vary in time. A typical reason for this variability is the finite visibility radius or sensing range of the robot. In a SLAM problem, the robot observes a subset of landmarks, whereas in a SD problem, the robot's position defines the observed cells, a subset of sensors yielding the reading of the phenomenon of interest. We denote by vector $\boldsymbol{\beta}$ the configuration of observed landmarks or cells. Let us start from SLAM.

1) $\boldsymbol{\beta}$ for Active SLAM: Let $\boldsymbol{\beta}_t \in \{0,1\}^{M(k)}$ be a random vector of Bernoulli variables, statistically independent given robot's pose x_t and a landmark, as will be shortly displayed by (1) and (2). Its dimensionality is the number of landmarks present in the belief. Each realization of $\boldsymbol{\beta}_t$ defines a subset of visible landmarks. Such a realization has ones at the indexes of visible landmarks and zeros else, such that $[\boldsymbol{\beta}]^j = 1 \, \forall j \in \{j^\nu\}_{\nu=1}^{n(\boldsymbol{\beta})}$, where $n(\boldsymbol{\beta}) = \sum_j [\boldsymbol{\beta}]^j$. (By $[\cdot]^j$ we indicate the coordinate j of a vector.) The superscript ν defines a subsequence of indices j^ν of visible landmarks [Fig. 1(c)]. Let us clarify, $j^1, j^2, \ldots$ represent, strictly increasing with ν, values of indexes of enumerated landmarks resulting in a random set $\{j^\nu\}_{\nu=1}^{n(\boldsymbol{\beta})}$, such that $j^\nu = j(\nu)$.

The mapping from the Boolean vector $\boldsymbol{\beta}$ to the random finite set of indices $\{j^1, j^2 \ldots\}$ is invertible. Therefore, one can define a probability over the random finite sets [34] instead of Boolean vectors.

One way to define a probabilistic model for visible landmarks configuration is as follows:

$$\begin{aligned} \mathrm{P}_{\boldsymbol{\beta}}\left([\boldsymbol{\beta}_t]^j = 1 | x_t, \ell^j\right) &= \mathbf{1}_{\{\|x_t - \ell^j\| \le r\}}\left(x_t, \ell^j\right) \\ \mathrm{P}_{\boldsymbol{\beta}}\left([\boldsymbol{\beta}_t]^j = 0 | x_t, \ell^j\right) &= 1 - \mathbf{1}_{\{\|x_t - \ell^j\| \le r\}}\left(x_t, \ell^j\right) \end{aligned} \tag{1}$$

where r is a visibility radius. Our approach is not limited to this specific model and supports any other model; for instance, in more complex scenarios (1) would imitate a camera field of view. Equation (1) portrays that each landmark deterministically has a visibility radius. If the robot is close enough, it receives a signal from the landmark. Overall we arrive at the following:

$$\mathrm{P}_{\boldsymbol{\beta}}\left(\boldsymbol{\beta}_t | x_t, \{\ell^j\}_{j=1}^{M(k)}\right) = \prod_{j=1}^{M(k)} \mathrm{P}_{\boldsymbol{\beta}}\left([\boldsymbol{\beta}_t]^j | x_t, \ell^j\right). \tag{2}$$

Here, we assumed that $t \ge k$ and the planner does not reveal new landmarks in a planning session, that is, $M(k)$ depends on the present time k but not the future time t. We define now a customized observation model for $n(\boldsymbol{\beta}) > 0$ as

$$O(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\beta}) \triangleq \mathbb{P}(\boldsymbol{z} | x, \{\ell^j\}_{j=1}^{M(k)}, \boldsymbol{\beta}) = \prod_{\nu=1}^{n(\boldsymbol{\beta})} \mathbb{P}_Z\left(z^\nu | x, \ell^{j^\nu}\right). \tag{3}$$

where x is the last robot pose in $\boldsymbol{x}$.

2) $\boldsymbol{\beta}$ for SD: As we mentioned above, in SD problem the variability of the dimension of observation stems from another source. The dimension of $\boldsymbol{\beta}$ is the number of cells. Vector $\boldsymbol{\beta}$ has one at the coordinates corresponding to the linear indexes (converted from Cartesian index) of the grid where active sensors yield an observation. The simplest model for $\boldsymbol{\beta}$ is as follows:

$$\begin{aligned} \mathrm{P}_{\boldsymbol{\beta}}([\boldsymbol{\beta}_t]^j = 1 | x_t) &= \mathbf{1}_{\{\texttt{LinInd}(\texttt{Cell}(x_t)) == j\}}(x_t) \\ \mathrm{P}_{\boldsymbol{\beta}}([\boldsymbol{\beta}_t]^j = 0 | x_t) &= 1 - \mathbf{1}_{\{\texttt{LinInd}(\texttt{Cell}(x_t)) == j\}}(x_t) \end{aligned} \tag{4}$$

describing that the observation is received from a single sensor at the cell of the robot location. The $\texttt{Cell}(x_t)$ function returns Cartesian indices of the cell there the robot is located. Overall we have that

$$\mathrm{P}_{\boldsymbol{\beta}}\left(\boldsymbol{\beta}_t|x_t\right)=\prod_{j=1}^{N}\mathrm{P}_{\boldsymbol{\beta}}([\boldsymbol{\beta}_t]^j|x_t). \tag{5}$$

The observation model for $n(\boldsymbol{\beta})>0$ materializes as

$$O(\boldsymbol{z},\boldsymbol{x},\boldsymbol{\beta})\triangleq\mathbb{P}(\boldsymbol{z}|x,\boldsymbol{\xi},\boldsymbol{\beta})=\mathbb{P}_Z(z^x|x)\prod_{\nu=1}^{n(\boldsymbol{\beta})}\mathbb{P}_Z(z^\nu|x,[\boldsymbol{\xi}]^{j^\nu}). \tag{6}$$

Now, we turn to the BSP objective to be maximized.

C. Objective

A common BSP objective is given by the following:

$$\mathcal{U}(b_k,a_{k+};\rho)=\mathbb{E}_{\boldsymbol{\beta}_{(k+1)+}}\left[\mathcal{U}^{\boldsymbol{\beta}_{(k+1)+}}(b_k,a_{k+};\rho)\big|b_k,a_{k+}\right] \tag{7}$$

where $\mathcal{U}^{\boldsymbol{\beta}_{(k+1)+}}(b_k,a_{k+};\rho)$ is

$$\mathbb{E}_{\boldsymbol{z}_{(k+1)+}}\left[\sum_{t=k}^{k+L-1}\rho(b_t,b_{t+1})\big|b_k,a_{k+},\boldsymbol{\beta}_{(k+1)+}\right] \tag{8}$$

and where t is the running time index and k is the present time instant. The inner expectation $\mathcal{U}^{\boldsymbol{\beta}_{(k+1)+}}(b_k,a_{k+};\rho)$ [see possible belief trees in Fig. 1(a) and (b)] corresponds to the utility given a static set of visible landmarks (SLAM problem) or active sensors (SD problem), or another constellation of parameters depending on the considered problem. Therefore, conditioned on a sequence $\boldsymbol{\beta}_{(k+1)+}$, per time index, the dimension of the observation is fixed (It can be different, however, for different time indices). Thus, the expectation operator is well defined. The outer expectation performs an average of such values, weighted in terms of $\boldsymbol{\beta}_{(k+1)+}$ [Fig. 1(c)]. Note that while it is appealing to fold the conditional expectations in (8) using the law of total expectation, we cannot do that since the dimension of the observation $\boldsymbol{z}_t$ depends on each specific realization of $\boldsymbol{\beta}_t$.

To summarize this section, BSP accommodates continuous spaces and varying dimension of observation conditioned on state. To verify our algorithms in different scenarios we will simulate both trees depicted in Fig. 1(a) and (b).

III. Problem Formulation and Approach

In this work, we define and tackle two novel problems. Both problems are explicitly aware of the distribution stemming from future observations and, therefore, are risk-aware.

A. Introducing Distribution Awareness into BSP

Our *first* problem formulation is the *optimality under a probabilistic constraint*

$$a^*\in\arg\max_{a_{k+}\in\mathcal{A}}\mathcal{U}(b_k,a_{k+};\rho)\ \text{subject to}$$
$$\mathrm{P}(c(b_{k:k+L};\phi,\delta)=1|b_k,a_{k+})\geq 1-\epsilon \tag{9}$$

where c is the indicator variable over inner condition, as we will shortly see, ϕ is the general belief-dependent operator, and δ and $0\leq\epsilon<1$ are scalars. The utility $\mathcal{U}$ in (9) conforms to (7). The parameters δ and ϵ are supplied by the user.

The inner expression $c(b_{k:k+L};\phi,\delta)$ in (9) can be of two forms. The first (cumulative) form is as follows:

$$c(b_{k:k+L};\phi,\delta)\triangleq\mathbf{1}_{\left\{\left(\sum_{t=k}^{k+L-1}\phi(b_t,b_{t+1})\right)>\delta\right\}}(b_{k:k+L}) \tag{10}$$

and the second (multiplicative) is

$$c(b_{k:k+L};\phi,\delta)\triangleq\prod_{t=k}^{k+L}\mathbf{1}_{\{\phi(b_t)\geq\delta\}}(b_t). \tag{11}$$

Note, the strict inequality marked by the red color in (10). Further, let us refer to the inner inequality as the inner constraint and correspondingly the outer inequality (9) as the probabilistic (outer) constraint. From now on, let us denote *constraining* return and the *actual* return operators as $s(b_{k:k+L};\phi)\triangleq\sum_{t=k}^{k+L-1}\phi(b_t,b_{t+1})$ and $s(b_{k:k+L};\rho)\triangleq\sum_{t=k}^{k+L-1}\rho(b_t,b_{t+1})$, respectively. To encapsulate both cases ρ and ϕ we will denote $s(b_{k:k+L};\cdot)$.

Now, we contemplate what will happen, if δ is a free parameter and not predetermined as before. In this case we would like to select action sequence corresponding to largest maximal feasible return [actual or constraining $s(b_{k:k+L};\cdot)$] with probability of at least $1-\epsilon$. That is, maximal δ yielding that, at most, a single action sequence is feasible. With this insight in mind, we arrive at our *second* problem formulation, which we named *maximal feasible return* defined as follows:

$$a^*\in\arg\max_{a_{k+}\in\mathcal{A}}\mathcal{V}(b_k,a_{k+};\epsilon) \tag{12}$$

where the VaR expressed by $\mathcal{V}(b_k,a_{k+};\epsilon)=\mathrm{VaR}_\epsilon(s(b_{k:k+L};\cdot)|b_k,a_{k+})$ defined by

$$\sup\{\delta:\mathrm{P}(s(b_{k:k+L};\cdot)\geq\delta|b_k,a_{k+})\geq 1-\epsilon\}. \tag{13}$$

It is noteworthy that in (13), we have nonstrict inner inequality $\geq\delta$ (marked by the red color). We will need it further in our approach. In contrast, in (10) the inequality involving δ is strict. This aspect will be clear in the sequel. Moreover, inclusion to or exclusion from the set in (13) of the δ that satisfies $\mathrm{P}(s(b_{k:k+L};\cdot)=\delta|b_k,a_{k+})\geq 1-\epsilon$ does not impact the outcome of supremum operator in (13).

Due to noncompliance to Bellman form of (13) computing (12) is notoriously challenging.

B. Averaging With Respect to Observations

Another way to introduce a belief-dependent constraint to POMDP would be by averaging with respect to observations. Namely, the probabilistic constraint in (9) is replaced by the condition $\mathcal{C}(b_k,a_{k+};\phi)>\delta$ (Note also here that the inequality is strict) given by

$$\mathcal{C}(b_k,a_{k+};\phi)=\mathbb{E}_{\boldsymbol{\beta}_{(k+1)+}}\left[\mathcal{C}^{\boldsymbol{\beta}_{(k+1)+}}(b_k,a_{k+};\phi)\big|b_k,a_{k+}\right]>\delta \tag{14}$$

where $\mathcal{C}^{\boldsymbol{\beta}_{(k+1)+}}(b_k, a_{k+}; \phi)$ equals to

$$\mathbb{E}_{\boldsymbol{z}_{(k+1)+}}\left[\sum_{t=k}^{k+L-1}\phi(b_t, b_{t+1})\big|b_k, a_{k+}, \boldsymbol{\beta}_{(k+1)+}\right]. \tag{15}$$

However, if one transfers the utility (7) to the constraint, in other words, when $\rho(\cdot)\equiv\phi(\cdot)$ such a constraint appears to be problematic. If $\mathcal{U}(\cdot)\equiv\mathcal{C}(\cdot)$, we can always maximize the utility and ask if the optimal utility is larger than δ (i.e., $\mathcal{U}^* > \delta$). In case that $\max_{a_{k+}\in\mathcal{A}}\mathcal{U}(b_k, a_{k+}; \rho) \leq \delta$, no feasible action sequence exists in $\mathcal{A}$. In general, this is the question of what one verifies first, *optimality* or *feasibility*. As we shall further see, in some cases the order does matter and we can save time by a fast feasibility check and cancellation of action sequences.

One important operator related to the averaging with respect to observations is MI. Assume that we can deduce $\boldsymbol{\beta}$ from the corresponding observation $\boldsymbol{z}$. In this case $b_t(\boldsymbol{x}_t) = \mathbb{P}(\boldsymbol{x}_t|h_t, \boldsymbol{\beta}_{1:t})$. We shed light on this fact in Section IV-A. Using this assumption, we can write (15) as $\sum_{t=k}^{k+L-1}\mathop{\mathbb{E}}_{\boldsymbol{z}_{k+1:t}}[\mathop{\mathbb{E}}_{\boldsymbol{z}_{t+1}|b_t,a_t,\boldsymbol{\beta}_{t+1}}[\phi(b_t, b_{t+1})]\big|b_k, a_{k+}, \boldsymbol{\beta}_{k+1:t}]$. Assume also that the belief is over the last robot pose and some static-in-time random term, e.g., map in SLAM or phenomenon of interest in SD. Let's call this static-in-time random term χ. Recall, that in our formulation of SLAM the robot does not reveal new landmarks in a planning session, so the map is static-in-time within planning. In SD the map is known and, therefore, is not part of the state. Suppose a myopic setting and define

$$\mathop{\mathbb{E}}_{\boldsymbol{z}_{k+1}|b_k,a_{k+},\boldsymbol{\beta}_{k+1}}[\phi(b_k, b_{k+1})] \triangleq \mathrm{MI}(x_{k+1}, \chi; \boldsymbol{z}_{k+1}|b_k, a_{k+}, \boldsymbol{\beta}_{k+1}) = \mathop{\mathbb{E}}_{\boldsymbol{z}_{k+1}|b_k,a_{k+},\boldsymbol{\beta}_{k+1}}[-\mathrm{h}(b_{k+1})] + \mathrm{h}(\mathbb{P}(x_{k+1}, \chi|b_k, a_{k+}, \boldsymbol{\beta}_{k+1})) \tag{16}$$

where the differential entropy of the belief $\mathrm{h}(b)$ is given by

$$\mathrm{h}(b) \triangleq -\int_x b(\boldsymbol{x})\log b(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{17}$$

We see that (16) is always nonnegative due to $\mathrm{MI}(\cdot) \geq 0$. In addition, differential entropy does not have units. At this point, we arrive to the question of selecting a meaningful δ. Thanks to the strict inequality in (14), we can set $\delta = 0$ and catch and discard the action sequences where the observations are statistically independent from the state. This is highly unlikely, however, that all the candidate action sequences will be not feasible. Therefore, such a constraint hardly can serve as a stopping exploration criterion.

If the robot is fully observable and the belief is solely over the fixed-in-time-term χ as in SD, by defining ϕ as IG in the most common sense

$$\phi(b, b') = \mathrm{IG}(b, b') = -\mathrm{h}(b') + \mathrm{h}(b) \tag{18}$$

we obtain a telescopic series in (15) and (15) equals to

$$\mathop{\mathbb{E}}_{\boldsymbol{z}_{(k+1)+}|b_k,a_{k+},\boldsymbol{\beta}_{(k+1)+}}[-\mathrm{h}(b_{k+L})+\mathrm{h}(b_k)] = \mathrm{MI}(\chi; \boldsymbol{z}_{k+1}|b_k, a_{k+}, \boldsymbol{\beta}_{(k+1)+}). \tag{19}$$

We again observe that to define a meaningful δ besides $\delta = 0$ and stop to explore will be problematic also here.

Let us now consider the belief is over the whole robot trajectory and the fixed-in-time random term χ. If we utilize (18), we obtain a telescopic series in (15), which becomes

$$\begin{aligned}\mathbb{E}_{\boldsymbol{z}_{(k+1)+}}[-\mathrm{h}(b_{k+L}) + \mathrm{h}(b_k)|b_k, a_{k+}, \boldsymbol{\beta}_{(k+1)+}] \\ = \mathrm{MI}\left(x_{0:k+L}, \chi; \boldsymbol{z}_{(k+1)+}\big|b_k, a_{k+}, \boldsymbol{\beta}_{(k+1)+}\right) \\ + \mathrm{h}\left(\mathbb{P}(x_{0:k+L}, \chi|b_k, a_{k+}, \boldsymbol{\beta}_{(k+1)+}\right).\end{aligned} \tag{20}$$

Here, with $\delta = 0$ the robot can stop to explore if all candidate actions yield $\mathop{\mathbb{E}}_{\boldsymbol{z}_{(k+1)+}|b_k,a_{k+},\boldsymbol{\beta}_{(k+1)+}}[-\mathrm{h}(b_{k+L})+\mathrm{h}(b_k)] \leq 0$. This is because of the additional to $\mathrm{MI}(\cdot)$ term in (20).

Now, we see the purpose of the strict inequality in (10). This is to allow the robot to explore only if the cumulative IG is nonnegative ($\delta = 0$). We continue to debate the matter of selecting δ in Section IV-C.

C. Single Observation Sample Approximation

Another option would be to use a ML episode of observations $\boldsymbol{z}^{\mathrm{ML}}_{k+1:k+L}$ and check $(\sum_{t=k}^{k+L-1}\phi(b_t, a_t, \boldsymbol{z}^{\mathrm{ML}}_{t+1}, b_{t+1}))>\delta$, where the ML observation $\boldsymbol{z}^{\mathrm{ML}}_{t+1}$ is obtained as follows. We start from a ML state $\boldsymbol{x}^{\mathrm{ML}}_{t+1}\in\arg\max_{\boldsymbol{x}_{t+1}}\mathbb{P}(\boldsymbol{x}_{t+1}|b_t, a_t)$, and then find $\boldsymbol{\beta}^{\mathrm{ML}}\in\arg\max_{\boldsymbol{\beta}_{t+1}}\mathrm{P}_{\boldsymbol{\beta}}(\boldsymbol{\beta}_{t+1}|\boldsymbol{x}^{\mathrm{ML}}_{t+1})$ (see Appendix A). This, in turn, results in $\boldsymbol{z}^{\mathrm{ML}}_{t+1} \in \arg\max_{\boldsymbol{z}_{t+1}}\mathbb{P}(\boldsymbol{z}_{t+1}|\boldsymbol{x}^{\mathrm{ML}}_{t+1}, \boldsymbol{\beta}^{\mathrm{ML}}_{t+1})$. The ML assumption approximates the observations episode likelihood as

$$\mathbb{P}\left(\boldsymbol{z}_{(k+1)+}|b_k, a_{k+}\right) = \delta\left(\boldsymbol{z}_{(k+1)+} - \boldsymbol{z}^{\mathrm{ML}}_{(k+1)+}\right) \tag{21}$$

where $\delta(\cdot)$ is Dirac delta function. Note that the probability in (13) can be written as

$$\begin{aligned}\int_{\boldsymbol{z}_{(k+1)+}}\mathrm{P}\left(\{s(b_{k:k+L};\cdot) \geq \delta\}|b_k, \boldsymbol{z}_{(k+1)+}, a_{k+}\right)\cdot \\ \mathbb{P}(\boldsymbol{z}_{(k+1)+}|b_k, a_{k+})\mathrm{d}\boldsymbol{z}_{(k+1)+} = \int_{\boldsymbol{z}_{(k+1)+}}\mathbf{1}_{\{s(b_{k:k+L};\cdot)\geq\delta\}}(b_{k+}) \\ \mathbb{P}(\boldsymbol{z}_{(k+1)+}|b_k, a_{k+})\mathrm{d}\boldsymbol{z}_{(k+1)+}.\end{aligned} \tag{22}$$

Plugging (21), this in turn yields the degeneration of the probability in (13) to $\mathbf{1}_{\{s(b_{k:k+L};\cdot)\geq\delta\}}(b^{\mathrm{ML}}_{k:k+L})$. In this case, the set in (13) is $\{\delta : \mathbf{1}_{\{s(b_{k:k+L};\cdot)\geq\delta\}}(b^{\mathrm{ML}}_{k:k+L}) \geq 1-\epsilon\}$, so if $0 \leq \epsilon<1$ the set above is $\{\delta{:}\delta \geq s(b^{\mathrm{ML}}_{k:k+L};\cdot)\}$ and $\sup\{\delta{:}\delta \leq s(b^{\mathrm{ML}}_{k:k+L};\cdot)\} = s(b^{\mathrm{ML}}_{k:k+L},\cdot)$. We conclude that under the ML assumption the expected return is equivalent to VaR with any confidence level $\epsilon \in [0, 1)$. In fact, this applies for any single sample approximation. We can conclude that using single sample approximation prevents the application of distribution aware operators, such that VaR or conditional VaR (CVaR).

D. Comparison

Now we are back to our distribution aware setting. We can interpret the difference between expected constraint (15) and

our probabilistic risk-aware constraint (9) as follows. The conventional constraint is unaware of the distribution of the cumulative values of operator ϕ. It decides whether the constraint is fulfilled or not solely using the expected value. The constraint's expected value may fail to represent the underlying distribution adequately. In contrast, our formulation is *distribution aware*. We explicitly regard the distribution of future laces of the beliefs using parameters ϵ and δ.

In the following sections, we develop a universal theory to evaluate the sample approximation of our proposed probabilistic inequality (9) *adaptively*. On top of that, we expedite the evaluation process even more by extending the *simplification* paradigm to our setting, enjoying the substantially improved celerity versus baseline approaches.

E. Adaptive Belief Tree

In reality to evaluate our probabilistic constraint in (9) we shall marginalize over observation episodes, leverage that $\mathrm{P}(c(b_{k:k+L};\phi,\delta)=1|b_k,a_{k+},\boldsymbol{z}_{(k+1)+})=c(b_{k:k+L};\phi,\delta)$ and solve

$$\int_{\boldsymbol{z}_{(k+1)+}} c(b_{k:k+L};\phi,\delta)\mathbb{P}(\boldsymbol{z}_{(k+1)+}|b_k,a_{k+})\mathrm{d}\boldsymbol{z}_{(k+1)+}. \quad (23)$$

The integral in (23) is not accessible in a general setting. One way to approximately evaluate the (23) is to sample from observation likelihood $\mathbb{P}(\boldsymbol{z}_{(k+1)+}|b_k,a_{k+})$. We assume that we have a fixed budget m of samples of observation laces. Our aim is to use the fact that we have a particular structure of the probabilistic condition (23) and to address its evaluation while constructing the belief tree, thereby saving valuable running time or providing a more accurate solution.

Imagine a candidate action sequence $a_{k:k+L-1}$. To approximate the utility and the probabilistic constraint (9), an online algorithm at the root (for each candidate action sequence) expands upon termination m laces appropriate to the drawn observations $\{\boldsymbol{z}^l_{k+1:k+L}\}_{l=1}^m$. Through the article we label the laces in the belief tree by the superscript l [yellow thick lace in Fig. 1(a) and (b)]. Each lace l corresponds to a particular realization of the sequence of the beliefs, return $s(b_{k:k+L};\rho)$ or constraining return $s(b_{k:k+L};\phi)$. The sample approximation of (23) from m laces is

$$\hat{\mathrm{P}}^{(m)}(c(b_{k:k+L};\phi,\delta)=1|b_k,a_{k+})=\frac{1}{m}\sum_{l=1}^{m}c(b^l_{k:k+L};\phi,\delta) \quad (24)$$

and the outer constraint in (9) becomes

$$\frac{1}{m}\sum_{l=1}^{m}c(b^l_{k:k+L};\phi,\delta)\geq 1-\epsilon. \quad (25)$$

We employ an already expanded part of the belief tree with $\tilde{m}$ laces to bound the expression of the probabilistic constraint (24) from each end using the following adaptive lower bound

$$\underbrace{\frac{1}{m}\sum_{l=1}^{\tilde{m}}c(b^l_{k:k+L};\phi,\delta)}_{\mathrm{lb}^{(1)}}\leq\frac{1}{m}\sum_{l=1}^{m}c(b^l_{k:k+L};\phi,\delta) \quad (26)$$

and the upper bound

$$\frac{1}{m}\sum_{l=1}^{m}c(b^l_{k:k+L};\phi,\delta)\leq\overbrace{\frac{m-\tilde{m}}{m}+\frac{1}{m}\sum_{l=1}^{\tilde{m}}c(b^l_{k:k+L};\phi,\delta)}^{\mathrm{ub}^{(1)}} \quad (27)$$

where, the algorithm already expanded $\tilde{m}\leq m$ laces. By adaptivity, we mean the expanding lowest number of laces $\tilde{m}$ to accept or discard the candidate action sequence.

F. Adaptive Simplified Constraint Evaluation

As introduced in [18], [19], [21], and [25], the simplification paradigm seeks to ease the computational burden in the decision making problem, while providing performance guarantees. The latter is achieved by applying bounds over various quantities in the decision making problem (e.g., bounds over a reward function). In this section, we extend this concept to our probabilistic belief-dependent constrained POMDP setting (9) and (12).

Suppose we have adaptive deterministic bounds over ϕ, i.e., these bounds hold for any realization of the beliefs. Further, evaluating these bounds is computationally cheaper than the operator ϕ. One example of such bounds can be found in [18] and [20]. Let us present the main theorem of this section, which will shed light on how these bounds can be utilized, propagating their adaptivity further to the adaptive probabilistic constraint evaluation.

Theorem 1 (Simplification machinery): Imagine a sampled set of the observations laces $\{\boldsymbol{z}^l_{k+1:k+L}\}_{l=1}^m$. Assume that $\forall l$

$$\underline{\phi}\left(b^l_{\ell+1},b^l_{\ell}\right)\leq\phi\left(b^l_{\ell+1},b^l_{\ell}\right)\leq\overline{\phi}\left(b^l_{\ell+1},b^l_{\ell}\right). \quad (28)$$

Let two forms of sampled inner constraint bounds variants be

$$\overline{c}\left(b^l_{k:k+L};\overline{\phi},\delta\right)\triangleq\mathbf{1}_{\left\{\left(\sum_{t=k}^{k+L-1}\overline{\phi}(b_{t+1},b_t)\right)>\delta\right\}}\left(b^l_{k:k+L}\right) \quad (29)$$

$$\underline{c}\left(b^l_{k:k+L};\underline{\phi},\delta\right)\triangleq\mathbf{1}_{\left\{\left(\sum_{t=k}^{k+L-1}\underline{\phi}(b_{t+1},b_t)\right)>\delta\right\}}\left(b^l_{k:k+L}\right) \quad (30)$$

for cumulative form (10) and

$$\overline{c}(b^l_{k:k+L};\overline{\phi},\delta)\triangleq\prod_{t=k}^{k+L}\mathbf{1}_{\left\{\overline{\phi}(b_t)\geq\delta\right\}}(b^l_t) \quad (31)$$

$$\underline{c}(b^l_{k:k+L};\underline{\phi},\delta)\triangleq\prod_{t=k}^{k+L}\mathbf{1}_{\left\{\underline{\phi}(b_t)\geq\delta\right\}}(b^l_t) \quad (32)$$

for multiplicative (11). Equation (28), in turn, implies that the following inequalities are satisfied without dependency on the

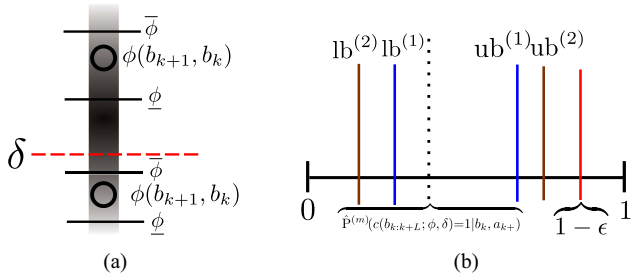


Fig. 2. (a) Conceptual visualization of our *simplification* approach (Section III-F). For clarity we show a myopic setting. Gradient displays the PDF, i.e., a larger number of samples lands in the area of greater intensity. Using the bounds, we want to assess whether the fraction of the sampled observation laces above δ is at least $1-\epsilon$. As we see, we can invalidate the bottom sample ϕ using solely the upper bound $\overline{\phi}$. In a similar manner, we can validate the upper sample ϕ using solely the lower bound $\underline{\phi}$. Note that the width of the vertical strip has no role in this visualization. (b) Simplification approach in this article delegates the bounds over ϕ to the second layer bounds $\mathrm{lb}^{(2)}$ and $\mathrm{ub}^{(2)}$.

form and for any m:

$$\sum_{l=1}^{m} \underline{c}\left(b^l_{k:k+L}; \underline{\phi}, \delta\right) \leq \sum_{l=1}^{m} c\left(b^l_{k:k+L}; \phi, \delta\right) \leq \sum_{l=1}^{m} \overline{c}\left(b^l_{k:k+L}; \overline{\phi}, \delta\right). \tag{33}$$

Importantly, this result holds with strict inequality in (10), (29), and (30) denoted by the red color and nonstrict.

We provide a detailed proof of Theorem 1 in Appendix B.

Let us now show how to speed up the process of evaluation of the probabilistic constraint from (9). The key component of the acceleration is that the adaptivity of the bounds (28) is delegated to adaptivity of the probabilistic constraint bounds (33). Assume the bounds from (28) are adaptive, using insights provided by Theorem 1, we first check if

$$\frac{1}{m}\sum_{l=1}^{m} \underline{c}\left(b^l_{k:k+L}; \underline{\phi}, \delta\right) \overset{?}{\geq} 1-\epsilon. \tag{34}$$

If the above relation holds (marked by ?), we declare that the outer constraint is fulfilled. Else, we probe if

$$\frac{1}{m}\sum_{l=1}^{m} \overline{c}\left(b^l_{k:k+L}; \overline{\phi}, \delta\right) \overset{?}{<} 1-\epsilon. \tag{35}$$

If yes, we declare that the outer constraint is violated. In case we are not able to say anything (both relations do not hold), we tighten the bounds. In other words, we make the bounds closer to the actual value of ϕ (e.g., by utilizing more particles [19], [20] or mixture belief components [25]). We presented a visualization of our simplification approach in Fig. 2.

Now our goal is to merge the insights gained in Section III-E with the simplification. Clearly, from (26) and by substituting m by $\tilde{m}$ in the left-hand side (LHS) of (33) we have that

$$1-\epsilon \overset{?}{\leq} \underbrace{\frac{1}{m}\sum_{l=1}^{\tilde{m}} \underline{c}(b^l_{k:k+L}; \underline{\phi}, \delta)}_{\mathrm{lb}^{(2)}} \leq \underbrace{\frac{1}{m}\sum_{l=1}^{\tilde{m}} c(b^l_{k:k+L}; \phi, \delta)}_{\mathrm{lb}^{(1)}}. \tag{36}$$

Similarly from (27) and right-hand side (RHS) of (33) the following holds:

$$\overbrace{\frac{m-\tilde{m}}{m} + \frac{1}{m}\sum_{l=1}^{\tilde{m}} c(b^l_{k:k+L}; \phi, \delta)}^{\mathrm{ub}^{(1)}} \leq$$
$$\overbrace{\frac{m-\tilde{m}}{m} + \frac{1}{m}\sum_{l=1}^{\tilde{m}} \overline{c}(b^l_{k:k+L}; \overline{\phi}, \delta)}^{\mathrm{ub}^{(2)}} \overset{?}{<} 1-\epsilon. \tag{37}$$

By a question mark, we denote the inequalities that shall be fulfilled online to check whether the outer constraint is met (36) or violated (37). If we cannot incur the status of the outer constraint we shall add more laces (adapt the first layer bounds $\mathrm{lb}^{(1)}, \mathrm{ub}^{(1)}$) or/and tighten the bounds from (28) (adapt the second layer bound $\mathrm{lb}^{(2)}, \mathrm{ub}^{(2)}$). Such an approach permits adaptive evaluation of the sample approximation of probabilistic constraint in (9) manifested by (24) before expanding the m laces of the belief sequences $b_{k:k+L}$. After a finite number of adaptation steps and smaller than or equal to m we guaranteed to evaluate (25) in the exact way. Specifically, only one of the inequalities (36) and (37) will be satisfied with some $\tilde{m}$. We validate (25) using the lower bound (36) or invalidate it using the upper bound (37). Using $\mathrm{lb}^{(1)}, \mathrm{ub}^{(1)}$, we save time that would be spent on the $m-\tilde{m}$ laces that would be expanded if one continues to sample the observation episodes (laces) up until the budget of samples is reached, namely, m laces. In addition, using $\mathrm{lb}^{(2)}, \mathrm{ub}^{(2)}$, we save time required to calculate the actual operator ϕ instead of the bounds (28) for the expanded $\tilde{m}$ laces.

G. Adaptation

It occurs that the proposed bounds have riveting properties. To describe a pair of lower $(\mathrm{lb}^{(1)}, \mathrm{lb}^{(2)})$ and a pair of upper bounds $(\mathrm{ub}^{(1)}, \mathrm{ub}^{(2)})$ simultaneously, we omit the superscript. The lower bound is bounded by zero $0 \leq \mathrm{lb}$ from below and the upper bound is bounded by one $\mathrm{ub} \leq 1$ from above. When we adapt the bounds, we add at most a single lace to the appropriate sum. Therefore, the step of adaptation of the bounds is at most $1/m$. When we expand a single lace $\tilde{m} \leftarrow \tilde{m}+1$, the lower bound makes a step if $c(b^l_{k:k+L}; \phi, \delta) = 1$, otherwise, the upper bound makes a step if $c(b^l_{k:k+L}; \phi, \delta) = 0$. Alternatively, when we increase the simplification level, some already expanded laces possibly switch from 0 to 1 ($\underline{c}(b^l_{k:k+L}; \underline{\phi}, \delta)$ for some l), contracting the lower bound, and some from 1 to 0 ($\overline{c}(b^l_{k:k+L}; \overline{\phi}, \delta)$ for some l), tightening the upper bound.

Importantly, when we expand a single observation lace and calculate $\underline{c}(b^l_{k:k+L}; \underline{\phi}, \delta)$ we will obtain one with probability at most $\mathrm{P}(c(b_{k:k+L}; \underline{\phi}, \delta) = 1|b_k, a_{k+})$. Similarly, we will obtain $\overline{c}(b^l_{k:k+L}; \overline{\phi}, \delta) = 0$ at the new expanded lace with probability at most $\mathrm{P}(c(b_{k:k+L}; \phi, \delta) = 0|b_k, a_{k+})$. Both these probabilities are not accessible.

Further, we have four scenarios illustrated in Fig. 3. By analyzing these scenarios, we can speculate about anticipated speedup. In Fig. 3 we show by the red vertical line several

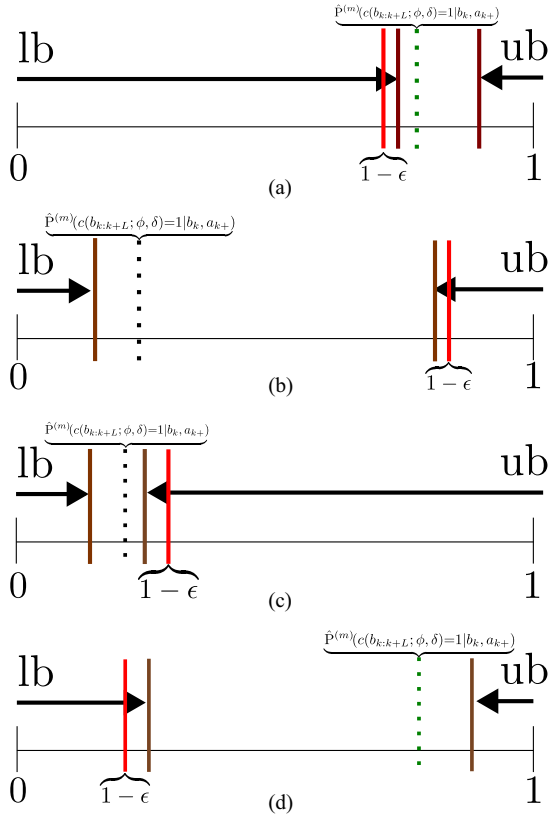


Fig. 3. Visualization of adaptation from Section III-G. Note, in all scenarios the value of dashed line is *unknown*. Red line represents the confidence level $1-\epsilon$ to be satisfied with probabilistic constraint. (a) Conceptual illustration of a challenging scenario. To *accept* such an action the lower bound shall go a long way. (b) Conceptual illustration of an easy scenario, with a few contractions of the upper bound, the action is *discarded*. (c) Another interesting situation, here the upper bound shall go a long way to *discard* the action sequence. (d) With a few shrinkage iterations the lower bound accepts the action sequence.

positions of the outer threshold $1-\epsilon$ from (9). The first scenario, shown in Fig. 3(a), is challenging. The value of (24) [shown by green dashed vertical line in Fig. 3(a)] is unavailable to us before the expansion of the m laces; therefore, no matter how many iterations we perform, invalidation using the calculated ub and (37) is not possible before reaching the budget of the m laces; only validation using lb and (36) will eventually be possible. As we observe, many contractions of the lb would be required, as we see in Fig. 3(a) up until lb becomes larger than $1-\epsilon$ according to (36). Conversely, if with a *large margin* the outer constraint is violated, as we see in Fig. 3(b), we discard the action sequence with a few tightening iterations using ub and (37). Note, the $\mathrm{P}(c(b_{k:k+L};\phi,\delta)=0|b_k,a_{k+})$ is large in this case. We contemplate a similar behavior in reciprocal cases [Fig. 3(c) and (d)]. To conclude the adaptation can be challenging in cases described in Fig. 3(a) and (c).

The fact that we have a pair of lower $(\mathrm{lb}^{(1)},\mathrm{lb}^{(2)})$ and a pair of upper bounds $(\mathrm{ub}^{(1)},\mathrm{ub}^{(2)})$ raises the question, which bound from a pair shall we adapt if a pair is inconclusive. When we cannot incur whether the outer constraint from (25) is fulfilled, we shall decide to refine the bounds $(\mathrm{lb}^{(2)},\mathrm{ub}^{(2)})$ or add more laces of observation episodes (refine $\mathrm{lb}^{(1)},\mathrm{ub}^{(1)}$). Luckily for us, these two operations are parallelizable via multithreading. We simultaneously refine the simplification levels, as in [18] of the bounds, and add more laces up until the decision is possible. Note that it will be problematic to parallelize (25) with respect to m laces. Due to the high dimensionality of the belief it will require an enormous memory capacity to hold all the m laces of the beliefs simultaneously. In fact, even taking into account sparsity aspects in SLAM, the number of variables is extremely large in real world applications. In the SD problem, the Information matrix is not anticipated to be sparse due to prior belief. Let us also mention that m shall be as large as possible due to the fact that larger m will increase the quality of sample approximation pictured by (24).

To conclude this section, we proposed a two-layered approach to ease the computational burden. The first layer expresses adaptivity in terms of the number of observation laces. The second layer permits utilization of the adaptive deterministic bounds on realizations of ϕ.

One example of using our technique is to save time in open loop planning or spend more time on the action sequences which fulfill the probabilistic constraint. With such an approach, we are able to cut down on the cost of exhaustively validating candidate action sequences without any sacrifice in performance. Another example is the closed loop setting, where we deal with policies. This is, however, out of the scope of this article.

Thus far, we presented general theory, and now we specifically address the second formulated problem (12).

H. Maximal Feasible Return

In this section, we develop an adaptive approach to identify an action sequence and δ maximizing (25) for both flavors of the inner constraint, i.e., cumulative (10) and multiplicative (10). Yet, in this article we focus on maximizing the cumulative form, which is motivated by IG along the planning horizon. Our goal is to solve the sample approximation from m laces of the formulated problem we named maximal feasible return (12). Picture in your mind that you guess the δ and the step size Δ. For clarity we drop the dependence of s on $b_{k:k+L}$. However, we shall remember that a single realization of s corresponds to a single lace in the belief tree [Fig. 1(a) and (b)]. Observe the following pair of relations:

$$\hat{\mathrm{P}}^{(m)}\left(s\geq\delta|b_k,a_{k+}\right)\geq\hat{\mathrm{P}}^{(m)}\left(s\geq\delta+\Delta|b_k,a_{k+}\right) \tag{38}$$

$$\hat{\mathrm{P}}^{(m)}\left(s\geq\delta|b_k,a_{k+}\right)\leq\hat{\mathrm{P}}^{(m)}\left(s\geq\delta-\Delta|b_k,a_{k+}\right) \tag{39}$$

where $\hat{\mathrm{P}}^{(m)}(s\geq\delta|b_k,a_{k+})=\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}_{\{s>\delta\}}(s^i)$. These relations hold several interesting properties. Suppose, we fulfill the probabilistic inequality with δ_0 for a subset of candidate action sequences, that is, $\hat{\mathrm{P}}^{(m)}(s\geq\delta_0|b_k,a_{k+})\geq1-\epsilon$ for $\{a^2,a^3\}$ in Fig. 4(a). We shall increase δ_0 to invalidate more candidate action sequences up until a single candidate action sequence is left. Before δ_0 is increased to δ_1, currently invalidated candidate action sequences can be discarded for eternity [$\{a^1\}$ in Fig. 4(a)], they will never fulfill the outer constraint with $\delta_2>\delta_0$ due to the never increasing step size in our approach of alternating increases and decreases of δ.

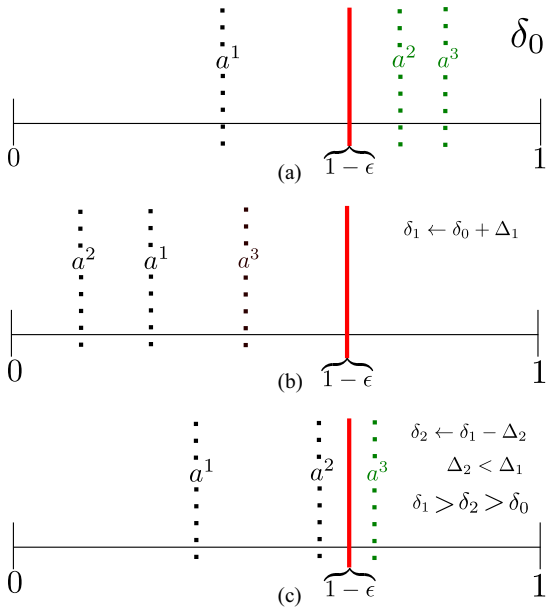


Fig. 4. Visualization of Algorithm 3. We never increase the step size. Therefore, as we see, each candidate action sequence in the bottom visualization (**c**) is shifted to the left relative to the situation displayed in the top (**a**). The action sequence a^1 can be safely discarded in the top illustration (**a**) (Section III-H). The middle visualization marked by (**b**) portray the situation when Δ_1 was too large.

Algorithm 1: Optimality Under Probabilistic Constraint (9) $\rho(\cdot) \equiv \phi(\cdot)$.

Input: $\mathcal{A}$ ▷ Set of the action sequences
$a^*_{k+} \leftarrow$ undef, $\hat{\mathcal{U}}^*_{(m)} \leftarrow -\infty$, $S \leftarrow \{\}$
for each $a_{k+} \in \mathcal{A}$ **do**
  for $\tilde{m}(a_{k+}) = 1 : m$ **do**
    Draw observation sequence $z^{\tilde{m}}_{k+1:k+L}$
    Calculate $c(b^{\tilde{m}}_{k:k+L}; \phi, \delta)$, $\sum_{t=k}^{k+L-1} \rho(b^{\tilde{m}}_t, b^{\tilde{m}}_{t+1})$
    if $1-\epsilon \leq \frac{1}{m}\sum_{l=1}^{\tilde{m}} c(b^l_{k:k+L}; \phi, \delta)$ **then**
      $S \leftarrow S \cup a_{k+}$ ▷ Accept the a_{k+}
      break ▷ check the next action seq.
    else if $\frac{1}{m}\sum_{l=1}^{\tilde{m}} c(b^l_{k:k+L}; \phi, \delta) < 1-\epsilon-\frac{m-\tilde{m}}{m}$ **then**
      break ▷ check the next action seq.
    end if
  end for
end for
for each $a_{k+} \in S$ **do** ▷ S contains all feasible a_{k+}
  expand missing laces and get $\hat{\mathcal{U}}^{(m)}(b_k, a_{k+}; \rho)$
  if $\hat{\mathcal{U}}^*_{(m)} < \hat{\mathcal{U}}^{(m)}(b_k, a_{k+}; \rho)$ **then**
    $a^*_{k+} \leftarrow a$, $\hat{\mathcal{U}}^*_{(m)} \leftarrow \hat{\mathcal{U}}^{(m)}(b_k, a_{k+}; \rho)$
  end if
end for
Return a^*_{k+}

Now, suppose all action sequences violate the probabilistic inequality with δ_1, that is, $1-\epsilon > \hat{\mathrm{P}}^{(m)}(s \geq \delta_1 | b_k, a_{k+})$ for all the candidate action sequences $[\{a^1, a^2, a^3\}$ in Fig. 4(b)]. We shall decrease the δ_1 (but in a smaller amount) to render more candidate action sequences feasible. If we will obtain δ_2, such that all the candidate action sequences besides the single one are invalidated, we know that this candidate action sequence maximizes (13). This happens in Fig. 4(b) with δ_2. Crucially, all the evaluations of the probabilities above we do using our adaptive simplification from Section III-F before actually expanding the m laces.

This is the underlying principle of Algorithm 3. See visualization in Fig. 4. As we see in Fig. 4, $\delta_2 > \delta_0$ so $\hat{\mathrm{P}}^{(m)}(s \geq \delta_0 | b_k, a_{k+}) \geq \hat{\mathrm{P}}^{(m)}(s \geq \delta_2 | b_k, a_{k+})$. To the step size, we employ the bisection principle. To rephrase it, we adaptively solve

$$
\begin{aligned}
&a^*_{k+}, \delta^* = \arg\max_{\{a_{k+}\}} \max_{\delta} \quad \delta \\
&\text{s.t. } \exists a_{k+} \in \mathcal{A} : \hat{\mathrm{P}}^{(m)}(c(b_{k:k+L}; \phi, \delta) = 1 | b_k, a_{k+}) \geq 1-\epsilon \\
&\text{s.t. } \delta^{\min} < \delta \leq \delta^{\max}(b_k) \qquad (40)
\end{aligned}
$$

actually evaluating m laces of observations only in worst case scenario. The $\delta^{\min}$ and $\delta^{\max}$ shall be supplied externally. Further, we extensively debate how to set these parameters for information gathering tasks. Crucially, in (40) we recognize why we need nonstrict inequality for δ in (13). The candidate action sequences satisfying the outer constraint with $\delta^{\max}$ must be accepted. Let us highlight that $\delta^* \triangleq \widehat{\mathrm{VaR}}^{(m)}_{\epsilon}(b_k, a^*_{k+})$, the sample approximation of (13) for the optimal action sequence a^*_{k+} in (12) utilizing (24). The formulation (40) is generalization of solving the maximal feasible return problem portrayed by (12) for two forms of inner constraints (10) and (11).

Note that depending on the scenario, it is possible that for many candidate actions, but not all, the $\widehat{\mathrm{VaR}}^{(m)}_{\epsilon}(b_k, a_{k+})$ is close to one of the edges of the bounds over δ. If it is a lower bound $\delta^{\min}$, we will be able to easily discard a candidate action a_{k+} (with appropriate ϵ regime) using Algorithm 3 as visualized in Fig. 3(b). Conversely, if it is the upper bound $\delta^{\max}$, it will be easy to accept a candidate action as in Fig. 3(d).

Before we continue, to algorithms let us emphasize the important points. In Appendix C, we discuss sample approximations used in our proposed algorithms. To remove unnecessary clutter, we formulate our algorithms for the first level bounds (26) and (27). However, given the monotonically converging to ϕ bounds as in (28), adjusting the algorithms does not pose a problem. In addition, the approach described in this section works also for solving (40) for a multiplicative form of the inner constraint (10). This, however, is outside the scope of this article, since in this article we focus on cumulative flavor (11). We are ready for the next section, where we formulate algorithms to tackle both of our formulated problems.

I. Algorithms

In this section, we present four algorithms. All the algorithms receive as input the set of candidate action sequences. For both our formulated problems, we propose our technique and describe the baseline.

Algorithm 2: Optimality of (7) Under Averaged Constraint (14) (Baseline) $\rho(\cdot) \equiv \phi(\cdot), \mathcal{U}(\cdot) \equiv \mathcal{C}(\cdot)$.

1: **procedure** PLAN
2: **Input:** $\mathcal{A}$
3: $a^*_{k+} \leftarrow$ undef, $\hat{\mathcal{U}}^*_{(m)} \leftarrow -\infty$,
4: **for** each $a_{k+} \in \mathcal{A}$ **do**
5: Expand m laces and get $\hat{\mathcal{U}}^{(m)}(b_k, a_{k+}; \rho)$
6: **if** $\hat{\mathcal{U}}^*_{(m)} < \hat{\mathcal{U}}^{(m)}(b_k, a_{k+}; \rho)$ **then**
7: $a^*_{k+} \leftarrow a_{k+}$, $\hat{\mathcal{U}}^*_{(m)} \leftarrow \hat{\mathcal{U}}^{(m)}(b_k, a_{k+}; \rho)$
8: **end if**
9: **end for**
10: **if** $\hat{\mathcal{U}}^*_{(m)} > \delta$ **then**
11: **Return** a^*_{k+}
12: **else**
13: **Return** No feasible a_{k+} ▷ $\max_{a_{k+} \in \mathcal{A}} \hat{\mathcal{U}}^{(m)}(b_k, a_{k+}; \rho) \leq \delta$
14: **end if**
15: **end procedure**

1) Optimality Under Probabilistic Constraint: For the first formulated problem (9), we adaptively check the feasibility of all the action sequences and select the optimal one from the set of feasible action sequences in Algorithm 1. If the condition in line 7 or 10 is not satisfied, it means that the Algorithm 1 will jump to the next iteration of the loop in line 4 and expand one more lace. This is in agreement with the explanation in Section II-I-F. Sooner or later, for $\tilde{m}(a_{k+}) \leq m$, one of these conditions will be met and Algorithm 1 will move to the next candidate action. The competing approach is finding the optimal action sequence and verifying feasibility afterward, see Algorithm 2. Since Algorithm 2 uses expectation for constraint as in (15) and Algorithm 1 uses our probabilistic constraint the selected best action sequence can differ for two algorithms.

2) Maximal Feasible Return: Here, we propose our adaptive method described in Section III-H and summarized in Algorithm 3 and evaluate/compare it versus the brute force maximization of $\widehat{\text{VaR}}^{(m)}_{\epsilon}$ by Algorithm 4. Importantly, Algorithm 3 is formulated for both flavors of the inner constraint, i.e., cumulative (10) and multiplicative (11). Algorithm 3 requires two parameters $\delta^{\min}$ and $\delta^{\max}$. The former, $\delta^{\min}$, is a requirement. The latter, $\delta^{\max}$, has to be supplied externally for a particular operator ϕ. In subsequent sections we extensively debate on how to do that. If no candidate action sequence a_{k+} fulfills the constraint with $\delta^{\min}$ we declare that no feasible solution exists. For exploration purposes (in SLAM and SD problems) we only care to select an optimal candidate action sequence maximizing (40) and that $\delta^* \geq \delta^{\min}$. To save valuable time we will not engage the optional hibiscus colored part of the Algorithm 3. In this case the Algorithm 3 selects a^*_{k+} as in (40), but returned $\delta^* \leq \widehat{\text{VaR}}^{(m)}_{\epsilon}(b_k, a^*_{k+})$. Note also that we need to expand a single lace in line 3 of Algorithm 3 in order to try to verify the (25) with a new value of δ before adding a lace in line 31.

Having introduced the algorithms we shall discuss possible drawbacks and overhead.

Algorithm 3: Maximal Feasible Return (Bisection method).

1: **Input:** $\mathcal{A}$, $\delta^{\min}$, $\delta^{\max}$, m
2: $S \leftarrow \mathcal{A}, T \leftarrow \mathcal{A}, \tilde{\delta}^{\min} \leftarrow \delta^{\min}, \tilde{\delta}^{\max} \leftarrow \delta^{\max}, \delta \leftarrow (\frac{\tilde{\delta}^{\min} + \tilde{\delta}^{\max}}{2})$
3: $\forall a_{k+} \in \mathcal{A}$ expand a lace and $\tilde{m}(a_{k+}) \leftarrow 1$ ▷ warm up
4: **while** true **do** ▷ cand. action seq. and laces loop
5: **for each** $a_{k+} \in S$ **do**
6: **if** !ADAPTBOUNDS(a_{k+}, $\tilde{m}(a_{k+})$, δ) **then**
7: $S \leftarrow S \setminus a_{k+}$,
8: **end if**
9: **end for**
10: **if** $|S| == 1$ **then**
11: store $a_{k+} \in S$ increase δ and adapt bounds for a_{k+} up until $S \subset \emptyset$ ▷ Hibiscus color denotes optionary part. See Section III-I2.
12: **return** a_{k+}, δ
13: **else if** $S \subset \emptyset$ **then**
14: **if** $\delta == \delta^{\min}$ **then**
15: **return** nothing ▷ No feasible solution
16: **end if**
17: $S \leftarrow T, \tilde{\delta}^{\max} \leftarrow \delta, \delta \leftarrow \frac{\tilde{\delta}^{\min} + \tilde{\delta}^{\max}}{2}$ ▷ $\delta \leftarrow \delta - \underbrace{\frac{\delta - \tilde{\delta}^{\min}}{2}}_{\Delta}$
18: **else if** $\delta == \delta^{\max}$ **then**
19: **return** some $a_k \in S, \delta$ ▷ All action seq. in S yield identical maximal possible objective
20: **else**
21: $T \leftarrow S, \tilde{\delta}^{\min} \leftarrow \delta, \delta \leftarrow \frac{\tilde{\delta}^{\min} + \tilde{\delta}^{\max}}{2}$ ▷ $\delta \leftarrow \delta + \underbrace{\frac{\tilde{\delta}^{\max} - \delta}{2}}_{\Delta}$ Some action seq. possibly were discarded for eternity.
22: **end if**
23: **end while**
24: **procedure** ADAPTBOUNDS(action seq: a_{k+}, $\tilde{m}$, δ) ▷ $\tilde{m}(a_{k+})$ is a global variable.
25: **while** true **do**
26: **if** $\frac{1}{m} \sum_{l=1}^{\tilde{m}(a_{k+})} c(b^l_{k:k+L}; \phi, \delta) < 1 - \epsilon - \frac{m - \tilde{m}}{m}$ **then**
27: **return** false
28: **else if** $1 - \epsilon \leq \frac{1}{m} \sum_{l=1}^{\tilde{m}(a_{k+})} c(b^l_{k:k+L}; \phi, \delta)$ **then**
29: **return** true
30: **end if**
31: $\tilde{m}(a_{k+}) \leftarrow \tilde{m}(a_{k+}) + 1$, Draw a lace $z^{\tilde{m}}_{k+1:k+L}$
32: Calculate $c(b^{\tilde{m}}_{k:k+L}; \phi, \delta)$
33: **end while**
34: **end procedure**

J. Adaptation Overhead

In Algorithm 3 we shall evaluate the inner constraint and sum up $\sum_{l=1}^{\tilde{m}(a_{k+})} c^l(b_{k:k+L}; \phi, \delta)$ for multiple values of δ. This necessitates to store $\sum_{t=k}^{k+L-1} \phi(b^l_t, b^l_{t+1})$, in case of (10), and $\{\phi(b^l_t)\}_{t=k}^{k+L}$, in case of (11), for every expanded l. Accordingly, the memory consumption is elevated, however, it does not require much memory, since these are one dimensional values. Nevertheless, as we believed and verified by the experiments, this overhead is neglectable compared with the time saved on skipped laces due to loop closures in SLAM or determinant calculation of a large matrix in SD, as we will further witness.

Algorithm 4: Baseline Maximizing $\widehat{\text{VaR}}_{\epsilon}^{(m)}$.

1: **Input:** $\mathcal{A}$
2: $a_{k+}^{*} \leftarrow$ undef, $\hat{\mathcal{V}}_{(m)}^{*} \leftarrow -\infty$
3: **for each** $a_{k+} \in \mathcal{A}$ **do**
4: Expand m laces and approximate $\widehat{\text{VaR}}_{\epsilon}^{(m)}$
5: **if** $\hat{\mathcal{V}}_{(m)}^{*} < \widehat{\text{VaR}}_{\epsilon}^{(m)}$ **then**
6: $a_{k+}^{*} \leftarrow a_{k+}$, $\hat{\mathcal{V}}_{(m)}^{*} \leftarrow \widehat{\text{VaR}}_{\epsilon}^{(m)}$
7: **end if**
8: **end for**
9: **Return** a_{k+}^{*}, $\hat{\mathcal{V}}_{(m)}^{*}$

Furthermore, these additional operations can be easily parallelized via multithreading.

We can, however, encounter a worst-case scenario. Imagine the ϵ is close to 1 from the left. Many action sequences will satisfy the probabilistic constraint. In general, we can say that a more accurate precision of δ will be required to differentiate between the action sequences since the working area is closer to zero and the interval $[0, 1-\epsilon)$ is shorter. Therefore, more iterations in Algorithm 3 will be required. Moreover, a pair of action sequences may be extremely close to each other in terms of $\widehat{\text{VaR}}_{\epsilon}^{(m)}$, requiring a tremendous amount of iterations of the Algorithm 3. To solve this issue, we shall introduce a final precision.

In addition, adaptation of the bounds (28) can take some toll in terms of time. This is out of the scope of this article.

K. Limitations and Drawbacks

Besides the drawbacks due to the adaptation and bookkeeping, our approach requires knowledge of the number of laces to be expanded m. We can fix that if $\epsilon = 0$ (see [30]). Further, the second layer bounds $\text{lb}^{(2)}, \text{ub}^{(2)}$ require externally supplied adaptive bounds for the operator ϕ as in (28).

IV. Application to Belief Space Planning

In this section, we apply our suggested theory to informative planning. We focus on SLAM and SD, two problems with a high-dimensional state under the umbrella of BSP. We express the exploration problem with our framework (9) as well as distributional aware high-dimensional BSP with (12).

A. Belief Structure

Let us delve into the mechanics of maintaining and updating high-dimensional belief on top of a stochastic process, sequential decision making. In this work we assume that the data association is solved. Namely, in general, the belief $\mathbb{P}(\boldsymbol{x}_k|b_0, a_{0:k-1}, \boldsymbol{z}_{1:k})$ would be (see, e.g., [35] and [36])

$$\sum_{\boldsymbol{\beta}_{1:k}} \mathbb{P}(\boldsymbol{x}_k|b_0, a_{0:k-1}, \boldsymbol{z}_{1:k}, \boldsymbol{\beta}_{1:k}) \frac{\mathrm{P}(\boldsymbol{\beta}_{1:k}|b_0, a_{0:k-1}, \boldsymbol{z}_{1:k})}{\sum_{\boldsymbol{\beta}_{1:k}} \mathrm{P}(\boldsymbol{\beta}_{1:k}|b_0, a_{0:k-1}, \boldsymbol{z}_{1:k})} \tag{41}$$

where the summation is over $\boldsymbol{\beta}_{1:k}$ appropriate to dimension of the corresponding observation $\boldsymbol{z}_{1:k}$. The dimension of observation always conveys the knowledge of number of visible landmarks resulted to such an observation in SLAM or number of sensors producing an observation in SD. For example, suppose the dimension of $\boldsymbol{z}_k$ is 2. We shall only cover $\boldsymbol{\beta}_k$ with two ones in the summation. Moreover, as we will further see the conditional PDF $\mathbb{P}(\boldsymbol{x}_k|b_0, a_{0:k-1}, \boldsymbol{x}_{1:k}, \boldsymbol{\beta}_{1:k})$ is not defined well if $\boldsymbol{z}_{1:k}$ and $\boldsymbol{\beta}_{1:k}$ disparate in terms of dimensions and number of ones reciprocally.

In this work we, however, (as done in many works) assume that the realization of the corresponding $\boldsymbol{\beta}$ is inferred exactly from the given observation (emphasized by the red color in the next equation). This simplifies the belief structure as such

$$\mathbb{P}(\boldsymbol{x}_k|b_0, a_{0:k-1}, \boldsymbol{z}_{1:k}) = \mathbb{P}(\boldsymbol{x}_k|b_0, a_{0:k-1}, \boldsymbol{z}_{1:k}, \boldsymbol{\beta}_{1:k}). \tag{42}$$

With this insight in mind we define the belief as, $b_k(\boldsymbol{x}_k) \triangleq \mathbb{P}(\boldsymbol{x}_k|b_0, a_{0:k-1}, \boldsymbol{z}_{1:k}, \boldsymbol{\beta}_{1:k})$. A standard and widely used tool to maintain a high-dimensional belief in case of (42) is a factor graph [37]. Its building blocks are the probabilistic motion and observation models. These models induce probabilistic dependencies over the state variables. The models are the factors that comprise the factor graph. Below we separately elaborate on specific aspects of belief structure for each considered problem.

1) Active SLAM: Applying Bayes rule to the belief, we get

$$b_k(\boldsymbol{x}_k) \propto b_0(x_0) \prod_{i=1}^{k} \left(\mathbb{P}_T\left(x_i \middle| x_{i-1}, a_{i-1}\right) \cdot \mathrm{P}_{\boldsymbol{\beta}}\left(\boldsymbol{\beta}_i \middle| x_i, \{\ell^j\}_{j=1}^{M(i)}\right) \prod_{\nu_i=1}^{n(\boldsymbol{\beta}_i)} \mathbb{P}_Z\left(z_i^{\nu_i} \middle| x_i, \ell^{j^{\nu_i}}\right) \right). \tag{43}$$

In this article, the stochastic motion and observation models for SLAM are described by the following dependencies involving Gaussian-distributed sources of stochasticity

$$x_{t+1} = f(x_t, a_t; w_t), \quad w_t \sim \mathcal{N}(0, W_t) \tag{44}$$

$$z_t^{\nu_t} = g(x_t, \ell^{j^{\nu_t}}; v_t), \quad v_t \sim \mathcal{N}(0, V_t) \tag{45}$$

where W_t and V_t are covariance matrices. The landmarks configuration model is as in (1) and (2). The prior belief $b_0(x_0)$ is assumed to be Gaussian. Similar to many other works [38], to model the belief as a multivariate Gaussian we omit the $\prod_{i=1}^{k} \mathrm{P}_{\boldsymbol{\beta}}(\boldsymbol{\beta}_i|x_i, \{\ell^j\}_{j=1}^{M(i)})$ terms and remain with

$$b_k(\boldsymbol{x}_k) \propto b_0(x_0) \prod_{i=1}^{k} \left(\mathbb{P}_T\left(x_i \middle| x_{i-1}, a_{i-1}\right) \prod_{\nu_i=1}^{n(\boldsymbol{\beta}_i)} \mathbb{P}_Z\left(z_i^{\nu_i} \middle| x_i, \ell^{j^{\nu_i}}\right) \right). \tag{46}$$

Equation (46) can be illustrated as a factor graph [39]. All in all, the overall belief (46) is modeled as a multivariate Gaussian and such a representation is exact for linear models since we have a quadratic function inside the exponent.

2) Sensor Deployment: In the SD problem the overall state $\boldsymbol{x}_k$ is a mix of a robot state x_k and a state of the phenomenon of interest $\boldsymbol{\xi}$. The belief (given $\boldsymbol{\beta}_{1:k}$) in this case takes the following

form:

$$b_k(\boldsymbol{x}_k) \propto \Big(\prod_{\nu_k=1}^{n(\boldsymbol{\beta}_k)} \mathbb{P}_Z(z_k^{\nu_k}|x_k,[\boldsymbol{\xi}]^{j^{\nu_k}}) \Big) \mathrm{P}_{\boldsymbol{\beta}}(\boldsymbol{\beta}_k|x_k) \mathbb{P}_Z(z_k^x|x_k) \cdot$$

$$\int_{x_{k-1}} \Bigg(\Big(\prod_{\nu_{k-1}=1}^{n(\boldsymbol{\beta}_{k-1})} \mathbb{P}_Z(z_{k-1}^{\nu_{k-1}}|x_{k-1},[\boldsymbol{\xi}]^{j^{\nu_{k-1}}}) \Big) \cdot \tag{47}$$

$$\mathrm{P}_{\boldsymbol{\beta}}(\boldsymbol{\beta}_{k-1}|x_{k-1}) \mathbb{P}_Z(z_{k-1}^x|x_{k-1}) \mathbb{P}_T(x_k|x_{k-1},a_{k-1}) \cdot$$

$$\Big(\int_{x_{k-2}} \dots \Big(\int_{x_0} b_0(\boldsymbol{\xi},x_0) \mathbb{P}_T(x_1|x_0,a_0) \mathrm{d}x_0 \Big) \dots \mathrm{d}x_{k-2} \Big) \Bigg) \mathrm{d}x_{k-1}.$$

Suppose that individual sensor observation model does not depend on the robot state. Moreover, typically there is no reason to assume that the prior of the quantity of interest $\boldsymbol{\xi}$ will be statistically dependent on the initial robot position x_0. In this case $b_0(\boldsymbol{\xi},x_0) = b_0(\boldsymbol{\xi}) b_0(x_0)$. This fact allows us to decompose also (47) as $b_k(\boldsymbol{\xi},x_k) = b_k(\boldsymbol{\xi}) b_k(x_k)$. Both beliefs $b_k(\boldsymbol{\xi})$ and $b_k(x_k)$ are given $\boldsymbol{\beta}_{1:k}$. Note that in general, if the belief is as in (41), such a decomposition does not hold. Equation (47) splits into two multiplicands $b_k(\boldsymbol{\xi})$ and $b_k(x_k)$ as follows:

$$b_k(\boldsymbol{\xi}) \propto \prod_{i=1}^{k} \Big(\prod_{\nu_i=1}^{n(\boldsymbol{\beta}_i)} \mathbb{P}_Z(z_i^{\nu_i}|[\boldsymbol{\xi}]^{j^{\nu_i}}) \Big) b_0(\boldsymbol{\xi}) \tag{48}$$

$$b_k(x_k) \propto \mathrm{P}_{\boldsymbol{\beta}}(\boldsymbol{\beta}_k|x_k) \mathbb{P}_Z(z_k^x|x_k) \cdot \tag{49}$$

$$\int_{x_{k-1}} \big(\mathrm{P}_{\boldsymbol{\beta}}(\boldsymbol{\beta}_{k-1}|x_{k-1}) \mathbb{P}_Z(z_{k-1}^x|x_{k-1}) \mathbb{P}_T(x_k|x_{k-1},a_{k-1}) \cdot$$

$$\big(\textstyle\int_{x_{k-2}} \dots \int_{x_0} b_0(x_0) \mathbb{P}_T(x_1|x_0,a_0) \mathrm{d}x_0 \dots \mathrm{d}x_{k-2} \big) \big) \mathrm{d}x_{k-1}.$$

Importantly, the decomposition of $b_k(\boldsymbol{\xi},x_k)$ into $b_k(\boldsymbol{\xi})$ and $b_k(x_k)$ and the dependence of each on different observations from independent models (6) allows us to update the belief separately for the quantity of interest $\boldsymbol{\xi}$ and robot pose x_k. In this work, the probabilistic models for SD problem adhere to

$$x_{t+1} = f(x_t,a_t;w_t) \tag{50}$$

$$z_t^{\nu_t} = g(\xi^{j^{\nu_t}};v_t), \quad v_t \sim \mathcal{N}(0,V_t) \tag{51}$$

$$z_t^x = x_t. \tag{52}$$

The noise of observation model (51) remains Gaussian as in SLAM problem. If, in addition, $b_0(\boldsymbol{\xi})$ is a Gaussian, this enables us to use standard well-researched solvers [38] to maintain the belief displayed by (48).

Further, for clarity of the explanation and in order to focus on the uncertainty of the quantity of the interest $\boldsymbol{\xi}$, we will assume that the robot state is discrete $x_t \in \mathbb{N}^2$. In due course, the noise w_t in motion model (50) is also discrete. We will describe it in depth in simulations section. In addition, for simplicity we assume that the robot state is fully observable (52). This is not an inherent limitation but only the choice to simplify simulations. Another representation of (52) is $\mathbb{P}_Z(z^x|x) \triangleq \delta(z^x - x)$. The sensors configuration model is as in (4) and (5). The initial robot position is also known, namely, $b_0(x_0) = \delta(x_0^{\mathrm{gt}} - x_0)$. This fact, alongside the deterministic model for $\boldsymbol{\beta}$ (4) significantly simplifies (49). Specifically, we have that $b_k(x_k) = \delta(z_k^x - x_k)$. We model the prior belief for quantity of interest $b_0(\boldsymbol{\xi})$ as Gaussian. This fact and the Gaussian noise in (51) yield that (48) has another representation as a Gaussian since after linearization inside the exponent we have a quadratic function [and this representation is exact with linear $g(\cdot)$ in (51)]. We will need this fact in the following section.

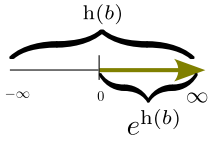


Fig. 5. New Information measure.

B. Information Measures

The forming point of informative planning is an information measure. We first delve into well-known such measures for Gaussian beliefs and, then, define our novel information measure for general beliefs.

1) Gaussian Beliefs: One possibility to define such a measure is to utilize trace of the covariance matrix of the marginal belief over the variables of interest. In such a case, commonly the information is defined (known as minus T-criterion [9]) as minus arithmetic mean of appropriate eigenvalues

$$\mathrm{I}(b) = -\frac{1}{d} \sum_{i=1}^{d} \lambda^i(b) \tag{53}$$

where d is the dimension of corresponding subset of the variables of interest. Another possibility is to utilize differential entropy $\mathrm{h}(b)$ given by (17). Differential entropy (17) was widely researched by robotics community [40] in the context of multivariate Gaussian beliefs and led to the formulation of the D-optimality criterion being the geometric mean of relevant eigenvalues of the covariance matrix of the belief (the volume of d-dimensional parallelepiped proportional to the volume of a hyperellipse manifested by the covariance matrix). The information becomes

$$\mathrm{I}(b) = -\sqrt[d]{\prod_{i}^{d} \lambda^i(b)} \tag{54}$$

where d is the dimension of the subset of the variables selected from the Gaussian belief. Observe that when Information is defined as in (53) or (54) it holds that $\mathrm{I}(b) \leq 0$ due to nonnegativity of eigenvalues of covariance matrices. Whereas differential entropy (17) is unbounded. As we will further see to define $\delta^{\max}$ for Algorithm 3 we will need that Information is bounded from above. Motivated by this requirement we define a novel Information measure for general beliefs.

2) General Beliefs: For general beliefs one possibility that is common in AI community [41], [42] is to define the Information as $\mathrm{I}(b) = -\mathrm{h}(b)$. Let us restate that multivariate Gaussian beliefs are not genuine limitation of our approach. The true requirement is upper bound on the Information measure. We can easily generalize for differential entropies on top of general beliefs by defining the Information measure as $\mathrm{I}(b) = -e^{h(b)}$. This way we again obtain $\mathrm{I}(b) \leq 0$. Observe a visualization in Fig. 5. Further, we assume that $\mathrm{I}(b) \leq 0$.

C. Information Gain

Having defined above the Information we are ready to define IG. Similar to [9], we define the operator ϕ as follows:

$$\phi(b, b') \triangleq \mathrm{IG}(b, b'). \tag{55}$$

There are various ways to define the IG over a pair of the successive beliefs. One option is

$$\mathrm{IG}(b, b') \triangleq \underbrace{\mathrm{I}'(b')}_{\leq 0} - \mathrm{I}(b) \leq -\mathrm{I}(b). \tag{56}$$

Another possibility is to define relative IG as such

$$\mathrm{IG}(b, b') \triangleq \frac{\mathrm{I}'(b') - \mathrm{I}(b)}{-\mathrm{I}(b)} \leq 1. \tag{57}$$

Let us elaborate on subsets of variables of interest for the calculation of (55). In a SLAM problem, since our focus is on the uncertainty of the environment surrounding the robot, we select *all the landmarks* as such a subset alongside the *current robot pose*, $\{x_t, \{\ell^j\}_{j=1}^{M(k)}\}$. Since we do not add landmarks in the planning session, the same dimensionality is preserved. With Gaussian beliefs and (53) and (54) this is not necessary, however. In the SD problem, we should take the belief over robot pose and the quantity of interest $\{x_t, \boldsymbol{\xi}\}$ (complete state). However, since we assumed perfect observability for the x_t, we take $\{\boldsymbol{\xi}\}$.

D. Deciding δ, δ^{min}, δ^{max}, and ϵ

In this section, we elucidate the sense of parameters of our approach separately for two of our problem formulations (9) and (12). We start from *optimality under a probabilistic constraint* (9).

1) Optimality Under a Probabilistic Constraint (Information Gathering Tasks): This problem formulation requires that the values of δ and ϵ are externally supplied. The ϵ, for example, can be close to one from the left. In this regime the practitioner enforces fulfilling the inner constraint with very high probability. Another case is ϵ very close to zero from the right. In this regime if there is a small chance of fulfilling the inner constraint, the robot will take it. For instance, if there is a small chance of decreasing uncertainty the robot will explore and will not stop. We now turn to an in-depth explanation of a meaningful δ in Information gathering tasks. For both problems under consideration, SLAM and SD, the one meaningful inner threshold is $\delta{=}0$ since it is not profitable to continue exploration or deploy the robot to operate online at all if it actually loses Information (with probability of at least $1{-}\epsilon$). Then, the robot has already deployed the candidate actions, with probability of at least $1{-}\epsilon$, leading to negative cumulative IG are redundant. Using our formulation (9) and (10) with $\delta = 0$ the robot can recognize to stop to explore the terrain (SLAM problem) or stop to deploy and make the readings from the sensors (SD problem). Recall the importance of the strict inequality in (10). The cumulative IG (55) can be nonpositive due to following reason. When the robot is active, at each time step, it increases the uncertainty due to a stochastic robot motion and decreases it by obtaining an observation. Note, however, that perfect robot observability in the SD problem makes (55) always positive. It will be clearly seen from the belief update discussed in Section V-C. If we use (57) we can set δ to be the desired fraction of the initial Information.

2) Maximal Feasible Return: The problem formulation (12) requires only manually set ϵ. Here, the value $1{-}\epsilon$ is a confidence level of VaR for each candidate action sequence a_{k+}. In other words, the fraction of sampled laces that yield return larger than VaR shall be at least $1{-}\epsilon$. To employ Algorithm 3 we require to supply minimal ($\delta^{\min}$) and maximal ($\delta^{\max}$) threshold. Let us unveil how we do that for the cumulative flavor of the inner constraint (10) and the formulation of the problem of *maximal feasible return* (12). In light of the previous discussion, we set $\delta^{\min} = 0$. Further, assume for the moment a myopic setting ($L = 1$). If (55) is in accord with (56), we elicit that the maximal feasible δ is $\delta^{\max}(b) \triangleq -\mathrm{I}(b)$. This means the uncertainty has been reduced to zero in the resulting belief. To rephrase that, the maximal Information has been reached. In this case robot can cease to operate. Whenever (55) is in accord with (57), $\delta^{\max} \triangleq 1$.

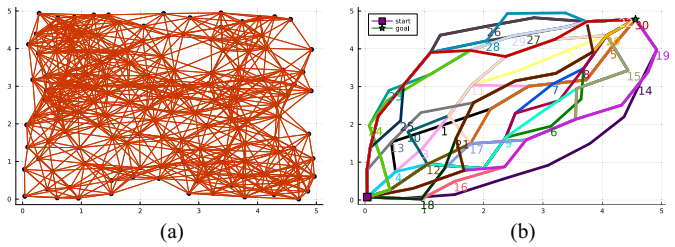
In practice our approach (Algorithm 3) requires $\delta^{\min}$ and $\delta^{\max}$ for the whole return $s(b_{k:k+L}; \cdot)$ for any L. With our definition (56) this is not a problem since we obtain telescopic series. If one uses (57) or deals with infinite horizons approximated by L steps ahead, where $\mathrm{IG}(b, b') = \gamma \mathrm{I}(b') - \mathrm{I}(b)$ [41], [42], $\delta^{\max}$ has to be adjusted accordingly. Alternatively, we can define relative IG for the terminal belief

$$\mathrm{IG}(b_k, a_{k+}, \boldsymbol{z}_{(k+1)+}, b_{k+L}) \triangleq \frac{\mathrm{I}(b_{k+L}) - \mathrm{I}(b_k)}{-\mathrm{I}(b_k)} \leq 1 = \delta^{\max}. \tag{58}$$

Having untangled these aspects, we are keen to demonstrate the superiority of the proposed approach in the following section.

V. Simulations and Results

The previous discussion leads us to the actual implementation and simulations of the proposed in Section III-I methods. It shall be noted that in this article we simulate only the first layer probabilistic constraint bounds ($\mathrm{lb}^{(1)}, \mathrm{ub}^{(1)}$). Moreover, we address in simulations only the cumulative form of the inner constraint (10). To demonstrate the advantages of the approach, we applied it on two incarnations of BSP. The first problem, we tackle, is the active SLAM while navigating in *unknown environments* to the goal. The simulation of this problem involves a *highly realistic* SLAM scenario using the GTSAM library [43]. On top of GTSAM wrapped for Python we use Julia language. Our second problem under consideration is SD. We implemented the simulations for SD purely in Julia language. In both problems under consideration the belief is multivariate Gaussian and the Information conforms to (54). Importantly, in our approach (Algorithms 1 and 3) and the baselines (Algorithms 2 and 4), we use an identical sampling method (see Appendix C). We also use the same seed per candidate action sequence in the comparisons with the baselines. This is needed to simulate identical sampling operations in baselines versus our methods according to our theory presented in Sections III-E and III-F. Before we proceed to simulations and results, let us present our measures of acceleration.



(a) (b)

Fig. 6. SLAM problem. (b) Separate, algorithmically selected paths to the goal on top of (a) PRM. We show the path number on the vertex, which is removed for finding the subsequent diverse paths. The last's path number is shown at its final vertex (the goal). Paths start from the vertex closest to the mean value of the belief in the end of the preliminary mapping session. (a) PRM. (b) Obtained diverse paths.

A. Acceleration Measures

The advantage of our proposed methods is acceleration without compromising the solution quality. We calculate the speedup, that is saved time relative to baseline time, using the following equation:

$$\frac{t^{\text{baseline}} - t^{\text{our}}}{t^{\text{baseline}}}. \tag{59}$$

We also do the same calculation in terms of laces. Namely, number of skipped laces relative to the number of laces expanded by the baseline

$$\frac{n^{\text{total}} - n^{\text{expanded}}}{n^{\text{total}}}. \tag{60}$$

Note that maximal values of (59) and (60) are 1. This means that our approach skipped all the laces [$n^{\text{expanded}} = 0$ in (60)] and run in zero time [$t^{\text{our}} = 0$ in (59)]. Moreover, the toll due to adaptation and added operations (added time divided by the baseline running time) will be the difference of (60) and (59).

B. Active SLAM While Navigating to the Goal

The generation of candidate paths is not the focus of this article. Therefore, we create candidate paths following a similar procedure to [44]. First, we employ a well-studied probabilistic road MAP (PRM) method [45]. Then, on top of PRM, to obtain diverse shortest paths, we remove a single vertex from the previous path and utilize breadth-first search on the reduced PRM. The path generation requires only the boundaries of an unknown map. In such a way, we obtain $|\mathcal{A}|$ diverse paths to the goal of various lengths. These paths constitute the space of action sequences $\mathcal{A}$ (Fig. 6b). To avoid confusion, we recite that any other method for generating candidate paths would be applicable to evaluate our proposed techniques. We illustrated the described above in Fig. 6. Let us emphasize that the paths generation depends on the starting vertex of PRM. For such a vertex we select the closest in terms of ℓ_2 norm vertex to the mean value of the belief (b_k) in the beginning of the planning session.

To keep the examination clear, we do not perform replanning sessions. Instead, we have a preliminary mapping session with manually supplied to the robot action sequence of unit length motion primitives. In the preliminary session, the robot starts from b_0, detects the landmarks, incorporates them into its state, and obtains the belief b_k. This belief serves as input to the planning session. After a single planning session, the robot follows the chosen best path.

As mentioned in Section IV-A, we assume Gaussian sources of stochasticity. The robot is described by a 2-D pose (position and bearing angle), and the landmark is a 2-D point. Our motion model (44) is a standard GTSAM odometry factor with $f(x_t, a_t; w_t) = x_t \oplus a_t + w_t$ (where $\oplus$ is a pose composition operator) with $W_t = \|a\|_2 \cdot \text{diag}(0.015, 0.015, 0.015)$. Our actions are desired pose displacement, such that $a_t = \hat{x}_{t+1} \ominus x_t$, where $\hat{x}_{t+1}$ is a nominal subsequent robot pose and $\ominus$ is the difference on manifold. Note that we need to multiply the motion model covariance matrix by the action length since our actions are of variable length. The observation (45) model is the bearing range GTSAM factor with $V_t = \text{diag}(0.001, 0.001)$. The boundaries of our map are $[0, 5] \times [0, 5]$.

We utilize the popular incremental solver ISAM2 [38] to maintain the belief. Noticeably, loop closures impose a computational challenge even with such a sophisticated incremental solver. Especially, since we need to perform inference for each posterior node in the constructed belief tree. This fact makes early eliminating or accepting actions highly important for efficient robot's operation.

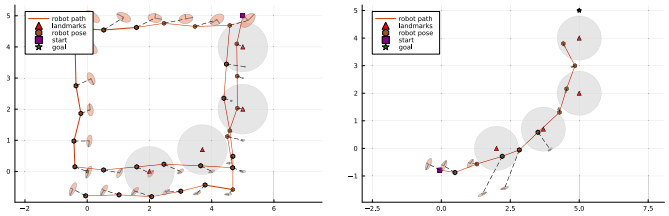
The robot constructs a belief tree of the form presented in Fig. 1(a) for each candidate path within planning session. With each promotion of the depth of the belief tree, we reduce the number of observations at each belief node by factor two, up to a possible single observation at the lowest levels. Once the maximal number of observations of the belief node is expanded, we maintain a circular slider that selects the subsequent observation with the following arrival at this belief node. The IG in SLAM problem is of the form of (56).

1) Optimality Under a Probabilistic Constraint: Following the previous discussion, we continue with the experiments. We start from our first problem (9) (optimality under a probabilistic constraint) and study Algorithm 1 versus Algorithm 2. In Algorithm 2 as opposed to Algorithm 1 we do not have a mechanism for early action dismissing until we expand all the observation laces per action sequence. In both Algorithms $\rho(\cdot) \equiv \phi(\cdot)$. We examine a scenario with four landmarks (Fig. 7). Our prior belief is Gaussain over the robot's pose $b_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ with the parameters $\mu_0 = (5.0, 5.0, 0.0)^T$, $\Sigma_0 = \text{diag}(0.001, 0.001, 0.001)$. We show the preliminary mapping session with goal at $(0.0, 0.0, 0.0)^T$ in Fig. 7(a). We elicit that, as anticipated, the uncertainty over the belief grows until the robot makes a full square and starts to experience loop closures. The path number 14 is highly likely to be optimal from an information perspective since this path lies closest to the landmarks. We employ Algorithm 1 with $m = 300$ laces per path from Fig. 6(b), $\delta = 0.0$ and various values of ϵ. We show a rigorous comparison versus Algorithm 2 with same parameters besides ϵ in Table I. Our resolution in terms of ϵ is $\Delta^{\epsilon} = 1/m$. Empirically we found that for $\epsilon \in [0, 0.023]$, without dependency on m as expected, all the paths were discarded as unfeasible (seven from 300 laces given path 14 violated the inner constraint). Meaning, no path is present with the fraction of the sampled laces

TABLE I
OPTIMALITY UNDER PROBABILISTIC CONSTRAINT

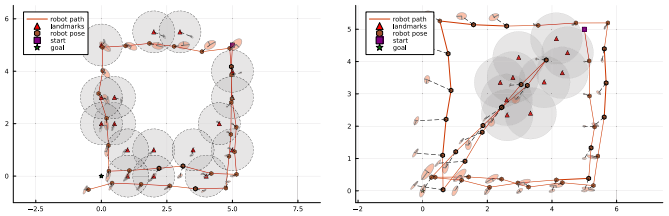
	ϵ	δ	$\mathcal{P}^*$	$\hat{\mathcal{U}}^*_{(m)}$	№discarded paths	time [s] $\pm$ std	speedup (59)	laces frac. (60)	total laces	№paths	№land.
Alg. 2	-	0.0	14	$36.98 \cdot 10^{-5}$	-	1171.21 ± 74.48		0	9000/9000	30	4
Alg. 1	0.023	0.0	no feasible	-	30	77.67 ± 4.01	**0.934**	0.95	459/9000		
Alg. 1	0.3	0.0	14	$36.98 \cdot 10^{-5}$	29	489.44 ± 26.46	**0.58**	0.60	3559/9000		
Alg. 1	0.5	0.0	14	$36.98 \cdot 10^{-5}$	29	813.29 ± 34.27	**0.31**	0.37	5685/9000		
Alg. 1	0.7	0.0	14	$36.98 \cdot 10^{-5}$	27	974.18 ± 51.14	**0.17**	0.134	7794/9000		
Alg. 1	0.8	0.0	14	$36.98 \cdot 10^{-5}$	23	1099.98 ± 45.41	**0.06**	0.029	8738/9000		
Alg. 1	0.9	0.0	14	$36.98 \cdot 10^{-5}$	0	1130.77 ± 56.47	**0.03**	0.0	9000/9000		

Here, we set m=300 observation laces per path. Each quantity was averaged over five trials with the same set of seeds for candidate action sequences. We emphasize with a bold font some of the results obtained using our approach.



(a) (b)

Fig. 7. (a) Robot's first preliminary mapping session, by transparent gray circles, we depict landmarks' visibility radius. The robot starts at the top right corner and moves toward the bottom left corner making two full squares. As we can see, the robot passed inside the visibility radius of the landmarks, detected them and incorporated them to its state. We show covariance ellipses for current robot poses. The landmarks visibility radius is 0.8. By the dashed line we connect estimated robot pose with ground truth. (b) Algorithm 2 and Algorithm 1 both selected path number 14 from Fig. 6(b) as optimal. We recognize that a pair of landmarks nearest to starting position $(5, 5)$ of preliminary mapping session in Fig. 7(a) greatly contribute to uncertainty diminishment since the robot twice made a loopclosure there.



(a) (b)

Fig. 8. (a) Robot second preliminary mapping session, by transparent gray circles encapsulated in dashed lines we depict landmarks' visibility radius. As we can see that robot detected the landmarks and incorporated to its state. The landmarks visibility radius is 0.8. We also show ellipses of the beliefs over corresponding to the time robot pose and the final landmarks uncertainty. The shaded ellipses correspond to one standard deviation. Note that if the ellipse for the landmark is not shown, this means that the robot has not seen this landmark, and such a landmark is not a part of the state. (b) Illustration of the third preliminary mapping session with randomly drawn landmarks. At each trial we draw randomly the landmarks positions.

larger than $1-0.023$ fulfilling inner constraint. For $\epsilon \leq 0.023$ our probabilistic constraint discards all candidate action sequences, but expected IG is larger than 0. This means that the expected IG is positive, whereas not all the laces yield positive IG. Our formulation is able to catch that. In Fig. 7(b), we display the robot following the identified best path. Note that with Algorithm 1, we do not accelerate decision making when we cannot discard action sequences. We shall note that due to internal GTSAM multithreading, measuring the time speedup is a challenging task. To alleviate that we repeat each run in Table I five times with identical set of seeds for candidate action sequences and report averaged running time and the speedup obtained from it. Remarkably, from the bottom line of Table I we observe that with extremely loose probabilistic constraint ($\epsilon = 0.9$) we do not eliminate any action sequence but the running time is not larger than the baseline. This fact indicates that the overhead from adaptation is so small that it was consumed by differences in running time along the trials. For more experiments with Algorithm 2, please refer to the Appendix E.

2) Maximal Feasible Return: We continue to our second problem (maximal feasible return (12)). As explained in Section IV-D, we set $\delta^{\min} = 0$ and $\delta^{\max}(b_k) = \sqrt[d]{\prod_i^d \lambda^i(b_k)}$. We set the final precision of Algorithm 3 to $\delta^{\max}(b_k) \cdot 10^{-6}$. Let us increase the number of landmarks to obtain more informative candidate paths for Information gathering. We show our second preliminary mapping session, with the same parameters as the previous one, in Fig. 8(a). Here we need many paths with nonegative IG to examine using Algorithm 3 early acceptance as well and not only early invalidation as was done in previous section. With a second preliminary mapping session [Fig. 8(a)], the starting vertex for path generation did not change. Thus, we received the candidate paths identical as in Fig. 6(b). Importantly, the paths with $\widehat{\text{VaR}}^{(m)}_{\epsilon} \leq \delta^{\min}$ are discarded for eternity (if exist at least single path with $\widehat{\text{VaR}}^{(m)}_{\epsilon} > \delta^{\min}$) with the first arrival of Algorithm 3 to line 7. So, more demanding for the Algorithm 3 simulation in terms of acceleration would be to come up with as many candidate paths with $\widehat{\text{VaR}}^{(m)}_{\epsilon} > \delta^{\min}$ as possible. Our baseline is Algorithm 4, which calculates VaR in a straightforward way. We report results in Table II, again using same set of seeds for candidate action sequences per trial. In Fig. 9(a) we visualize the execution of the optimal path and in Fig. 9(b) we display the robot trajectories sampled in planning session. Both these figures correspond to the configuration of $\epsilon = 0.3$ in Table II. In addition, note in Table II that δ^* returned with Algorithm 3 is slightly less than one returned with Algorithm 4, except when ϵ=0.5. This is an expected result as we explained in Section III-I. We did not engage customary part of Algorithm 3. The fact that when $\epsilon = 0.3$, our approach (Algorithm 3) returned larger δ^* we think is a result of the accuracy of Julia language library sample approximation of $\widehat{\text{VaR}}^{(m)}_{\epsilon}$ used in baseline method (Algorithm 4).

TABLE II
SOLVING MAXIMUM FEASIBLE RETURN PROBLEM (12) FOR SLAM ON TOP OF 30 CANDIDATE PATHS [FIG. 6(b)] WITH SCENARIO PRESENTED IN FIG. 8(A)

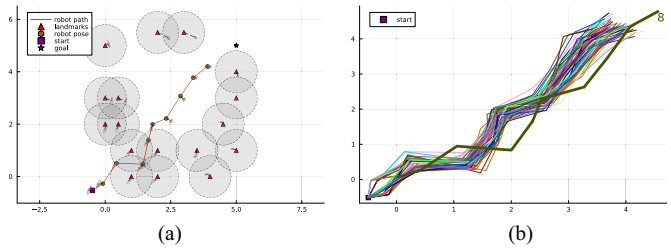
	ϵ	$\mathcal{P}^*$	δ^*	time [s] $\pm$ std	speedup (59)	laces frac. (60)	laces evaluations	№paths with $\widehat{\mathrm{VaR}}_\epsilon^{(m)} > \delta^{\min}$	№paths	№ landmarks
Alg. 4	0.3	8	$1.86 \cdot 10^{-5}$	511.03 ± 22.52	-	0	1920/1920	21	30	17
Alg. 3		8	$1.87 \cdot 10^{-5}$	$\mathbf{349.85 \pm 7.36}$	**0.32**	0.35	1257/1920			
Alg. 4	0.5	20	$2.71 \cdot 10^{-5}$	505.56 ± 23.26	-	0	1920/1920	25		
Alg. 3		20	$2.65 \cdot 10^{-5}$	$\mathbf{348.54 \pm 7.61}$	**0.31**	0.35	1245/1920			
Alg. 4	0.7	11	$2.83 \cdot 10^{-5}$	476.76 ± 12.98	-	0	1920/1920	29		
Alg. 3		11	$2.82 \cdot 10^{-5}$	$\mathbf{393.06 \pm 8.36}$	**0.18**	0.18	1565/1920			

In this study, the number of observation laces is m=64 per path. We observe that the speedup is approximately as the fraction of expanded laces, as expected since it is a little overhead from the adaptation. The values of time are averaged over ten trials with same seed. Therefore the laces evaluations in this Table per trial. The speedup is calculated from mean planning time. By the bold font we indicate our adaptive approach.

TABLE III
ANALYSIS OF THE BEHAVIOR WITH RANDOMLY DRAWN LANDMARKS

№ paths	30	30	30	30	30
min speedup	0.14	0.092	0.14	0.08	0.019
max speedup	0.57	0.43	0.32	0.22	0.081
mean/accumulated time based speedup	0.35	0.26	0.21	0.14	0.055
mean time [sec] $\pm$ std Alg. 4	817.79 ± 132.12	771.59 ± 119.82	1670.69 ± 260.39	1607.60 ± 248.46	1671.69 ± 259.52
mean time [sec] $\pm$ std Alg. 3	530.47 ± 162.30	574.55 ± 130.11	1320.71 ± 216.86	1382.89 ± 245.19	1580.00 ± 230.63
accumulated time [sec] Alg. 4	8177.92	7715.88	16706.90	16075.98	16716.86
accumulated time [sec] Alg. 3	5304.69	5745.51	13 207.11	13 828.99	15 800.04
accumulated skipped laces frac.	0.36	0.29	0.23	0.15	0.065
accumulated expanded laces Alg. 3	12 285	13 689	14 800	16 304	17 944
total № of laces	19 200	19 200	19 200	19 200	19 200
№trials	10	10	10	10	10
№ landmarks	10	10	10	10	10
ϵ	0.2	0.3	0.5	0.7	0.9

In this study, the number of laces is m=64 per candidate path. Each trial we have randomly drawn ten landmarks in the square [2, 5]×[2, 5]. Here the visibility radius of the landmarks is 0.8. Note that mean time-based speedup and the accumulated time-based speedup are identical since the difference in running time in two possibilities is only the division by the number of trials.



(a) (b)

Fig. 9. This figure corresponds to the first row of the Table II, namely, $\epsilon = 0.3$. (a) Algorithm 4 and Algorithm 3 both selected path number 8 from Fig. 6(b) as optimal. (b) Here by the thick green line we show the candidate path sequence. Note that here we show actual candidate path from Fig. 6(b). This path is converted to candidate action sequence of increments. By the thin lines of various colors we visualize the robot trajectories in planning session.

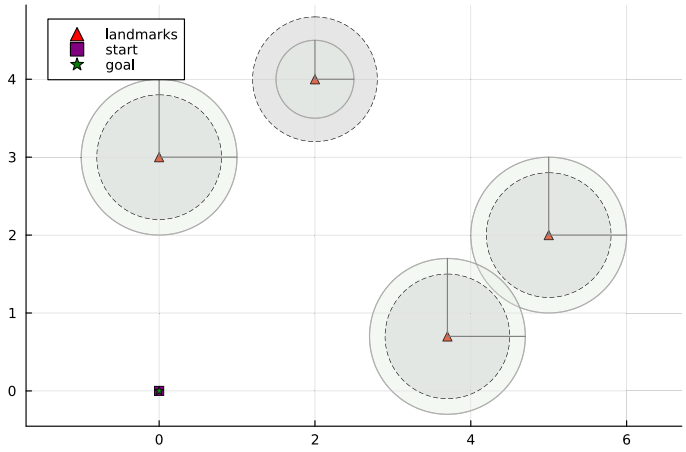


Fig. 10. Visualization of the scenario for verifying that ML observation assumption can be destructive. Robot starts to plan from b_0. Each landmark has prior shown by light green circle and the visibility radius shown by gray circle with dashed line.

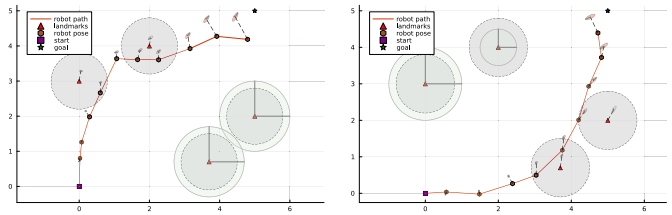
We also have an additional simulation with randomly drawing landmarks. In this simulation each trial has different set of seeds for candidate actions. For GTSAM stability purposes we add random landmarks uniformly on the square $[2, 5] \times [2, 5]$. We also slightly changed the preliminary action sequence [Fig. 8(b)]. Results are presented in Table III. As we witness from Tables II and III, we mostly obtain a significant speedup. Yet, early action elimination appears to be more prominent than early accept. The reader can find the explanation why this is happening at the end of Section III-H.

3) Maximum Likelihood Observation: Successively, we shall verify that m observation laces are needed and we indeed loose quality of decision making using a single ML observation. Note that this was already shown by [12]. Toward this end, we simulate the scenario presented in Fig. 10. The robot does

TABLE IV
SOLVING MAXIMUM FEASIBLE RETURN PROBLEM (12) FOR SENSOR DEPLOYMENT WITH SCENARIO PRESENTED IN FIG. 13

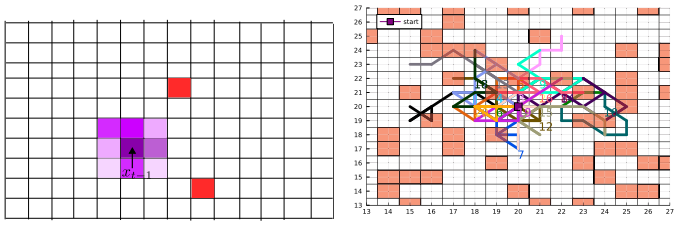
	ϵ	time [s] $\pm$ std	speedup (59)	laces frac. (60)	laces evaluations	L	m	σ^2	no sensors cells	№candidate paths
Alg. 4	0.1	1948.55 ± 259.48	-	0	75000/75000	15	150	0.1	750	20
Alg. 3		$\mathbf{1299.77 \pm 100.82}$	**0.33**	**0.39**	**45761/75000**					
Alg. 4	0.3	1146.21 ± 117.05	-	0	40000/40000	10	200	0.1	300	20
Alg. 3		$\mathbf{951.17 \pm 43.45}$	**0.17**	**0.20**	**32109/40000**					
Alg. 4	0.7	1350.74 ± 287.18	-	0	20000/20000	100	100	$1 \cdot 10^{-6}$	0	20
Alg. 3		$\mathbf{708.36 \pm 89.14}$	**0.48**	**0.57**	**8685/20000**					
Alg. 4	0.9	1199.47 ± 216.11	-	0	20000/20000	100	50	$1 \cdot 10^{-6}$	0	40
Alg. 3		$\mathbf{229.17 \pm 39.83}$	**0.81**	**0.86**	**2842/20000**					

In this study, the various number of laces per path and scalability with growing number of candidate action sequences are shown. We observe that the speedup is approximately as the fraction of expanded laces, as expected since it is a little overhead from the adaptation. The values of time are averaged over ten trials with different seeds so that we simulate a new covariance matrix of $b_k(\boldsymbol{\xi})$ each time. Therefore, we show accumulated laces evaluations over whole ten trials. The speedup is calculated from mean planning time. We emphasize our approach by the bold font.



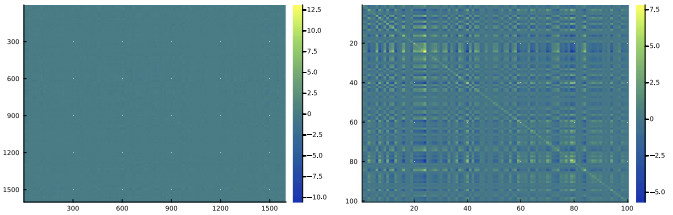
(a) (b)

Fig. 11. (a) Execution of the optimal action sequence number 30 selected by Algorithm 4 with $\epsilon = 0.5$ and $m = 728$. (b) Execution of the optimal action sequence number 14 selected by Algorithm 4 with $\epsilon = 0.5$ and ML assumption.



(a) (b)

Fig. 12. (a) Conceptual illustration of our scenario and the transition model structure for SD problem. In time index $t-1$ the robot take an action $\uparrow$ and by time t, the robot can be in one of the purple cells. The intensity designate the chance to be there. The red cells are not suitable for deploying the sensors. (b) Example of candidate paths for SD problem. By red opaque color we mark cells which are nonsuitable for deploying a sensor. However, we still desire to measure the quantity of interest in these cells using statistical dependence of the cells.



(a) (b)

Fig. 13. $\boldsymbol{\xi} \in \mathbb{R}^{1600}$ (a) Covariance of $b_0(\boldsymbol{\xi})$; (b) Zoom in.

not do any preliminary actions, but each landmark has a prior. The belief for planning b_k is prior belief b_0 with parameters $\mu_0 = (0.0, 0.0, 0.0)^T$, $\Sigma_0 = \mathrm{diag}(0.001, 0.001, 0.001)$. The starting vertex for paths generation was identical as in Fig. 6(b), so the obtained candidate paths are also as in Fig. 6(b).

We apply Algorithm 4 for planning with $\epsilon = 0.5$ and compare $m = 728$ with an ML assumption. As we recognize in Fig. 11(a) and (b), the two settings result in different optimal paths. With an ML assumption, Algorithm 4 identified the path number 14 as the best with $\widehat{\mathrm{VaR}}^{\mathrm{ml}}_{0.5} = 0.036$, whereas for path number 30 the objective was $\widehat{\mathrm{VaR}}^{\mathrm{ml}}_{0.5} = 0.032$. In contrast, using $m = 728$ observation laces, Algorithm 4 selected the path number 30 as the best, with $\widehat{\mathrm{VaR}}^{(728)}_{0.5} = 0.032$, whereas for path number 14 the objective was $\widehat{\mathrm{VaR}}^{(728)}_{0.5} = -0.014$. We witness that for path 14 the ML observation fails to adequately represent the underlying distribution.

C. Sensor Deployment

There are up to L sensors that should be scattered in a larger area. For the sake of simplicity, we discretize the area into an $n \times n$ grid. The robot takes a path of length of L cells. In each cell, it can deploy the sensor and make a reading or just make a reading if there is already a sensor there, or do nothing if the sensor can not be deployed in this cell. We still want to measure the quantity of interest in this cell leveraging statistical dependence between the cells. Using linear indices, all random variables of interest from an $n \times n$ field are combined to a random vector of size N. Our prior belief $b_0(\boldsymbol{\xi})$ has covariance $\Sigma_0 \in \mathbb{R}^{N \times N}$ with $N \triangleq n^2$. For simplicity, we assume that a single sensor at the robot sighting contributes to the observations. Meaning, $\boldsymbol{\beta}$ has single 1 in the cell of the robot's current location if there is a sensor in this cell and all the rest zeros. Our observation model is

$$\mathbb{P}_Z(z|\boldsymbol{\xi}, \boldsymbol{\beta}) \triangleq \mathcal{N}(z; \boldsymbol{\beta}^T \cdot \boldsymbol{\xi}, \sigma^2). \tag{61}$$

If no sensor is located in a cell, the $\boldsymbol{\beta}$ is all zeros, such a cell will not produce an observation and the robot will perform next action. With the observation model (61), the belief update is exact, as we describe in Appendix D. We implemented the belief update by ourselves and not used GTSAM library [43]. As we witness in (70) of Appendix D, the Information (covariance) matrix does not depend on the actual observation but only depends on the robot pose, which yielded the corresponding observation through dependence of the observation model on $\boldsymbol{\beta}$, so that the IG in this case also depends only on the robot pose. This is happening since our observation model (61) is linear and noise variance σ^2 does not depend on the state ($\boldsymbol{\xi}$).

In this problem solely for simplicity we utilize the relative IG and select (58) as a return. Our action space of motion

primitives consists of nine possible actions $\mathcal{A} = \{a_1, \ldots, a_9\}$, such that $a^1 = (1,0)^T$, $a^2 = (-1,0)^T$, $a^3 = (0,1)^T$, $a^4 = (0,-1)^T$, $a^5 = (1,1)^T$, $a^6 = (-1,1)^T$, $a^7 = (1,-1)^T$, $a^8 = (-1,-1)^T$, and $a^9 = (0,0)^T$. The agent is fully observable with the following motion model $x_{t+1} = x_t + a_t + w_t$. At the places far enough from the fringes of the map the w_t follows $\mathbb{P}(w_t) = \sum_{i=1}^{9} \mathrm{P}^i \delta(w_t - a^i)$, where P^i can be any probabilities [see Fig. 12(a)]. Close to the fringes of the map we leave only allowed actions and renormalize the above PDF accordingly. One possibility is to take a weight as a value of Gaussian with covariance matrix Σ_t, and the mean $\mu_t = 0$. We have that $\mathbb{P}(w_t) = \sum_{i=1}^{9} \frac{\mathcal{N}(a^i;0,\Sigma)}{\sum_{i=1}^{9}\mathcal{N}(a^i;0,\Sigma)} \delta(w_t - a^i)$, where by $\mathcal{N}(\square;\mu,\Sigma)$ we denote Gaussian distribution evaluated at the point $\square$. For the candidate path creation we sample uniformly actions from our action space [see an example in Fig. 12(b)]. The belief tree in this problem is as in Fig. 1(b). In Fig. 13, we show the covariance of the prior belief $b_0(\boldsymbol{\xi})$. We select $n = 40$, thereby our grid is of the dimension 40×40, resulting in $\boldsymbol{\xi} \in \mathbb{R}^{1600}$. We present results for the maximal feasible return problem (12).

1) Maximal Feasible Return: We set $\delta^{\min} = 0$, $\delta^{\max} = 1$ and compare Algorithm 3 versus Algorithm 4. With perfect robot observabiltiy in SD the uncertainty can only decrease as we observe from the belief update (70). Therefore, the IG is always nonnegative. We present results in Table IV. We observe substantial speedup in all configurations. The best speedup of 0.81 was obtained with $\epsilon = 0.9$ since many candidate paths yielded $\widehat{\mathrm{VaR}}_{0.9}^{(100)} = 1.0$ due to very low noise in observation model. In baseline approach Algorithm 4 it is impossible to catch such a situation. Note that since we simulate a new covariance matrix each trial, we obtain a different best path and δ^*. We do not show these values in Table IV, however, as in SLAM, typically δ^* returned by Algorithm 3 is slightly smaller than the one returned by Algorithm 4. This is a direct result of not engaging customary part of our approach (Line 11 in Algorithm 3) as explained in Section III-I-2.

D. Technical Details

We used four computers with the following characteristics:

1) 8 cores Intel(R) Xeon(R) CPU E5-1620 v4 working at 3.50 GHz with 80 GB of RAM;
2) 8 cores Intel(R) Xeon(R) CPU E5-1620 v4 working at 3.50 GHz with 64 GB of RAM;
3) 16 cores 11th Gen Intel(R) Core(TM) i9-11900 K working 3.50 GHz with 64 GB of RAM; and
4) 32 hardware threads AMD Ryzen 9 7945HX with 32 GB of RAM.

VI. Conclusion

We presented a novel adaptive technique for two problems, BSP with probabilistic belief-dependent constraints and BSP with VaR as an objective. Both problems are relevant in the context of Information gathering tasks. On top of that, we provably extended the simplification paradigm of decision making problems to our setting. Our rigorous theory is summarized by two novel adaptive algorithms, solving optimality under a probabilistic constraint problem and the maximal feasible return problem where we adaptively maximize VaR. Our algorithms are guaranteed to return an identical-quality solution in a fraction of the baseline running time. In addition, our framework provides a mechanism for stopping exploration, which would happen either when all candidate action sequences do not satisfy the constraint (25) in Algorithm 1, or, in the second considered problem (Algorithm 3), when the upper bound of a maximum feasible return is achieved ($\delta^{\max}$). Extensive simulations show the superiority of our methods. In the exceptionally challenging problems of active SLAM and SD, both with a high-dimensional state, we obtained a typical speedup of 30%. In the SD problem we obtained maximal speedup of 81% when the noise of observation model is very small. Our acceleration is entirely harmless regarding the quality of the decision making. The same action is always calculated as the corresponding, not accelerated, approach. Future work includes applying our approach to finding a maximal feasible multiplicative inner constraint.

Appendix A
Theoretical Observation Likelihood

To express the observation in terms of probabilistic models available to our disposal we marginalize over the $\boldsymbol{x}_{t+1}$

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{z}_{t+1}|b_t, a_t, \boldsymbol{\beta}_{t+1})\mathrm{P}(\boldsymbol{\beta}_{t+1}|b_t, a_t) \\
&\quad = \int_{\boldsymbol{x}_{t+1}} \mathbb{P}(\boldsymbol{z}_{t+1}|b_t, a_t, \boldsymbol{\beta}_{t+1}, \boldsymbol{x}_{t+1}) \cdot \qquad (62) \\
&\mathbb{P}(\boldsymbol{x}_{t+1}|b_t, a_t, \boldsymbol{\beta}_{t+1})\mathrm{P}(\boldsymbol{\beta}_{t+1}|b_t, a_t)\mathrm{d}\boldsymbol{x}_{t+1} \\
&\quad = \int_{\boldsymbol{x}_{t+1}} \mathbb{P}(\boldsymbol{z}_{t+1}|b_t, a_t, \boldsymbol{\beta}_{t+1}, \boldsymbol{x}_{t+1}) \cdot \\
&\mathbb{P}(\boldsymbol{x}_{t+1}|b_t, a_t)\mathrm{P}_{\boldsymbol{\beta}}(\boldsymbol{\beta}_{t+1}|\boldsymbol{x}_{t+1})\mathrm{d}\boldsymbol{x}_{t+1}. \qquad (63)
\end{aligned}$$

All quantities in (63) are available. Such a representation enables us to draw the observations in look-ahead step $t+1$.

Appendix B
Proof of Theorem 1 (Simplification Machinery)

We first provide the proof for the strict inequality in (10) and then explain changes that need to be done for the nonstrict inequality (10) to support our adaptive approach for problem (12) as stated after (40). It is sufficient to show that the following holds for every sample $z^l_{k+1:k+L}$:

$$\underline{c}(b^l_{k:k+L}; \underline{\phi}, \delta) \le c(b^l_{k:k+L}; \phi, \delta) \le \overline{c}(b^l_{k:k+L}; \overline{\phi}, \delta). \qquad (64)$$

Let us start from the left inequality of (64). We shall prove that $\underline{c}(b^l_{k:k+L}; \underline{\phi}, \delta) - c(b^l_{k:k+L}; \phi, \delta) \le 0$. Assume in contradiction that $\exists b^l_{k:k+L}, \underline{\phi}(\cdot), \phi(\cdot), \delta$, such that

$$\underline{c}(b^l_{k:k+L}; \underline{\phi}, \delta) - c(b^l_{k:k+L}; \phi, \delta) > 0. \qquad (65)$$

The fact that $c, \underline{c} \in \{0,1\}$ implies that this is equivalent to $\underline{c}(b^l_{k:k+L}; \underline{\phi}, \delta) = 1$ and $c(b^l_{k:k+L}; \phi, \delta) = 0$. For the inner constraint of the form (10), this can happen if and only if $(\sum_{t=k}^{k+L-1} \underline{\phi}(b^l_{t+1}, b^l_t)) > \delta$ and $(\sum_{t=k}^{k+L-1} \phi(b^l_{t+1}, b^l_t)) \le \delta$. We

behold a contradiction to the LHS part of (28), namely, the contradiction to the fact that $\underline{\phi}(\cdot) \leq \phi(\cdot)$.

Subsequently, for the multiplicative flavor (11), inequality (65) is equivalent to the existence of t, such that $\phi(b_t^l) < \delta$ (to render $c = 0$). In the same time $\forall t$ it must hold that $\underline{\phi}(b_t^l) \geq \delta$ (to render $\underline{c} = 1$) producing again a contradiction to the LHS part of (28).

To prove the right inequality of (64), we shall prove that $c(b_{k:k+L}^l; \phi, \delta) - \overline{c}(b_{k:k+L}^l; \overline{\phi}, \delta) \leq 0$. We can bear out the desired result by switching the roles of $\underline{c}(b_{k:k+L}^l; \underline{\phi}, \delta)$ to $c(b_{k:k+L}^l; \underline{\phi}, \delta)$ and $c(b_{k:k+L}^l; \phi, \delta)$ to $\overline{c}(b_{k:k+L}^l; \overline{\phi}, \delta)$ in (65) and arguing in a similar manner using $\overline{\phi}$ and the RHS part of (28). To fix the proof for the nonstrict inequality as in (13), one needs to change the inequalities marked by the red color from strict to nonstrict and vice versa. This concludes the proof. Note that we also land at an identical result (convergence almost surely) for theoretical counterparts of following probabilities and not sample approximations by taking the limits:

$$\lim_{m\to\infty} \frac{1}{m} \sum_{l=1}^{m} \underline{c}(b_{k:k+L}^l; \underline{\phi}, \delta) \leq \lim_{m\to\infty} \frac{1}{m} \sum_{l=1}^{m} c(b_{k:k+L}^l; \phi, \delta) \tag{66}$$

$$\lim_{m\to\infty} \frac{1}{m} \sum_{l=1}^{m} c(b_{k:k+L}^l; \phi, \delta) \leq \lim_{m\to\infty} \frac{1}{m} \sum_{l=1}^{m} \overline{c}(b_{k:k+L}^l; \overline{\phi}, \delta). \tag{67}$$

■

Appendix C
Sample Approximations

The core of our sample approximations is sequential sampling the observations from $\mathbb{P}(\boldsymbol{z}_{t+1}|b_t, a_t, \boldsymbol{\beta}_{t+1})$ using previously sampled $\boldsymbol{\beta}_{t+1} \sim \mathrm{P}(\boldsymbol{\beta}_{t+1}|b_t, a_t)$. Following the theoretical derivation presented in Appendix A, we leverage the structure verified by (63) in the following way.

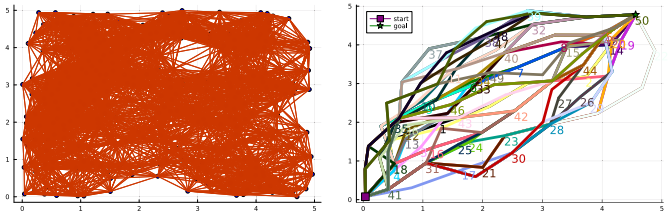
1) SLAM: First, we sample the last pose and the landmarks from the corresponding marginal of the belief. Our belief is Gaussian, thus, we just pull the appropriate portion of the covariance matrix and the mean value followed by sampling from $(x_{t+1}, \{\ell^j\}_{j=1}^{M(k)})^o \sim \mathbb{P}(x_{t+1}, \{\ell^j\}_{j=1}^{M(k)}|b_t, a_t)$. Afterward, we sample $\boldsymbol{\beta}_{t+1}$ using (2) and draw samples of the observation lace from the observation model (3).

2) Sensor Deployment: In SD problem we have that

$$\begin{aligned}\mathbb{P}(x_{t+1}|b_t, a_t) &= \int_{x_t} \mathbb{P}(x_{t+1}|x_t, b_t, a_t)\mathbb{P}(x_t|b_t)\mathrm{d}x_t \\ &= \int_{x_t} \mathbb{P}_T(x_{t+1}|x_t, a_t)\delta(x_t - z_t^x)\mathrm{d}x_t = \mathbb{P}_T(x_{t+1}|z_t^x, a_t).\end{aligned} \tag{68}$$

Having sampled the state from (68), we can sample $\boldsymbol{\beta}_{t+1}$ from (5) and the observation from (6).

Finally, the sample approximation of $\mathcal{U}$ and $\mathcal{C}$ are denoted by $\hat{\mathcal{U}}^{(m)}$ and $\hat{\mathcal{C}}^{(m)}$, respectively, and calculated by sample means of $\{s(b_{k:k+L}^l)\}_{l=1}^m$; $\widehat{\mathrm{VaR}}_\epsilon^{(m)}$ is obtained by sample quantile.



(a) (b)

Fig. 14. SLAM problem. (b) Algorithmically selected paths to the goal on top of denser (a) PRM. We show the path number on the vertex, which is removed for finding the subsequent diverse path. The last's path number is shown at its final vertex (the goal). (a) PRM (b) Obtained diverse paths.

Appendix D
Sensor Deployment Belief Update

For completeness of this article, in this section, we develop an exact belief update for SD problem with observation model as in (61), namely, $\mathbb{P}_Z(z|\boldsymbol{\beta}\cdot\boldsymbol{\xi}) = \frac{\exp(-\frac{1}{2}\|\sigma^{-1}(\boldsymbol{\beta}^T\cdot\boldsymbol{\xi}-z)\|_2^2)}{\sigma\sqrt{(2\pi)}}$, where vector $\boldsymbol{\beta}$ has one at the linear index of coordinate of the cell that resulted in this observation. Now, we need to update the belief with an action a and the observation z. Without loosing generality, suppose we have Gaussian $b_{k-1}(\boldsymbol{\xi}_{k-1})$ with mean μ_{k-1} and covariance Σ_{k-1}. Our goal is to update it with observation. We have that $b_k(\boldsymbol{\xi}_k) \propto \mathbb{P}_Z(z|\boldsymbol{\beta}^T\cdot\boldsymbol{\xi})b_{k-1}(\boldsymbol{\xi}_{k-1})$. As we explained in Section IV-A-2, the above expression will be an another Gaussian with mean $\boldsymbol{\xi}^*$, which is a unique solution to $\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \|\sigma^{-1}(\boldsymbol{\beta}^T\boldsymbol{\xi} - z)\|_2^2 + \|\Sigma_{k-1}^{-1/2}(\boldsymbol{\xi} - \mu_{k-1})\|_2^2$. Rearranging the terms, the previous equation becomes

$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \|\breve{A}\boldsymbol{\xi} - \breve{b}\|_2^2 \tag{69}$$

where $\breve{A} = \begin{pmatrix}\sigma^{-1}\boldsymbol{\beta}^T \\ \Sigma_{k-1}^{-1/2}\end{pmatrix}$, $\breve{b} = \begin{pmatrix}\sigma^{-1}z \\ \Sigma_{k-1}^{-1/2}\mu_{k-1}\end{pmatrix}$ and $\breve{A}$ has a full column rank with number of rows larger than number of columns. Solving the least squares problem (69), we have that $\boldsymbol{\xi}^* = (\breve{A}^T\breve{A})^{-1}\breve{A}\breve{b}$ and

$$\Lambda_k = \breve{A}^T\breve{A} = \Lambda_{k-1} + \boldsymbol{\beta}\sigma^{-2}\boldsymbol{\beta}^T \tag{70}$$

where $\Lambda_k = \Sigma_k^{-1}$ is the unique Information matrix of the desired Gaussian. From (70), we see that at each time, we increase the diagonal value of Λ_{k-1} corresponding to the active sensor.

Appendix E
Additional Simulations

In this section, we show additional simulations of Algorithm 2 applied to the problem of active SLAM. The preliminary robot mapping section is as in Fig. 7(a).

We first experiment with Algorithm 2 on top of the PRM as in Fig. 6(a) and paths from Fig. 6(b). From Table V, we infer that, indeed, the sensitivity to the number of samples is low. Using only ten observation laces, Algorithm 2 identified path 14 as optimal. Note that we can not recognize such a behavior before planning with $m = 200$ observation laces. The reason for such good decision making using a tiny amount of the samples of the observation episodes is that the best candidate path is far in terms of the objective from other paths. To verify this claim, we

TABLE V
IN THIS SIMULATION THE $\delta = 0$ AND NUMBER OF CANDIDATE PATHS IS 30

$\mathcal{P}^*$	$\hat{U}^*_{(m)}$	m
24	$6.55e-04$	5
14	$4.30e-04$	10
14	$4.23e-04$	50
14	$4.16e-04$	100
14	$3.88e-04$	200

The set of seeds is identical to the comparison in Table I.

TABLE VI
IN THIS SIMULATION THE $\delta = 0$ AND NUMBER OF CANDIDATE PATHS IS 50

$\mathcal{P}^*$	$\hat{U}^*_{(m)}$	m
41	$4.92e-04$	5
41	$3.17e-04$	10
41	$9.12e-05$	50
27	$7.12e-05$	100
21	$4.10e-05$	200

The set of seeds for the first 30 paths is identical to the comparison in Table I.

make PRM denser, as shown in Fig 14(a), and find 50 candidate diverse paths (Fig. 14(b)). We present results in Table VI. As we see in Table VI, increasing the number of sampled laces m changes the selected optimal path.

REFERENCES

[1] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Int. J. Robot. Res.*, vol. 33, pp. 1271–1287, 2014.

[2] J. A. Placed et al., "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1686–1705, Jun. 2023.

[3] D. Kopitkov and V. Indelman, "No belief propagation required: Belief space planning in high-dimensional state spaces via factor graphs, matrix determinant lemma and re-use of calculation," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1088–1130, Aug. 2017.

[4] J. V. D. Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *Int. J. Robot. Res.*, vol. 31, no. 11, pp. 1263–1278, 2012.

[5] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 849–882, 2015.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: The MIT press, 2005.

[7] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Pérez, "Belief space planning assuming maximum likelihood observations," in *Proc. Robot.: Sci. Syst.*, 2010, pp. 587–593.

[8] R. Valencia, J. A.-Cetto, R. Valencia, and J. A.-Cetto, "Active Pose SLAM," *Mapping, Plan. Exploration Pose SLAM*, vol. 119, pp. 89–108, 2018.

[9] J. A. Placed and J. A. Castellanos, "Enough is enough: Towards autonomous uncertainty-driven stopping criteria," *IFAC-PapersOnLine*, vol. 55, no. 14, pp. 126–132, 2022.

[10] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using RAO-blackwellized particle filters," in *Proc. Robot.: Sci. Syst.*, 2005, pp. 65–72.

[11] E. Farhi and V. Indelman, "IX-BSP: Incremental belief space planning," 2021, *arXiv:2102.09539*.

[12] E. I. Farhi and V. Indelman, "IX-BSP: Belief space planning through incremental expectation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 7180–7186.

[13] N. Roy, G. J. Gordon, and S. Thrun, "Finding approximate POMDP solutions through belief compression," *J. Artif. Intell. Res.*, vol. 23, pp. 1–40, 2005.

[14] M. Araya-López, O. Buffet, V. Thomas, and F. o. Charpillet, "A POMDP extension with belief-dependent rewards," in *Proc. Neural Inf. Process. Syst.*, 2010, pp. 64–72.

[15] M. Fehr, O. Buffet, V. Thomas, and J. Dibangoye, "RHO-POMDPS have Lipschitz-continuous epsilon-optimal value functions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6933–6943.

[16] L. Dressel and M. J. Kochenderfer, "Efficient decision-theoretic target localization," in *Proc. 27th Int. Conf. Automated Plan. Scheduling*, 2017, pp. 70–78.

[17] Z. Sunberg and M. Kochenderfer, "Online algorithms for POMDPs with continuous state, action, and observation spaces," in *Proc. Int. Conf. Automated Plan. Scheduling*, 2018, pp. 259–263.

[18] A. Zhitnikov, O. Sztyglic, and V. Indelman, "No compromise in solution quality: Speeding up belief-dependent continuous POMDPs via adaptive multilevel simplification," 2023, *arXiv:2310.10274*.

[19] A. Zhitnikov and V. Indelman, "Simplified risk aware decision making with belief dependent rewards in partially observable domains," *Artif. Intell.*, vol. 312, 2022, Art. no. 103775.

[20] O. Sztyglic and V. Indelman, "Speeding up online POMDP planning via simplification," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 7174–7181.

[21] K. Elimelech and V. Indelman, "Simplified decision making in the belief space using belief sparsification," *Int. J. Robot. Res.*, vol. 41, no. 5, pp. 470–496, 2022.

[22] V. Indelman, "No correlations involved: Decision making under uncertainty in a conservative sparse information space," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 407–414, Jan. 2016.

[23] A. Kitanov and V. Indelman, "Topological multi-robot belief space planning in unknown environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 5726–5732.

[24] A. Kitanov and V. Indelman, "Topological information-theoretic belief space planning with optimality guarantees," 2019, *arXiv:1903.00927*.

[25] M. Shienman and V. Indelman, "D2A-BSP: Distilled data association belief space planning with performance guarantees under budget constraints," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 11 058–11 065.

[26] M. Shienman and V. Indelman, "Nonmyopic distilled data association belief space planning under budget constraints," in *Proc. Int. Symp. Robot. Res.*, 2022, pp. 102–118.

[27] M. Barenboim, I. Lev-Yehudi, and V. Indelman, "Data association aware POMDP planning with hypothesis pruning performance guarantees," *IEEE Robot. Autom. Lett.*, vol. 8, no. 10, pp. 6827–6834, Oct. 2023.

[28] M. Barenboim and V. Indelman, "Adaptive information belief space planning," in *Proc. 31st Int. Joint Conf. Artif. Intell. 25th Eur. Conf. Artif. Intell.*, 2022.

[29] P. Santana, S. Thiébaux, and B. Williams, "RAO*: An algorithm for chance-constrained POMDP's," in *Proc. AAAI Conf. Artif. Intell.*, 2016.

[30] A. Zhitnikov and V. Indelman, "Risk aware adaptive belief-dependent probabilistically constrained continuous POMDP planning," 2022, *arXiv:2209.02679*.

[31] C. Cadena et al., "Simultaneous localization and mapping: Present, future, and the robust-perception age," *Comput. Sci.*, 2016.

[32] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, 2008.

[33] G. C. Pflug and A. Pichler, "Time-consistent decisions and temporal decomposition of coherent risk functionals," *Math. Operations Res.*, vol. 41, no. 2, pp. 682–699, 2016.

[34] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A random-finite-set approach to Bayesian SLAM," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 268–282, Apr. 2011.

[35] S. Pathak, A. Thomas, and V. Indelman, "A unified framework for data association aware robust belief space planning and perception," *Int. J. Robot. Res.*, vol. 32, no. 2/3, pp. 287–315, 2018.

[36] V. Tchuiev, Y. Feldman, and V. Indelman, "Data association aware semantic mapping and localization via a viewpoint-dependent classifier model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 7742–7749.

[37] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: The MIT Press, 2009.

[38] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 217–236, Feb. 2012.

[39] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Found. Trends Robot.*, vol. 6, no. 1/2, pp. 1–139, 2017.
[40] H. Carrillo, I. Reid, and J. Castellanos, "On the comparison of uncertainty criteria for active SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 2080–2087.
[41] J. Fischer and O. S. Tas, "Information particle filter tree: An online algorithm for POMDPs with belief-based rewards on continuous domains," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3177–3187.
[42] A. Eck, L.-K. Soh, S. Devlin, and D. Kudenko, "Potential-based reward shaping for finite horizon online POMDP planning," *Auton. Agents Multi-Agent Syst.*, vol. 30, pp. 403–445, 2016.
[43] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology, Atlanta, GA, USA, Tech. Rep. GT-RIM-CP&R-2012-002, Sep. 2012.
[44] V. Indelman, "Cooperative multi-robot belief space planning for autonomous navigation in unknown environments," *Auton. Robots*, vol. 42, pp. 1–21, 2017.
[45] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

Andrey Zhitnikov received the B.Sc. degree in electrical engineering from the School of Electrical Engineering, Tel Aviv University, Tel Aviv, Israel, in 2014, and the M.Sc. degree in electrical and computer engineering, in 2018, from Technion, Haifa, Israel, where he is currently working toward the Ph.D. degree with Autonomous Navigation and Perception Lab (ANPL).

His current research interest focuses on efficient belief space planning, decision-making under uncertainty, and constrained partially observable Markov decision processes.

Vadim Indelman received the B.A. and B.Sc. degrees in computer science and aerospace engineering, respectively, in 2002, and the Ph.D. degree in aerospace engineering, in 2011, from the Technion—Israel Institute of Technology, Haifa, Israel.

Between 2012 and 2014, he was a Postdoctoral Fellow with the Institute of Robotics and Intelligent Machines (IRIM), Georgia Institute of Technology, Atlanta, GA, USA. He is currently an Associate Professor with the Department of Aerospace Engineering, the Technion—Israel Institute of Technology, and he is also a Member of the Technion Autonomous Systems Program (TASP), the Technion Artificial Intelligence Hub (Tech. AI), and the Israeli Smart Transportation Research Center (ISTRC). In addition, he is a Member of the European Laboratory for Learning and Intelligent Systems (ELLIS). His current research interests include planning under uncertainty, probabilistic inference, semantic perception, and simultaneous localization and mapping (SLAM) in single and multirobot systems.

Dr. Indelman was an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS (RA-L), an Editor for IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), and a Co-Chair of IEEE Robotics and Automation Society Technical Committee on Algorithms for the Planning and Control of Robot Motion.

No Compromise in Solution Quality: Speeding Up Belief-dependent Continuous POMDPs via Adaptive Multilevel Simplification

Andrey Zhitnikov[1], Ori Sztyglic[2], and Vadim Indelman[3]

Abstract

Continuous POMDPs with general belief-dependent rewards are notoriously difficult to solve online. In this paper, we present a complete provable theory of adaptive multilevel simplification for the setting of a given externally constructed belief tree and MCTS that constructs the belief tree on the fly using an exploration technique. Our theory allows to accelerate POMDP planning with belief-dependent rewards without any sacrifice in the quality of the obtained solution. We rigorously prove each theoretical claim in the proposed unified theory. Using the general theoretical results, we present three algorithms to accelerate continuous POMDP online planning with belief-dependent rewards. Our two algorithms, SITH-BSP and LAZY-SITH-BSP, can be utilized on top of any method that constructs a belief tree externally. The third algorithm, SITH-PFT, is an anytime MCTS method that permits to plug-in any exploration technique. All our methods are guaranteed to return exactly the same optimal action as their unsimplified equivalents. We replace the costly computation of information-theoretic rewards with novel adaptive upper and lower bounds which we derive in this paper, and are of independent interest. We show that they are easy to calculate and can be tightened by the demand of our algorithms. Our approach is general; namely, any bounds that monotonically converge to the reward can be utilized to achieve significant speedup without any loss in performance. Our theory and algorithms support the challenging setting of continuous states, actions, and observations. The beliefs can be parametric or general and represented by weighted particles. We demonstrate in simulations a significant speedup in planning compared to baseline approaches with guaranteed identical performance.

Keywords

Decision-making under Uncertainty, Belief Space Planning, POMDP, Belief-dependent Rewards, Planning with Imperfect Information

1 Introduction

EFFICIENTLY solving Partially Observable Markov Decision Processes (POMDPs) implies enabling autonomous agents and robots to plan under uncertainty (Smith and Simmons 2004; Kurniawati et al. 2008; Silver and Veness 2010; Ye et al. 2017; Sunberg and Kochenderfer 2018; Garg et al. 2019). Typical sources of uncertainty are the imprecise actions, sensor type, sensor noise, imprecise models, and unknown agent surroundings. However, solving a POMDP is notoriously hard. Specifically, it was proven to be PSPACE-complete (Papadimitriou and Tsitsiklis 1987).

The actual POMDP state is hidden. Instead, at each time step, the robot shall decide which action to take based on the distribution over the state, given the corresponding history of performed actions and observations received so far. Such a distribution received the name "belief". In a planning session, the robot has to take into account all possible future actions interleaved with possible observations. Each such future history of the length of predefined horizon defines a lace of the future beliefs (blue lace in Fig. 1) and corresponding cumulative rewards named return. Solving POMDP in the most common sense means finding a mapping from belief to action called policy, which maximizes the expected return.

Earlier *offline* solvers such as (Smith and Simmons 2004; Kurniawati et al. 2008) are applicable to small or moderately sized discrete POMDP. These methods require passage over all possible states and observations (Kochenderfer et al. 2022) since they are built on value iteration of α-vectors, so called full-width methods (Silver and Veness 2010). More recent *online* solvers are suitable for POMDPs with large but discrete action, state, and observation spaces (Ye et al. 2017; Silver and Veness 2010). Still, continuous state, action, and observation spaces remain to be an open problem (Sunberg and Kochenderfer 2018). Another challenging aspect of solving POMDP and the subject of interest in this paper is general belief distributions represented by weighted particles. Further in the manuscript we will regard the combination of both, nonparametric beliefs and a fully continuous POMDP as a **nonparametric fully continuous** setting.

[1] Technion Autonomous Systems Program (TASP), Technion - Israel Institute of Technology
[2] Department of Computer Science, Technion - Israel Institute of Technology
[3] Department of Aerospace Engineering, Technion - Israel Institute of Technology

Corresponding author:
Andrey Zhitnikov, Technion - Israel Institute of Technology, Haifa 3200003, Israel.
Email: andreyz@campus.technion.ac.il

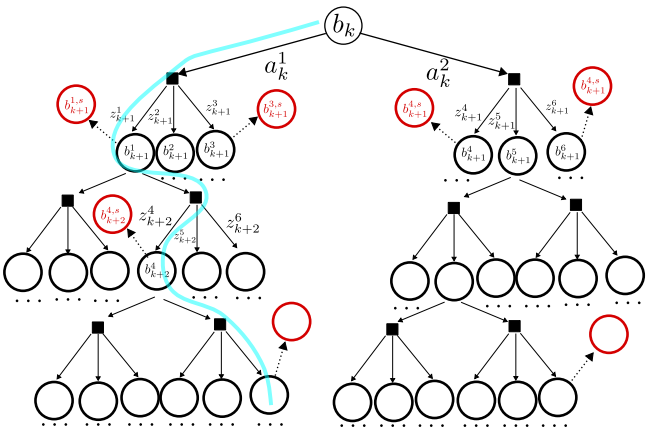


Figure 1. Schematic visualization of the belief tree and the inplace simplification. The superscript in this visualization denotes the index in the belief tree. By b^s we denote the simplified version of the belief b.

In a fully continuous setting with parametric or general beliefs one shall resort to sampling of future possible actions and observations. In a sampled form, this abundance of possible realizations of action-observation pairs constitutes a *belief tree*. Building the full belief tree is intractable since each node in the tree repeatedly branches with all possible actions and all possible observations as illustrated in Fig. 1. The number of nodes grows exponentially with the horizon. This problem is known as the *curse of history*.

The reward function in a classical POMDP is assumed to have a specific structure, namely, to be the expectation with respect to the belief of the state-dependent reward function. While alleviating the solution, this formulation does not support more general, belief-dependent reward functions, such as information-theoretic rewards.

However, POMDP planning with belief-dependent rewards is essential for various problems in robotics and Artificial Intelligence (AI), such as informative planning (Hollinger and Sukhatme 2014), active localization (Burgard et al. 1997), active Simultaneous Localization and Mapping (SLAM) (Stachniss et al. 2005), Belief Space Planning (BSP) (Indelman et al. 2015; Van Den Berg et al. 2012; Platt et al. 2010). The authors of (Araya et al. 2010) provide an extensive motivation for general belief-dependent rewards. One of the widely used such rewards is Information Gain, which involves the difference between differential entropies of two consecutive in time beliefs. Such a reward is crucial in exploration tasks because, in these tasks, the robot's goal is to decrease uncertainty over the belief. For instance, uncertainty measures such as differential entropy and determinant of the covariance matrix of the belief cannot be represented as expectation over a state-dependent reward with respect to the belief. Another example of a belief-dependent reward is entropy over discrete variables that correspond to data association hypotheses (Pathak et al. 2018). Computationally-efficient information-theoretic BSP approaches have been investigated in recent years, considering Gaussian distributions (Kopitkov and Indelman 2017, 2019; Elimelech and Indelman 2022; Kitanov and Indelman 2024).

Yet, POMDP planning with general belief-dependent rewards in particular, when the beliefs are represented by particles exacerbate the computational challenge of the solution even more. For example information theoretic rewards such as differential entropy, are computationally expensive.

Let us focus for the moment on differential entropy. Even if the belief is parametric but not Gaussian, calculating the exact value of differential entropy involves intractable integrals. This fact also motivates to use a weighted particles representation for the belief. In this case differential entropy can be estimated, for instance by Kernel Density Estimation (KDE) (Fischer and Tas 2020) or a model-based estimator (Boers et al. 2010). However, these estimators have quadratic cost in the number of samples and are usually the bottleneck of planning algorithms. The reason is that this increased computational burden is incurred for all nodes in the belief tree. Importantly, the estimation errors of these estimators with respect to differential entropy over theoretical belief are out of the reach due to the unavailability of both, the theoretical belief and the entropy on top of it. Yet, due to the convergence of the belief represented by particles to the theoretical belief (almost sure convergence (Crisan and Doucet 2002)), the mentioned above estimators converge to the exact differential entropy. This prompts us to use **as many belief particles as possible** to get closer to the theoretical belief. Nevertheless, increasing the number of belief particles severely impacts planning time.

In this paper we accelerate online decision making in the setting of nonparametric fully continuous POMDPs with general belief dependent rewards. Crucially, planning performance of our accelerated approach is the same as that of the baseline approaches without our acceleration. Before stating our contributions, we review the most relevant works in this context.

1.1 Related Work

Allowing general belief-dependent rewards in POMDP while solving such a problem efficiently is a long standing effort. Some previous seminal works such as ρ-POMDP (Araya et al. 2010; Fehr et al. 2018) as well as (Dressel and Kochenderfer 2017) have focused on discrete domains, small sized spaces and have tackled the offline solvers. Furthermore, these approaches are limited to piecewise linear and convex or Lipschitz-continuous rewards. Another work named POMDP-IR (Spaan et al. 2015) suggest an interesting framework for specific form of information rewards involving manipulations on the action space. Still, in (Araya et al. 2010; Fehr et al. 2018; Dressel and Kochenderfer 2017) the state, action and observation spaces are discrete and small sized. Another line of works is Belief Space Planning (BSP) (Platt et al. 2010; Van Den Berg et al. 2012; Indelman et al. 2015). These approaches are designed for fully continuous POMDPs, but limited to Gaussian beliefs. In striking contrast, our approach is centered in the more challenging fully continuous domain and nonparametric general beliefs represented by particles while at the same time our framework is general and supports also exact parametric beliefs.

One way to tackle a nonparametric fully continuous setting with belief dependent rewards is to reformulate POMDP as a Belief-MDP (BMDP). On top of this reformulation one can utilize MDP sampling based methods

such as Sparse Sampling (SS) proposed by Kearns et al. (2002). However, this algorithm still suffers from the curse of history and such that increasing the horizon is still problematic.

Monte Carlo Tree Search (MCTS) made a significant breakthrough in overcoming the course of history by building the belief tree incrementally and exploring only the "promising" parts of the tree using the exploration strategy. An inherent part of MCTS based algorithms is the exploration strategy designed to balance exploration and exploitation while building the belief tree. Most widely used exploration technique is Upper Confidence Bound (UCB) (Kocsis and Szepesvári 2006).

MCTS algorithms assume that calculating the reward over the belief node does not pose any computational difficulty. Information-theoretic rewards violate this assumption. When the reward is a general function of the belief, the origin of the computational burden is shifted towards the reward calculation. Moreover, in a non-parametric setting, belief-dependent rewards require a complete set of belief particles at each node in the belief tree. Therefore, algorithms such as POMCP (Silver and Veness 2010), and its numerous predecessors are inapplicable since they simulate each time a single particle down the tree when expanding it. DESPOT based algorithms behave similarly (Ye et al. 2017), with the DESPOT-α as an exception (Garg et al. 2019). DESPOT-α simulates a complete set of particles. However, the DESPOT-α tree is built using α-vectors, such that they are an indispensable part of the algorithm. The standard α-vectors technique requires that the reward is state dependent, and the reward over the belief is merely expectation over the state reward. In other words, DESPOT-α does not support belief-dependent rewards since it contradicts the application of the α-vectors.

The only approach posing no restrictions on the structure of belief-dependent reward and not suffering from limiting assumptions is Particle Filter Tree (PFT). The idea behind PFT is to apply MCTS over Belief-MDP (BMDP). The authors of (Sunberg and Kochenderfer 2018) augmented PFT with Double Progressive Widening (DPW) to support continuous spaces in terms of actions, states and observations, and coined the name PFT-DPW. PFT-DPW utilizes the UCB strategy and maintains a complete belief particle set at each belief tree node. Recently, Fischer and Tas (2020) presented Information Particle Filter Tree (IPFT), a method to incorporate information-theoretic rewards into PFT. The IPFT simulates small subsets of particles sampled from the root of the belief tree and averages entropies calculated over these subsets, enjoying a fast runtime. However, differential entropy estimated from a small-sized particle set can be significantly biased. This bias is unpredictable and unbounded, therefore, severely impairs the performance of the algorithm. In other words, celerity comes at the expense of quality. Oftentimes, the policy defined by this algorithm is very far from optimal given a time budget. Fischer and Tas (2020) provides guarantees solely for the asymptotic case, i.e, the number of subsampled from the root belief state samples (particles) tends to infinity. Asymptotically their algorithm behaves precisely as the PFT-DPW in terms of running speed and performance. Yet, in practice the performance of IPFT in terms of optimality can degrade severely compared to PFT-DPW. Moreover, Fischer and Tas (2020) does not provide any study of comparison of IPFT against PFT-DPW with an information-theoretic reward. Another undesired characteristic of IPFT is that the averaging of the differential entropies is done implicitly and the number of averaged entropies per belief is the visitation count of the corresponding belief. Therefore, to properly compare IPFT with PFT-DPW one shall increase the number of simulations inside IPFT algorithm. We explain this aspect more thoroughly in Section 8.3.5. Prompted by these insights, we chose the PFT-DPW as our *baseline* approach, which we aim to accelerate. In contrast to IPFT designed specifically for differential entropy, our approach is suitable for any belief dependent reward and explicitly guarantees an *identical* solution to PFT-DPW with an information-theoretic reward, for *any* size of particle set representing the belief and serving as input to PFT-DPW.

The computational burden incurred by the complexity of POMDP planning inspired many research works to focus on approximations of the problem on top of existing solvers, e.g., multilevel successive approximation of a motion model (Hoerger et al. 2019), lazy belief extraction on top of a particle based representation (Hoerger and Kurniawati 2021), linearity based solvers (Hoerger et al. 2020), and averaging differential entropy estimated from tiny subsets of particles (Fischer and Tas 2020). Typically, these works provide only asymptotical guarantees (Hoerger et al. 2019; Fischer and Tas 2020), or no guarantees at all. In addition many of these approximations leverage the assumption that the belief-dependent reward is an averaged state-dependent reward, e.g, (Hoerger et al. 2019; Hoerger and Kurniawati 2021), and therefore cannot accommodate belief dependent-rewards with general structure (e.g. do not support information-theoretic rewards such as differential entropy).

Recently, the novel paradigm of *simplification* has appeared in literature (Zhitnikov and Indelman 2022b; Barenboim and Indelman 2022, 2023; Zhitnikov and Indelman 2024; Sztyglic and Indelman 2022; Elimelech and Indelman 2022; Shienman and Indelman 2022; Kitanov and Indelman 2024; Lev-Yehudi et al. 2024). The simplification is concerned with carefully replacing the nonessential elements of the decision making problem and quantifying the impact of this relaxation. Specifically, simplification methods are accompanied by stringent guarantees. A prominent aspect of a simplification paradigm is the usage of the bounds over the reward or the objective function. As opposed to approximations, the simplification framework always keeps some sort of connection to the original unsimplified problem and by that provides deterministic guarantees relative to the given solver. Despite that various objective function bounds have been practiced in (Ye et al. 2017; Smith and Simmons 2004; Walsh et al. 2010; Kochenderfer et al. 2022), these techniques are not applicable in the realm of belief-dependent rewards and a fully continuous setting. In addition commonly these approaches assume that the state dependent reward is trivially bounded from below and above by some constant.

1.2 Contributions

This work is about accelerating online decision making while obtaining exactly the same solution as without acceleration. Specifically, we contribute an adaptive multi-level simplification framework that accounts for belief-dependent rewards, possibly nonparametric beliefs, and continuous state, observation and action spaces.

Our framework accepts as input adaptive monotonical and computationally inexpensive bounds over the exact or estimated reward. Given such reward bounds, it accelerates online decision-making. Specifically, given such adaptive monotonical reward bounds, it is possible to adaptively bound the value function for any given policy and expedite decision-making. If the value function bounds for different candidate policies do not overlap, we do not pay in terms of quality, namely, we obtain the same solution as the equivalent unsimplified method. In the case these bounds do overlap, then we can progressively tighten them by invoking a process that we shall call simplification adaptation or *resimplification* until they no longer overlap.

Our techniques return exactly the same solution as the unsimplified equivalent. Such an unsimplified baseline can correspond to decision-making problems where the reward can be exactly calculated (analytically), or where the reward is estimated. In either case, if the bounds over the corresponding reward are provided and satisfy the assumptions stated in Section 3.3, one can apply our framework to speedup the decision making process while obtaining the same best action as with the original rewards instead of the bounds. Such a capability is therefore particularly appealing in light of the information-theoretic rewards that are essential in numerous problems in robotics, but are often the computational bottleneck.

Further, there are two settings that we separately and explicitly discuss in this paper. We start from a given belief tree, that can be constructed by a POMDP solver that is not coupled with the solution, e.g., SS. In this setting we can prune branches of the belief tree whenever the mentioned objective bounds for different candidate policies or actions do not overlap.

We then discuss an anytime setting of MCTS, where the belief tree construction is coupled with the solution due to an exploration strategy (e.g. UCB). The exploration strategy builds upon an exploratory objective. Since the exploratory objective typically requires access to the objective estimates to select an action at each arrival to a belief node, we cannot prune suboptimal candidate actions. Instead, we can only dismiss them until the next arrival to this belief node. The simplification and reward bounds are used here to bound the exploratory objective and the value function at the root of the belief tree.

Finally, we focus on a specific simplification of nonparametric beliefs represented by particles and a differential entropy estimator as the reward function. Our simplification is subsampling of the original belief to a smaller sample size. We contribute novel computationally cheaper bounds over the differential entropy estimator on top of such a simplified belief and incorporate these bounds into our framework. By that we produce a specific embodiment of the general framework presented earlier.

To summarize, we list down the contributions of this work, in the order they are presented in the manuscript.

1. Building on **any** adaptive monotonically convergent bounds over belief-dependent reward, we present in this paper a **provable** general theory of adaptive multilevel simplification with deterministic performance guarantees.
2. For the case of a given belief tree as in Sparse Sampling, we develop two algorithms, Simplified Information Theoretic Belief Space Planning (SITH-BSP) and a faster variant, LAZY-SITH-BSP. Both are complementary to any POMDP solver that does not couple belief tree construction with an objective estimation while exhibiting a significant speedup in planning with a guaranteed same planning performance.
3. In the context of MCTS, we embed the theory of simplification into the PFT-DPW algorithm and introduce SITH-PFT. We provide stringent guarantees that exactly the same belief tree is constructed by SITH-PFT and PFT-DPW. We focus on a UCB exportation technique, but with minor adjustments, an MCTS with any exploration method will be suitable for acceleration.
4. We derive novel lightweight adaptive bounds on the differential entropy estimator of (Boers et al. 2010) and prove the bounds presented are monotonic and convergent. Moreover, these bounds can be incrementally tightened. We believe these bounds are of interest on their own. The bounds are calculated using the simplified belief (See Fig. 1). We emphasize that any other bounds fulfilling assumptions declared in Section 3.3 can be utilized within our framework.
5. We present extensive simulations that exhibit a significant improvement in planning time without any sacrifice in planning performance.

This paper is an extension of the work presented in (Sztyglic and Indelman 2022), which proposed novel adaptive bounds on the differential entropy estimator of (Boers et al. 2010) and introduced the simplification paradigm in the context of a given belief tree. To be precise we explicitly clarify how this work differs from the conference version of this paper (Sztyglic and Indelman 2022). In this version, we extend the simplification framework to the rewards depending on a pair of consecutive-in-time beliefs, e.g., Information Gain as opposed to the conference version where such an extension was only mentioned. In this version, we provide alternative proof of these bounds and prove that these reward bounds are monotonic. In the setting of a given belief tree we present an additional algorithm, that we call LAZY-BSP. This algorithm is faster than SITH-BSP suggested in (Sztyglic and Indelman 2022). Importantly, we extend our simplification framework to support also anytime MCTS planners. Additionally, we provide extensive performance evaluation of our methods in simulations.

1.3 Paper Organization

The remainder of this paper is structured as follows. Section 2 provides background in terms of POMDPs, theoretical objective and commonly used objective estimators. We

devote Section 3 to our general adaptive multi-level simplification framework. In Section 4 we consider a given belief tree setting in which the belief tree construction is not coupled with the solution. In Section 5 we delve into the MCTS approach in the context of our multilevel simplification. In Section 6 we consider a specific simplification and develop novel bounds on an information-theoretic reward function. Section 7 assesses the general adaptation overhead of our methodology. Finally, Section 8 presents simulations and results corroborating our ideas. In order not to disrupt the flow of the presentation, proofs are presented in appropriate Appendices.

2 Background

In this section we present the background. To elaborate, we present a POMDP with belief dependent rewards followed by theoretical and estimated objectives that correspond to different online POMDP solvers. Our techniques work with estimated objectives.

2.1 POMDPs with Belief-dependent Rewards

A POMDP is a tuple

$$\langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, \rho, \gamma, b_0 \rangle \tag{1}$$

where $\mathcal{X}, \mathcal{A}, \mathcal{Z}$ are state, action, and observation spaces, respectively. In this paper we consider continuous state, observation and action spaces. $T(x, a, x') = \mathbb{P}_T(x'|x, a)$ is the stochastic transition model from the past state x to the subsequent x' through action a, $O(z, x) = \mathbb{P}_O(z|x)$ is the stochastic observation model, $\gamma \in (0, 1]$ is the discount factor, b_0 is the belief over the initial state (prior), and ρ is the reward function. Let $h_k = \{b_0, a_0, z_1, \dots, a_{k-1}, z_k\}$ denote *history* of actions and observations obtained by the agent up to time instance k and the prior belief. The posterior belief at time instant k is given by $b_k(x_k) = \mathbb{P}(x_k|h_k)$.

In our generalized formulation, the reward is a function of two subsequent in time beliefs, an action and an observation:

$$\rho(b_k, a_k, z_{k+1}, b_{k+1}) = (1-\lambda)\rho^x(b_k, a_k, b_{k+1}) + \tag{2}$$
$$+\lambda\rho^I(b_k, a_k, z_{k+1}, b_{k+1}), \tag{3}$$

where $\lambda \geq 0$. The first reward component $\rho^x(b_k, a_k, b_{k+1})$ is the expectation over the state and action dependent reward $r(x_k, a_k)$ or $r(a_k, x_{k+1})$. Correspondingly, these two possibilities yield

$$\rho^x(b_k, a_k, b_{k+1}) = \mathop{\mathbb{E}}_{x_k \sim b_k}[r(x_k, a_k)] \approx \frac{1}{n_x}\sum_{\xi=1}^{n_x} r(x_k^\xi, a_k), \tag{4}$$

or

$$\rho^x(b_k, a_k, b_{k+1}) = \mathop{\mathbb{E}}_{x_{k+1} \sim b_{k+1}}[r(a_k, x_{k+1})] \approx \frac{1}{n_x}\sum_{\xi=1}^{n_x} r(a_k, x_{k+1}^\xi). \tag{5}$$

which is commonly approximated by sample mean using n_x samples of the belief. The second reward component $\rho^I(b_k, a_k, z_{k+1}, b_{k+1})$ is an information-theoretic reward weighted by λ, which in general can be dependent on consecutive beliefs and the elements relating them, e.g. information gain or specific estimators as (Boers et al. 2010) for nonparametric beliefs represented by particles. For instance, in Section 6.1 we consider the entropy estimator introduced by Boers et al. (2010). As will be seen in the sequel, although the theoretical entropy is only a function of a single belief b_{k+1}, the mentioned estimator utilizes b_k, a_k, z_{k+1} and b_{k+1}; hence the second reward component, $\rho^I(b_k, a_k, z_{k+1}, b_{k+1})$, depends on these quantities.

The *policy* is a mapping from belief to action spaces $a_k = \pi_k(b_k)$. Let $\pi_{\ell+}$ be a shorthand for policy for $\ell - k + L$ consecutive steps ahead starting at index ℓ, namely $\pi_{\ell:k+L-1}$ for $\ell \geq k$.

2.2 Theoretical Objective

The decision making goal is to find an optimal policy π_{k+} maximizing the value function as such:

$$V(b_k, \pi_{k+}) \text{ s.t. } b_{\ell+1} = \psi(b_\ell, \pi_\ell(b_\ell), z_{\ell+1}), \tag{6}$$

where $V(b_k, \pi_k)$ is defined by

$$\mathop{\mathbb{E}}_{z_{k+1:k+L}}\Big[\sum_{\ell=k}^{k+L-1}\gamma^{\ell-k}\rho(b_\ell, \pi_\ell(b_\ell), z_{\ell+1}, b_{\ell+1})|b_k, \pi_{k+}\Big] \tag{7}$$

and ψ is the Bayesian belief update method. Utilizing the Bellman formulation (7) takes the form of

$$V(b_k, \pi_{k+}) = \mathop{\mathbb{E}}_{z_{k+1}}\big[\rho(b_k, \pi_k(b_k), z_{k+1}, b_{k+1})|b_k, \pi_k\big] + \\ +\gamma \mathop{\mathbb{E}}_{z_{k+1}}\big[V(\psi(b_k, a_k, z_{k+1}), \pi_{(k+1)+})|b_k, \pi_k\big]. \tag{8}$$

The action-value function under arbitrary policy is given by

$$Q(b_k, \{a_k, \pi_{(k+1)+}\}) = \mathop{\mathbb{E}}_{z_{k+1}}\big[\rho(b_k, a_k, z_{k+1}, b_{k+1})|b_k, a_k\big] + \\ +\gamma \mathop{\mathbb{E}}_{z_{k+1}}\big[V(\psi(b_k, a_k, z_{k+1}), \pi_{(k+1)+})|b_k, a_k\big]. \tag{9}$$

The relation between (8) and (9) is $V(b_k, \pi_{k+}) = Q(b_k, \{\pi_k(b_k), \pi_{(k+1)+}\})$. If π is the optimal policy we denote it by π^*. For clarity, let us designate for action-value function under optimal future policy $Q(b_k, \{a_k, \pi^*_{(k+1)+}\})$ a short notation $Q(b_k, a_k)$. If $Q(b_k, a_k)$ can be calculated, the online POMDP solution for the current belief b_k will be

$$\pi_k^*(b_k) \in \arg\max_{a_k} Q(b_k, a_k). \tag{10}$$

Linearity of the expectation and the structure displayed by equations (2) and (3) lead to a similar decomposition of action-value function (9) as such

$$Q(\cdot) = (1-\lambda)Q^x(\cdot) + \lambda Q^I(\cdot), \tag{11}$$

where Q^x is induced by state dependent rewards and Q^I by the information-theoretic rewards.

From here on, for the sake of clarity, we will use the notation of history h_k and the belief b_k interchangeably for any time k. In a similar manner, we shall use the notations b_k, a_k and $h_k a_k$ interchangeably.

2.3 Estimated Objective

The continuous observation space makes the theoretical expectations in (7) and (9) attainable in very limited cases.

Generally we shall resort to estimators. Similar to theoretical counterparts, the relation between the estimated optimal value and action-value function reads

$$\hat{V}(b_k, \pi^*_{k+}) = \max_{a_k} \hat{Q}(b_k, a_k). \tag{12}$$

Also in Eq. (10), the theoretical $Q(b_k, a_k)$ is substituted by the estimator $\hat{Q}(b_k, a_k)$. Naturally, we expect from the estimator to admit the decomposition

$$\hat{Q}(b_k, a_k) = (1-\lambda)\hat{Q}^x(b_k, a_k) + \lambda \hat{Q}^I(b_k, a_k). \tag{13}$$

Typically the $\hat{Q}^x$ element is easy to calculate, thus it is out of our focus, whereas $\hat{Q}^I$ is computationally expensive to compute.

Below we present two common sample based estimators that will be used in this paper.

2.3.1 Objective Estimator in Case of a Given Belief Tree

We turn to the setting of a given externally-constructed belief tree, e.g. by a SS algorithm. For the sake of clarity and to ease the explanation, we assume that the number of child posterior beliefs is constant at each nonterminal belief and denoted by n_z. Relaxing this assumption is straightforward. The Bellman form representation of (7) using such an estimator is

$$\begin{aligned}\hat{V}(b_k, \pi_{k+}) = &\frac{1}{n_z}\sum_{i=1}^{n_z} \rho(b_k, \pi_k(b_k), z^i_{k+1}, b^i_{k+1}) + \\ &+\gamma \frac{1}{n_z}\sum_{i=1}^{n_z} \hat{V}(\psi(b_k, \pi_k(b_k), z^i_{k+1}), \pi_{(k+1)+}),\end{aligned} \tag{14}$$

and the corresponding estimator for (9) under an optimal future policy reads

$$\begin{aligned}\hat{Q}(b_k, a_k) = &\frac{1}{n_z}\sum_{i=1}^{n_z} \rho(b_k, a_k, z^i_{k+1}, b^i_{k+1}) + \\ &+\gamma \frac{1}{n_z}\sum_{i=1}^{n_z} \hat{V}(\psi(b_k, a_k, z^i_{k+1}), \pi^*_{(k+1)+}),\end{aligned} \tag{15}$$

where n_z is the number of children of b_ℓ under the execution policy $\pi_{\ell+}$ and i is the child index.

2.3.2 Interchangeability Between the history and Belief

The purpose of this section is to clarify why further we will use interchangeably belief and the history. The belief is merely a reinterpretation of the knowledge about the POMDP state stored in history in the form of a PDF. The belief b_k is a function of the history h_k. Therefore different histories may yield the same belief. To avoid ambiguity and relate the objectives and their position in the belief tree with some abuse of notation we sometimes switch the dependence on the belief to dependence on corresponding history. In general we can write $b_\ell(h_\ell)$.

2.3.3 Coupled Action-Value Function Estimation and Belief Tree Construction

The estimator presented above leverages symmetric, in terms of observations, Bellman form. However in MCTS methods due to exploration driven by, for example, UCB (16), the estimators are assembled from laces of the returns. In each simulation a single lace is added to the estimator at each posterior belief. Whenever a new posterior belief node is expanded, a rollout is commenced such that the lace is complemented to the whole horizon.

MCTS repetitively descends down the tree, adding a lace of cumulative rewards (or updates visitation counts of an existing lace) and ascends back to root. On the way down it selects actions according to an exploration strategy e.g., (16). This results in a policy tree, that represents a stochastic policy represented by visitation counts $\frac{N(ha)}{N(h)}$. Further we will focus on UCB exploration strategy, however all derivations of our approach are general and are valid for any exploration strategy, e.g, P-UCT (Auger et al. 2013) or ϵ-greedy exploration (Sutton and Barto 2018).

A UCB-based MCTS over a Belief-MDP (BMDP) (Auer et al. 2002; Sunberg and Kochenderfer 2018) constructs a policy tree by executing multiple simulations. Each simulation adds a single belief node to the belief tree or terminates by terminal state or action. To steer towards more deeper and more beneficial simulations, MCTS selects an action a^* at each belief node according to the following rule $a^* = \arg\max_{a\in\mathcal{A}} \mathrm{UCB}(ha)$ where

$$\mathrm{UCB}(ha) = \hat{Q}(ha) + c\cdot\sqrt{\log(N(h))/N(ha)}, \tag{16}$$

where $N(h)$ is the visitation count of the belief node defined by history h, $N(ha)$ is the visitation count of the belief-action node, c is the exploration parameter and, $\hat{Q}(ha)$ is the estimator of the action-value function Q for node ha obtained by simulations. The rule described by (16) is a result of modelling exploration as Multi Armed Bandit (MAB) problem (Kocsis and Szepesvári 2006; Munos 2014; Auger et al. 2013). When the action is selected, a question arises either to open a new branch in terms of observation and posterior belief or to continue through one of the existing branches. In continuous action, and observation spaces, this can be resolved by the Double Progressive Widening (DPW) technique (Sunberg and Kochenderfer 2018; Auger et al. 2013). If a new branch is expanded, an observation z' is created from state x' drawn from the belief b propagated with an action a.

Let the return, corresponding to lace i starting from some belief b^i_ℓ at depth $\ell - k$, be $g(b^i_\ell, a_\ell, z^i_{\ell+1:k+L})$ for $\ell \in [k : k+L-1]$. More specifically, suppose the new posterior belief was expanded at depth d^i of the belief tree such that $d^i > \ell$. We have that $g(b^i_\ell, a_\ell, z^i_{\ell+1:k+L})$ is composed from two parts, the already expanded tree part and the rollout added part such that

$$g(b^i_\ell, a_\ell, z^i_{\ell+1:k+L}) =$$

$$\underbrace{\rho(b^i_\ell, a_\ell, z^i_{\ell+1}, b^i_{\ell+1}) + \sum_{l=\ell+1}^{k+d^i-1} \gamma^{l-\ell}\rho(b^i_l, \pi^{*,i}_l(b^i_l), z^i_{\ell+1}, b^i_{l+1}) +}_{\text{belief tree}} \tag{17}$$

$$+ \underbrace{\sum_{l=k+d^i}^{k+L-1} \gamma^{l-\ell}\rho(b^i_l, \mu(b^i_l), z^i_{l+1} b^i_{l+1}),}_{\text{rollout}} \tag{18}$$

where L is the horizon (tree depth), $\pi^{*,i}$ is an optimal tree policy depending on the number of the simulation i through

$\hat{Q}$ and visitation counts in (16) and μ is the rollout policy. Importantly, in rollout the observations are drawn randomly and since we are in continuous spaces the beliefs in the rollouts are unique. A new belief node is added for $l = k + d^i$. If due to DPW no new belief node was added to the belief tree, no rollout depicted by (18) is commenced and the return sample takes the form of

$$g(b^i_\ell, a_\ell, z^i_{\ell+1:k+L}) = \rho(b^i_\ell, a_\ell, z^i_{\ell+1}, b^i_{\ell+1}) + \sum_{l=\ell+1}^{k+L-1} \gamma^{l-\ell} \rho(b^i_l, \pi^{*,i}_l(b^i_l), z^i_{l+1}, b^i_{l+1}). \quad (19)$$

The estimate for (9) under optimal future policy is assembled from laces in accordance to

$$\hat{Q}(h_\ell a_\ell) = \frac{1}{N(h_\ell a_\ell)} \sum_{i=1}^{N(h_\ell a_\ell)} g(b^i_\ell, a_\ell, z^i_{\ell+1:k+L}), \quad (20)$$

where each reward $\rho(b, a, z', b')$ in the belief tree appears the number of times according to the visitation count of the node b', namely $N(h')$. We note that for both estimators (15) and (20), the formulation in (13) holds.

Now we move to the details of our general approach.

3 Our Approach

This section is the core of our general approach. We first describe bounds over the theoretical and the estimated objectives. We then endow the rewards bounds with discrete simplification levels. Finally, instead of calculating rewards, we calculate the bounds over them and if they are not tight enough we tighten them so we can make faster decisions with bounds over the objectives instead of objectives themselves.

3.1 Theoretical Simplification Formulation

Simplification is any kind of relaxation of POMDP tuple (1) elements, accompanied by guarantees that quantify the (worst-case or potential) impact of a particular simplification technique on planning performance. In this section, we present a general simplification framework that is applicable to any reward bounds that satisfy the assumptions stated in Section 3.3.

Our framework applies without any change to parametric and non-parametric beliefs, and to closed-form belief-dependent rewards (that can be calculated exactly, i.e. analytically), as well as to estimated rewards. Therefore, in this paper we do not differentiate between these cases and denote the belief-dependent reward by $\rho(b_\ell, a_\ell, z_{\ell+1}, b_{\ell+1})$, without using the notation $\hat{\square}$ for estimators. In other words, depending on the setting, $\rho()$ and b_ℓ can represent, respectively, an analytical reward and a parametric belief, or a reward estimator and a nonparametric belief. In all cases, if one can provide monotonically adaptive bounds on the reward, our framework will return an identical solution as if the decision making was performed with original reward calculations (i.e. depending on the setting, either an analytical reward calculation or reward estimator calculation). In Section 6 we provide a specific incarnation of the framework considering non-parametric beliefs represented by a set of weighted samples and a reward estimator, and where the simplification corresponds to utilizing only a subset of the samples.

As mentioned, we aim to simplify the belief-dependent reward $\rho(b_\ell, a_\ell, z_{\ell+1}, b_{\ell+1})$ calculations. Namely, the original reward ρ is bounded using the simplified belief b^s instead of original belief b. This operation materializes in the form of following inequality

$$\begin{aligned} &\underline{\rho}(b^s_\ell, b_\ell, a_\ell, z_{\ell+1}, b_{\ell+1}, b^s_{\ell+1}) \leq \\ &\leq \rho(b_\ell, a_\ell, z_{\ell+1}, b_{\ell+1}) \leq \\ &\leq \overline{\rho}(b^s_\ell, b_\ell, a_\ell, z_{\ell+1}, b_{\ell+1}, b^s_{\ell+1}), \end{aligned} \quad (21)$$

where $\underline{\rho}$ and $\overline{\rho}$ are the corresponding lower and upper bounds, respectively. The superscript s denotes the fact that the corresponding belief was simplified as we depict in Fig. 1. Notice that in (21) the pair of consecutive beliefs, b_ℓ and $b_{\ell+1}$, can be simplified differently.

Henceforth, in order to avoid unnecessary clutter we will omit the dependence on the observation and denote the bounds over the reward using simplified beliefs as follows

$$\underline{\rho}^s(b, a, b') \leq \rho(b, a, b') \leq \overline{\rho}^s(b, a, b'). \quad (22)$$

It should be stressed that since in the belief tree b' always has a single parent b, the reader should think about such a reward as one corresponding to b'.

A key requirement is reduced computational complexity of these bounds compared to the complexity of the original reward. Instead of calculating the expensive reward $\rho(b, a, b')$ for each pair of beliefs b, b', we first obtain the corresponding simplified beliefs b^s and b'^s, as illustrated in Fig. 1, and then formulate the bounds $\underline{\rho}^s$ and $\overline{\rho}^s$ from (22). However, we note that the form (22) is actually more general and not limited to belief simplification.

Further we formulate bounds over the value function (8) and action-value function (9), both under the optimal policy. In fact, our bounds hold under an arbitrary policy. We narrow the discussion to optimal polices solely for the clarity of the explanation and this is not a limitation of our approach.

Suppose inequality (22) holds for any possible pair of consecutive beliefs, e.g. these are analytical bounds, as opposed to (Zhitnikov and Indelman 2022b). A direct consequence of this fact, alongside the structure of (7), is that

$$\underline{V}(b_\ell, \pi^*_{\ell+}) \leq V(b_\ell, \pi^*_{\ell+}) \leq \overline{V}(b_\ell, \pi^*_{\ell+}), \quad (23)$$

holds for any belief b_ℓ and $\ell \in [k, k+L-1]$. Using the Bellman representation as in (8) the bounds (23) take the form of

$$\begin{aligned} \overline{V}(b_\ell, \pi^*_{\ell+}) &= \mathop{\mathbb{E}}_{z_{\ell+1}} \left[\overline{\rho}^s(b_\ell, \pi^*_\ell(b_\ell), b^i_{\ell+1}) + \overline{V}(b^i_{\ell+1}, \pi^*_{(\ell+1)+}) \right] \\ \underline{V}(b_\ell, \pi^*_{\ell+}) &= \mathop{\mathbb{E}}_{z_{\ell+1}} \left[\underline{\rho}^s(b_\ell, \pi^*_\ell(b_\ell), b^i_{\ell+1}) + \underline{V}(b^i_{\ell+1}, \pi^*_{(\ell+1)+}) \right]. \end{aligned} \quad (24)$$

The bounds over the value function (8) in (24) are initialized at the Lth time step in the planning horizon as $\overline{V}(b_{k+L}, \pi_{k+L}) = 0$ and $\underline{V}(b_{k+L}, \pi_{k+L}) = 0$. Similarly the bounds over the action-value function (9) under an optimal future policy are

$$\underline{Q}\left(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}\right) \leq Q(b_\ell, a_\ell) \leq \overline{Q}\left(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}\right), \quad (25)$$

where the policy $\pi^*_{(\ell+1)+}$ is optimal. Note, as we observe in (24), the simplification assumed herein does not affect the distribution of future observations with respect to which the expectation is taken.

Bounding the Belief Dependent Element of the Reward At this point, we want to recall that commonly, the state-dependent element (2) is much easier to calculate than the belief dependent one. Leveraging the structure manifested by (11) the immediate bounds over (3) induce bounds over $Q^I(\cdot)$ as such

$$\underline{Q}^I(b_k,a_k) \leq Q^I(b_k,a_k) \leq \overline{Q}^I(b_k,a_k), \tag{26}$$

and utilizing (11) we arrive at

$$\overline{Q}(b_k,a_k) = (1-\lambda)Q^x(b_k,a_k) + \lambda\overline{Q}^I(b_k,a_k) \tag{27}$$

$$\underline{Q}(b_k,a_k) = (1-\lambda)Q^x(b_k,a_k) + \lambda\underline{Q}^I(b_k,a_k). \tag{28}$$

Importantly, the belief dependent element (3) does not have to be information-theoretic. The simplification paradigm is general and works for any belief-dependent operator given appropriate bounds.

3.2 Bounds over the Estimated Objective

As we explained in section 2.3 in practice the value and action-value function are estimated. Instead of using (23) and (25) we have

$$\underline{\hat{V}}\left(b_\ell,\pi^*_{\ell+}\right) \leq \hat{V}\left(b_\ell,\pi^*_{\ell+}\right) \leq \overline{\hat{V}}\left(b_\ell,\pi^*_{\ell+}\right), \tag{29}$$

and

$$\underline{\hat{Q}}\left(b_\ell,\{a_\ell,\pi^*_{(\ell+1)+}\}\right) \leq \hat{Q}\left(b_\ell,a_\ell\right) \leq \overline{\hat{Q}}\left(b_\ell,\{a_\ell,\pi^*_{(\ell+1)+}\}\right), \tag{30}$$

respectively.

The bounds, in case of symmetric estimators from section 2.3.1, are

$$\begin{aligned}
\overline{\hat{V}}(b_\ell,\pi^*_{\ell+}) &= \frac{1}{n_z}\sum_{i=1}^{n_z}\overline{\rho}^s(b_\ell,\pi^*(b_\ell),b^i_{\ell+1}) + \\
&+\gamma\frac{1}{n_z}\sum_{i=1}^{n_z}\overline{\hat{V}}(b^i_{\ell+1},\pi^*_{(\ell+1)+}) \\
\underline{\hat{V}}(b_\ell,\pi^*_{\ell+}) &= \frac{1}{n_z}\sum_{i=1}^{n_z}\underline{\rho}^s(b_\ell,\pi^*(b_\ell),b^i_{\ell+1}) + \\
&+\gamma\frac{1}{n_z}\sum_{i=1}^{n_z}\underline{\hat{V}}(b^i_{\ell+1},\pi^*_{(\ell+1)+}),
\end{aligned} \tag{31}$$

where, to clarify we repeat that n_z is the number of children of b_ℓ under the execution policy $\pi_{\ell+}$ and i is the child index. The bounds over the estimated value function in (31) are initialized at the Lth time step in the planning horizon as $\overline{\hat{V}}(b_{k+L},\pi_{k+L}) = 0$ and $\underline{\hat{V}}(b_{k+L},\pi_{k+L}) = 0$.

In a similar manner we define also bounds over (15) as such

$$\begin{aligned}
\overline{\hat{Q}}(b_\ell,\{a_\ell,\pi^*_{(\ell+1)+}\}) &= \frac{1}{n_z}\sum_{i=1}^{n_z}\overline{\rho}^s(b_\ell,a_\ell,b^i_{\ell+1}) + \\
&+\gamma\frac{1}{n_z}\sum_{i=1}^{n_z}\overline{\hat{V}}(b^i_{\ell+1},\pi^*_{(\ell+1)+}) \\
\underline{\hat{Q}}(b_\ell,\{a_\ell,\pi^*_{(\ell+1)+}\}) &= \frac{1}{n_z}\sum_{i=1}^{n_z}\underline{\rho}^s(b_\ell,a_\ell,b^i_{\ell+1}) + \\
&+\gamma\frac{1}{n_z}\sum_{i=1}^{n_z}\underline{\hat{V}}(b^i_{\ell+1},\pi^*_{(\ell+1)+}),
\end{aligned} \tag{32}$$

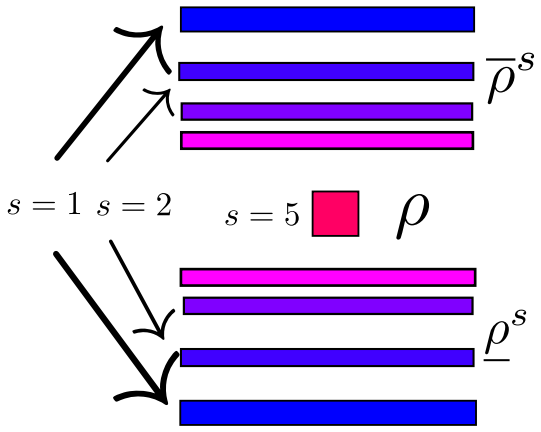


Figure 2. Reward bounds and different levels of the simplification. Here $n_{\max}=5$. Warmer colors visualize tighter bounds. Whereas colder colors (blue) indicate looser bounds and cheaper to calculate.

We emphasize that the superscript i in (31) and (32) denotes the child posterior nodes of b_ℓ.

The bounds over MCTS estimator (20) are

$$\begin{aligned}
\overline{\hat{Q}}(ha) &= \frac{1}{N(ha)}\sum_{i=1}^{N(ha)}\Big(\overline{\rho}^s(b^i_\ell,a_\ell,b^i_{\ell+1}) + \\
&+\sum_{l=\ell+1}^{k+d^i-1}\gamma^{l-\ell}\overline{\rho}^s(b^i_l,\pi^{*,i}_l(b^i_l),b^i_{l+1}) + \sum_{l=k+d^i}^{k+L-1}\gamma^{l-\ell}\overline{\rho}^s(b^i_l,\mu(b^i_l),b^i_{l+1})\Big) \\
\underline{\hat{Q}}(ha) &= \frac{1}{N(ha)}\sum_{i=1}^{N(ha)}\Big(\underline{\rho}^s(b^i_\ell,a_\ell,b^i_{\ell+1}) + \\
&+\sum_{l=\ell+1}^{k+d^i-1}\gamma^{l-\ell}\underline{\rho}^s(b^i_l,\pi^{*,i}_l(b^i_l),b^i_{l+1}) + \sum_{l=k+d^i}^{k+L-1}\gamma^{l-\ell}\underline{\rho}^s(b^i_l,\mu(b^i_l),b^i_{l+1})\Big).
\end{aligned} \tag{33}$$

Let us clarify again that in (33) the superscript i denotes the number of the simulation. Moreover, the reward bounds within the tree repeat in more than a single simulation according to the visitation count of the corresponding posterior belief. Clearly, the decomposition displayed by Eq. (27) and (28) is valid for both bounds (32) and (33). We have that

$$\overline{\hat{Q}}(b_k,a_k) = (1-\lambda)\hat{Q}^x(b_k,a_k) + \lambda\overline{\hat{Q}}^I(b_k,a_k) \tag{34}$$

$$\underline{\hat{Q}}(b_k,a_k) = (1-\lambda)\hat{Q}^x(b_k,a_k) + \lambda\underline{\hat{Q}}^I(b_k,a_k). \tag{35}$$

Impact of the Information Weight λ Allow us to linger on the λ from eq. 11 and 13. It is hard to predict how the objective will behave with various values of λ. Nevertheless, if the bounds are over the belief-dependent element of the reward, by subtracting (35) from (34), we arrive at

$$\overline{\hat{Q}}(b_k,a_k) - \underline{\hat{Q}}(b_k,a_k) = \lambda\left(\overline{\hat{Q}}^I(b_k,a_k) - \underline{\hat{Q}}^I(b_k,a_k)\right). \tag{36}$$

The width of the bounds is monotonically increasing with λ. Of course, it will also happen to a theoretical analog of such a bounds displayed by eq. (27) and (28). We can envision more speedup from applying the simplification paradigm with lower values of λ and will see it in the simulations.

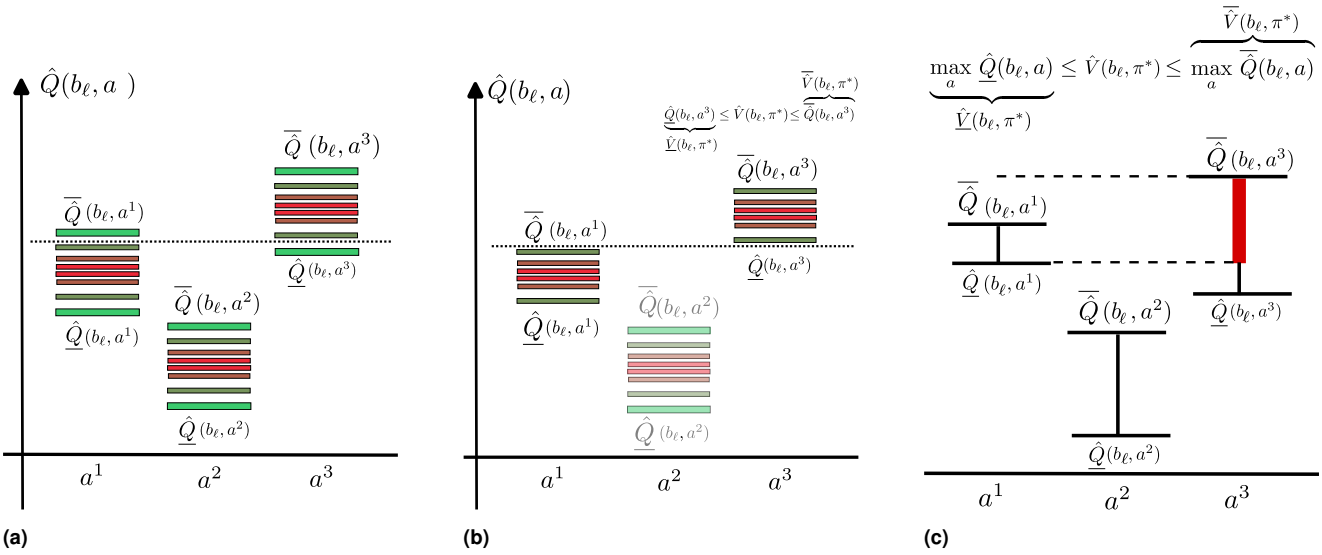


Figure 3. In this illustration we have three candidate actions $\{a^1, a^2, a^3\}$ that can possibly be taken by the robot from the belief node b_ℓ.**(a)** We observe that $\overline{Q}(b_\ell, a^1_\ell) > \underline{Q}(b_\ell, a^3_\ell)$ and prune action a^2. **(b)** After the resimplification no overlap an we can safely decide that a^3_ℓ is optimal. Moreover we prune the withered interval corresponding to the a^2_ℓ. **(c)** Another situation where we are not concerned about optimal action, we solely want to send up to the tree the bounds over optimal value function.

Further we will consider the estimated action-value or value functions and therefore omit the word "estimated". We will also omit mentioning each time that our bounds are under the optimal policy.

3.3 Multi-Level Simplification

We now extend the definition of simplification as we envision it to be an *adaptive paradigm*. We denote *level of simplification* as how "aggressive" the suggested simplification is. Observe an illustration in Fig. 2.

With this setting, we can naturally define many discrete levels such that $s \in \{1, 2, \dots, n_{\max}\}$ represents the simplification level, where 1 and $n_{\max}$ correspond to the coarsest and finest simplification levels, respectively. For instance, suppose the belief is represented by a set of samples (particles), as in Section 6. Taking a small subset of particles to represent the simplified belief corresponds to a *coarse* simplification. If one takes many particles, this corresponds to a *fine* simplification.

Remark: From now on the superscript s denotes the discrete simplification level. Importantly we always have a **finite** number, denoted by $n_{\max}$, of simplification levels.

Further, we assume bounds monotonically become tighter as the simplification level is increased and that the bounds for the finest simplification level $n_{\max}$ converge to the original reward without simplification. More formally, denote $\overline{\Delta}^s(b, a, b') \triangleq \overline{\rho}^s(b, a, b') - \rho(b, a, b')$ and $\underline{\Delta}^s(b, a, b') \triangleq \rho(b, a, b') - \underline{\rho}^s(b, a, b')$.

Assumption 1. Monotonicity. Let $n_{\max} \geq 2$, $\forall s \in [1, n_{\max} - 1]$ we get: $\overline{\Delta}^s(b, a, b') \geq \overline{\Delta}^{s+1}(b, a, b')$ and $\underline{\Delta}^s(b, a, b') \geq \underline{\Delta}^{s+1}(b, a, b')$.

Assumption 2. Convergence. $\forall b, a, b'$ we get: $\overline{\rho}^{s=n_{\max}}(b, a, b') = \underline{\rho}^{s=n_{\max}}(b, a, b') = \rho(b, a, b')$.

In Section 6, we derive novel bounds on top of a particular simplification that takes a subset of belief samples instead of a complete set. We prove that these bounds indeed satisfy both assumptions.

The simplification levels of the reward bounds for different posterior belief nodes in the belief tree determine how tight the bounds over the value or action-value function are. To tighten the bounds over the objective, we have the freedom to select any rewards the belief tree and tighten the bounds over these selected rewards by increasing their simplification levels; this, in turn, would contract the bounds over the objective.

We call a particular algorithmic scheme to select the rewards a **resimplification strategy**. A general valid resimplificaiton strategy is defined as follows.

Definition 1. Resimplification strategy. Given a pair of lower $\underline{\hat{V}}(b_\ell, \pi_{\ell+})$ $(\underline{\hat{Q}}(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}))$ and upper bounds $\overline{\hat{V}}(b_\ell, \pi_{\ell+})$ $(\overline{\hat{Q}}(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}))$ over the estimated objective, the resimplification strategy is a rule to promote one or more simplification levels of the rewards in the the subtree rooted at b_ℓ and defined by the mentioned above estimated objective. If the resimplification does not promote the simplification level for any reward, so $\overline{\hat{Q}}(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}) - \underline{\hat{Q}}(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}) = 0$.

Note that, all the rewards within a subtree defined by $\overline{\hat{Q}}(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}), \underline{\hat{Q}}(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\})$ are being at the maximal simplification level implies $\overline{\hat{Q}}(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}) - \underline{\hat{Q}}(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}) = 0$, but the inverse implication is not necessarily true. Once initiated, a **valid** strategy can select no reward for simplification level promotion only if $\overline{\hat{Q}}(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}) - \underline{\hat{Q}}(b_\ell, \{a_\ell, \pi^*_{(\ell+1)+}\}) = 0$.

Theorem 1. Monotonicity and Convergence of Estimated Objective Function Bounds. *If the bounds over the reward are monotonic (assumption 1) and convergent (assumption 2), for both estimators* (32) *and* (33)*, the bounds on the sample approximation* (30) *are monotonic as a function of*

the number of resimplifications and convergent after at most $n_{\max} \cdot M$ *resimplifications for* **any** *resimplification strategy. Here* M *is the number of posterior beliefs in* (32) *or* (33) *. Namely, if all the rewards are at the maximal simplfciation level* $n_{\max}$ *we have to reach*

$$\underline{\hat{Q}}(\cdot) = \hat{Q}(\cdot) = \overline{\hat{Q}}(\cdot). \tag{37}$$

Similarly for Optimal value function the equality $\underline{\hat{V}}(\cdot) = \hat{V}(\cdot) = \overline{\hat{V}}(\cdot)$ *holds.*

The reader can find the proof in the Appendix 11.1. Theorem 1 ensures that if the resimplification strategy is valid (Definition 1), we do not get stuck in an infinite loop of resimplifications if instead of $\hat{Q}(\cdot)$ we use its bounds. In particular, if (37) is reached, there is no reason to activate the resimplification routine.

Importantly, as we discuss next and corroborate by simulations in many cases we can identify the optimal action before reaching the maximal number of resimplificaitons.

3.4 Adaptive Simplification Mechanics

Our adaptive simplification approach is based on two key observations. The *first key observation* is that we can compare bounds over (30) constituted by rewards at different levels of simplification. Our *second key observation* is that we can reuse calculations between different simplification levels avoiding recalculation of the simplification from scratch.

Naturally we do not want to reach (37). Let us begin by explaining how we determine an optimal action by using bounds over the action-value function instead of its explicit calculation and obtain a significant speedup in planning time. If there is no overlap between the intervals originated from the upper and lower bounds (30) of each candidate action, we can determine the optimal action and therefore there is no reason to call the resimplification routine.

Contemplate about some belief b_ℓ in the belief tree. We annotate by superscript j candidate actions emanating from b_ℓ, such that the index j corresponds to the jth candidate action. We first select a candidate action using the lower bound (30) over $\hat{Q}\big(b_\ell, a_\ell^j\big)$ as

$$j^\dagger(b_\ell(h_\ell)) = \arg\max_j \Big\{\underline{\hat{Q}}(b_\ell(h_\ell), \{a_\ell^j, \pi^*_{(\ell+1)+}\}) + c^j(h_\ell a^j)\Big\}, \tag{38}$$

where c^j is an action dependent constant. In case of a given belief tree $c^j = 0 \ \forall j$, whereas in case of MCTS, it is a constant originated from UCB as in (16).

We then ask the question whether or not an overlap with another candidate action exists,

$$\begin{aligned} &\underline{\hat{Q}}(b_\ell, \{a_\ell^{j^\dagger}, \pi^*_{(\ell+1)+}\}) + c^{j^\dagger} \overbrace{\geq}^{?} \\ &\geq \max_{j \in \{1\dots\}\setminus\{j^\dagger\}} \Big\{\overline{\hat{Q}}(b_\ell, \{a_\ell^j, \pi^*_{(\ell+1)+}\}) + c^j\Big\} \end{aligned} \tag{39}$$

See a visualization in Fig. 3a.

If the condition displayed by equation (39) is not fulfilled, as depicted in Fig. 3a, we shall tighten the bounds (30) by calling a **resimplification strategy** . Importantly, in case of a given belief tree, even if an overlap is present similar to branch-and-bound technique (Kochenderfer et al. 2022) we can prune any subtree obtained with action j satisfying

$$\underline{\hat{Q}}(b_\ell, \{a_\ell^{j^\dagger}, \pi^*_{(\ell+1)+}\}) + c^{j^\dagger} \geq \overline{\hat{Q}}(b_\ell, \{a_\ell^j, \pi^*_{(\ell+1)+}\}) + c^j. \tag{40}$$

We illustrated this aspect in Fig. 3b. If the belief tree is constructed gradually as in MCTS based methods and anytime setting, instead of pruning, we still can use (40) to dismiss suboptimal, at current simulation of MCTS, actions (See Fig. 4).

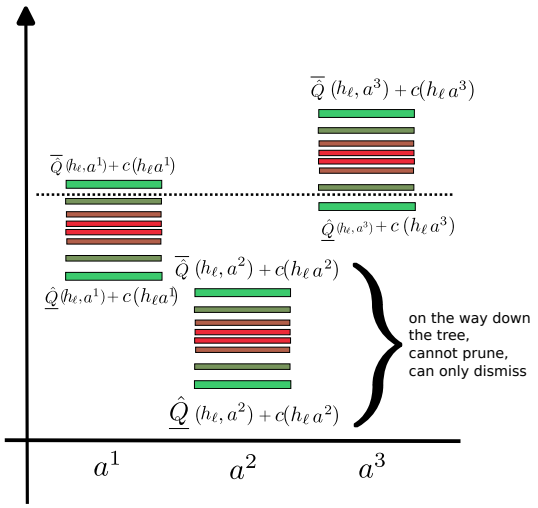


Figure 4. Demonstration of our approach in the setting of MCTS. In contrast to Fig. 3b, we cannot prune action a^2 and can only dismiss it to not participate in resimplifications. This is because, in the next tree queries, a^2 may be the best action for the robot to take.

Once no overlap is present (the condition (39) is fulfilled) we can declare that the selected action is optimal ($\pi^*_\ell(b_\ell) = a_\ell^{j^\dagger(b_\ell)}$). Utilizing the optimal action we can bound the *optimal* value function $\hat{V}(b_\ell, \pi^*_{\ell+})$ as such

$$\overline{\hat{V}}(b_\ell, \{\pi^*_\ell, \pi^*_{(\ell+1)+}\}) \triangleq \overline{\hat{Q}}(b_\ell, \{a_\ell^{j^\dagger(b_\ell)}, \pi^*_{(\ell+1)+}\}), \tag{41}$$

$$\underline{\hat{V}}(b_\ell, \{\pi^*_\ell, \pi^*_{(\ell+1)+}\})\}) \triangleq \underline{\hat{Q}}(b_\ell, \{a_\ell^{j^\dagger(b_\ell)}, \pi^*_{(\ell+1)+}\}). \tag{42}$$

Let us recite that the bounds (41) and (42) are conditioned on the fact that there is no overlap of the bounds intervals that correspond to different candidate actions, namely the condition (39) is met for *each* belief b_ℓ in the belief tree. This situation is visualized in Fig. 3b.

On the other hand, to identify the optimal immediate action a_k^*, we require no overlap between bounds of different actions only at the root of the belief tree (where the belief is b_k). This means that at each belief node b_ℓ in the tree, besides the root, we only want to bound the value function for the optimal action (and under optimal future policy). While it is possible to do so by first determining the optimal action, as in (41) and (42), we can bypass this step and directly bound the value function over the optimal action as follows,

$$\overline{\hat{V}}(b_\ell, \{\pi^*_\ell, \pi^*_{(\ell+1)+}\}) \triangleq \max_j \overline{\hat{Q}}(b_\ell, \{a^j, \pi^*_{(\ell+1)+}\}), \tag{43}$$

$$\underline{\hat{V}}(b_\ell, \{\pi^*_\ell, \pi^*_{(\ell+1)+}\})\}) \triangleq \max_j \underline{\hat{Q}}(b_\ell, \{a^j, \pi^*_{(\ell+1)+}\}), \tag{44}$$

i.e. relaxing the requirement of no overlap between bounds for different actions at any node b_ℓ besides b_k. See illustration of (43) and (44) in Fig. 3c. In turn, eliminating a single overlap at the root results in lower rewards simplification levels in the tree, although such a value bounds may be looser. As we shall see, this approach would typically yield more speedup.

Nevertheless, when we need a policy tree we still have to obtain an optimal action at each belief node within the tree. This requires no bounds overlap at each node, as in the former setting. This situation arises for example when the action and observation spaces are large but discrete. In this case the robot sometimes does not do re-planning at each time step. Instead the robot uses the policy tree as a representation of the policy and selects an optimal action that corresponds to the received observation. In addition, such a strategy accommodates possible reuse calculations in such a solved belief tree (Farhi and Indelman 2019, 2021).

To conclude this section let us summarize. As discussed, we have the following two variants:

- The resimplification is initiated at each nonterminal posterior belief node b_ℓ up until no overlap between candidate actions is present and the optimal action $\pi^*_\ell(b_\ell)$ is selected. This way we bound the optimal value function of the descendant to b_k nodes using an optimal action according to (41) and (42). We named this approach Policy Tree (`PT`).
- The resimplification is commenced solely at the root b_k of the whole belief tree. We eliminate the overlap and obtain an optimal action only at b_k. This way we use (43) and (44) to bound the optimal value function of the descendant to b_k nodes. We shall refer to this variant of our approach as `LAZY`.

3.5 Specific Resimplification Strategies

In this paper we consider two specific resimplification strategies that are elaborated in the next sections: `Simplification Level` (`SL`) and `Gap`. We note that additional valid resimplification strategies exist and can be plugged-in into the above-proposed general theory.

Simplification level: The resimplification strategy can be directly tied to the simplification level. In this situation the resimplifcation strategy promotes simplification level of the rewards inside the belief tree corresponding to bounds in (29) or (30) based on the simplification level itself. We provide further details in the setting of a given belief tree, considering a `PT` variant in Section 4.

Gap: Another possibility is that the resimplification is tied to the gap $\overline{\rho}^s - \underline{\rho}^s$. Such a resimplification promotes the simplification level if the reward bounds gap satisfies a certain condition. We describe thoroughly this resimplification flavor in the setting of a given belief tree, considering `LAZY` variant in Section 4.2, and in MCTS setting, considering a `PT` variant in Section 5.4.

Each of these strategies can be used in conjunction with any of the variants `PT` and `LAZY`. In the sequel, we shall denote these combinations explicitly, e.g. `PT-SL`, `LAZY-Gap` and `PT-Gap`.

The preceding discussion raises the question of how do we actually incorporate the proposed bounds into online decision making. This brings us to the next section. We first consider a given belief tree and then coupled belief tree construction and solution as in MCTS methods. It shall be noted that further presented resimplification strategies are also suitable for static candidate action sequences, with minor modifications.

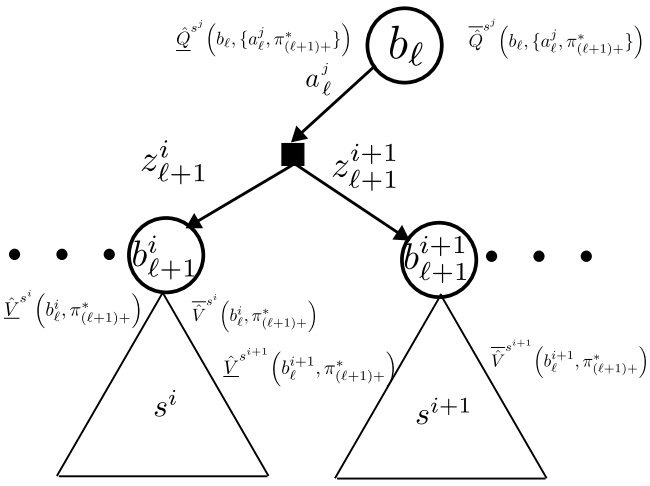


Figure 5. Pruning the subtrees by adaptively promoting the simplification levels of the rewards inside. Here the simplification levels of a subtrees are not equal. It is possible that $s^i \neq s^{i+1}$. Note that here the superscripts are relative to b_ℓ as opposed to Fig. 1 and Fig. 6.

4 Adaptive Simplification in the Setting of a Given Belief Tree

We start with the assumption that the belief tree was generated in some way and that it is given, e.g, Sparse Sampling (SS) algorithm introduced by Kearns et al. (2002). In other words the belief tree construction is not coupled with rewards calculation and estimation of the objective.

In this setting, we contribute two resimplification strategies. The first strategy is described in Section 4.1. The general idea is to break down recursively a given belief tree $\mathbb{T}$ into its sub-problems (subtrees), denoted as $\{\mathbb{T}^j\}_{j=1}^{|\mathcal{A}|}$ (each subtree j at the root belief has a single action j), and solve each sub-problem with its own simplification level of the corresponding belief subtree. Ultimately this would lead to the solution of the entire problem via action-value function bounds (32). This strategy is based on `Simplification Level` and it is a `PT` strategy. The action-value bounds should not overlap **at each node** in the given belief tree.

The second strategy is described in Section 4.2. This resimplification strategy is based on `Gap` and it is a `LAZY` strategy. Here, the general idea is to first substitute all the rewards in a given belief tree by bounds with the coarsest simplification level. We then eliminate an overlap between candidate actions only at the root belief node b_k by a repetitive descending to the belief tree, promoting the simplification levels along a single lace chosen according to largest gap and ascending back. We emphasize that in this setting, the action-value bounds should not overlap **only at the root node** in the given belief tree.

As mentioned in the beginning of Section 2.3.1, only for simplicity we consider a symmetric setting in terms of

sampled actions and the observations, but the approach is applicable without any limitations to any given belief tree.

4.1 *Resimplification strategy:* `PT-SL`

This section presents our first resimplification strategy. We now turn to thorough description.

Not to be confused with **policy tree** represented by the (14) or (15) the **given belief tree** ($\mathbb{T}$) has more than a single action emanating from each belief node besides the leafs.

We now assign a simplification level to the bounds on the value and action value functions. Consider again some belief node b_ℓ in the belief tree, and assume recursively for *each* of its children belief nodes $b_{\ell+1}$ we already calculated the optimal policy $\pi^*_{(\ell+1)+}(b_{\ell+1})$ and the corresponding upper and lower bounds $\underline{\hat{V}}^s\left(b_{\ell+1}, \pi^*_{(\ell+1)+}\right)$ and $\overline{\hat{V}}^s\left(b_{\ell+1}, \pi^*_{(\ell+1)+}\right)$. In general, these bounds for each child sub-policy tree of b_ℓ can correspond to different simplification levels.

From now on let the superscript s over the action-value function bounds from (32) and (31) denote the simplification level stemmed from pertaining reward bounds. The bounds previously described by Eqs. (32) for belief node b_ℓ, incorporating simplification level, are now modified to

$$\begin{aligned}
\overline{\hat{Q}}^{s^j}(b_\ell, \{a^j_\ell, \pi^*_{(\ell+1)+}\}) &= \frac{1}{n_z}\sum_{i=1}^{n_z}\overline{\rho}^s(b_\ell, a^j_\ell, b^i_{\ell+1}) + \\
&+\gamma\frac{1}{n_z}\sum_{i=1}^{n_z}\overline{\hat{V}}^{s^i}(b^i_{\ell+1}, \pi^*_{(\ell+1)+}) \\
\underline{\hat{Q}}^{s^j}(b_\ell, \{a^j_\ell, \pi^*_{(\ell+1)+}\}) &= \frac{1}{n_z}\sum_{i=1}^{n_z}\underline{\rho}^s(b_\ell, a^j_\ell, b^i_{\ell+1}) + \\
&+\gamma\frac{1}{n_z}\sum_{i=1}^{n_z}\underline{\hat{V}}^{s^i}(b^i_{\ell+1}, \pi^*_{(\ell+1)+}),
\end{aligned} \tag{45}$$

as illustrated in Fig. 5. We shall pinpoint the abuse of notation here. In contrast to (32) the superscript s over the immediate reward bounds denotes a specific simplification level instead of indicating a general simplification.

Note equation (45) applies for each $a^j_\ell \in \mathcal{A}$, and as mentioned, each belief node $b^i_{\ell+1}$ (one for each observation $z^i_{\ell+1}$) has, in general, its own simplification level s^i. In other words, for each $b^i_{\ell+1}$, s^i is the simplification level that was sufficient for calculating the bounds $\left\{\overline{\hat{V}}^{s^i}(b^i_{\ell+1}, \pi^*_{(\ell+1)+}), \underline{\hat{V}}^{s^i}(b^i_{\ell+1}, \pi^*_{(\ell+1)+})\right\}$ and the corresponding optimal policy $\pi^*_{(\ell+1)+}$. Thus, when addressing belief node b_ℓ in (45), for each belief node $b^i_{\ell+1}$ and its corresponding simplification level s^i, these bounds are already available.

Further, as seen in (45), the immediate reward and the corresponding bounds $\overline{\rho}$ and $\underline{\rho}$, in general, can be calculated with their own simplification level s. In particular, when starting calculations, s could correspond to a default coarse simplification level, e.g. coarsest level $s = 1$. Another possibility is to set $s = s^i$ for corresponding simplification level of value function bounds of the i-th child belief.

To define simplification level s^j of the bounds (45) we leverage the recursive nature of the Bellman update and

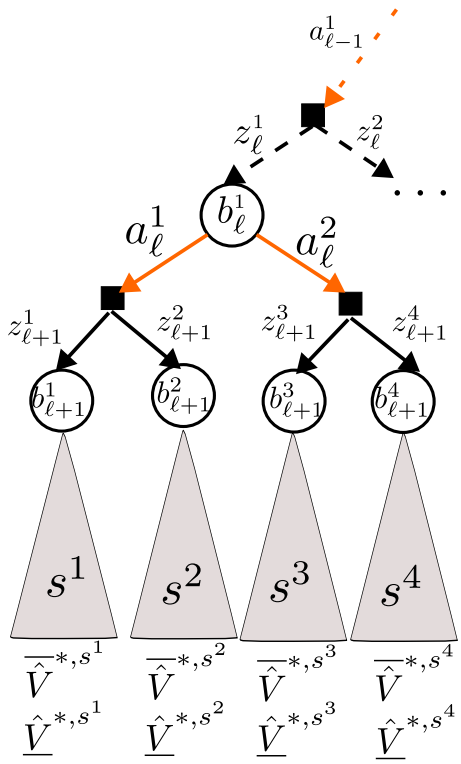


Figure 6. An example of the simplification paradigm. The superscript here denotes **global** number of the belief, observation or action in the belief tree as opposed to equation (45) and Fig 5.

define

$$s^j \triangleq \min\{\underbrace{s}_{\substack{\overline{\rho}^s \\ \underline{\rho}^s}}, \underbrace{s^{i=1}, s^{i=2} \dots s^{i=n_z}}_{\substack{\overline{\hat{V}}^{s^i}(b^i_{\ell+1}, \pi^*_{(\ell+1)+}) \\ \underline{\hat{V}}^{s^i}(b^i_{\ell+1}, \pi^*_{(\ell+1)+})}}\}, \tag{46}$$

where $\{s^{i=1}, s^{i=2}, \dots, s^{i=n_z}\}$ represent the (generally different) simplification levels of optimal value functions of belief nodes $b^i_{\ell+1}$ considered in the expectation approximation in (45).

We now wish to decide which action $a^{j^\dagger(b_\ell)}_\ell \in \mathcal{A}$ is optimal from belief node b_ℓ; the corresponding optimal policy would then be $\pi^*_{\ell+} = \{a^*_\ell, \pi^*_{(\ell+1)+}\}$, where $\pi^*_{(\ell+1)+}$ is the already-calculated optimal policy for belief nodes $\{b^i_{\ell+1}\}_{i=1}^{n_z}$ that a^*_ℓ leads to. See illustration in Fig. 5.

Let us utilize now a general simplification approach described in section 3.4. Overall in each belief node we have n_a candidate actions indexed by superscript j in (45).

At each belief node we first select an optimal action candidate according to (38) with a nullified action dependent constant ($\forall j \; c^j = 0$). Further, in any `PT` resimplification strategy there are three possible scenarios.

- No overlap is present ((39) is satisfied) and we are at the root i.e. $b_\ell = b_k$. In this case the optimal action shall be returned.
- No overlap is present ((39) is satisfied) and we not at the root b_k. In this case, using the optimal action we bound optimal value function using the (41) and (42).
- Eq. (39) is not satisfied, meaning an overlap is present. In the presence of overlap we shall prune actions

according to (40) and commence resimplification routine based on resimplification strategy.

We now discuss how the simplification level is updated recursively from the simplification level of pertaining reward bounds, and revisit the process to calculate the optimal policy and the corresponding bounds. For some belief node b_ℓ in the belief tree, consider the bounds $\overline{\hat{Q}}^{s^j}(b_\ell, \{a_\ell^j, \pi^*_{(\ell+1)+}\})$ and $\underline{\hat{Q}}^{s^j}(b_\ell, \{a_\ell^j, \pi^*_{(\ell+1)+}\})$ from (45) for different actions $a_\ell^j \in \mathcal{A}$, that partially overlap and therefore could not be pruned. Each policy tree corresponding to action a_ℓ^j can generally have its own simplification level s^j. We now iteratively increase the simplification level by 1. This can be done for each of the branches, if s^j is identical for all branches, or only for the branch with the coarsest simplification level.

Consider now any such branch whose simplification level needs to be adapted from s^j to s^j+1. Recall, that at this point, the mentioned bounds were already calculated, thus their ingredients, in terms of $\{\overline{\rho}^s(b_\ell, a_\ell^j, b_{\ell+1}^i), \underline{\rho}^s(b_\ell, a_\ell^j, b_{\ell+1}^i)\}_{i=1}^{n_z}$ and $\{\overline{\hat{V}}^{s^i}(b_{\ell+1}^i, \pi^*_{(\ell+1)+}), \underline{\hat{V}}^{s^i}(b_{\ell+1}^i, \pi^*_{(\ell+1)+})\}_{i=1}^{n_z}$, involved in approximating the expectation in (45), are available. Recall also (46), i.e. each element in $\{s, s^{i=1}, s^{i=2}, \dots, s^{i=n_z}\}$ is either equal or larger than s^j. We now discuss both cases, starting from the latter.

As we assumed bounds to improve monotonically as simplification level increases, see Assump. 1, for any $s^i > s^j + 1$ we already have readily available bounds $\overline{\hat{V}}^{s^i}(b_{\ell+1}^i, \pi^*_{(\ell+1)+}), \underline{\hat{V}}^{s^i}(b_{\ell+1}^i, \pi^*_{(\ell+1)+})$ which are tighter than those that would be obtained for simplification level s^j+1. Thus, we can *safely skip* the calculation of the latter and use the existing bounds from level s^i as is.

For the former case, i.e. $s^i = s^j$, we now have to adapt the simplification level of a child tree i to s^j+1 by calculating the bounds $\overline{\hat{V}}^{s^i+1}(b_{\ell+1}^i, \pi^*_{(\ell+1)+}), \underline{\hat{V}}^{s^i+1}(b_{\ell+1}^i, \pi^*_{(\ell+1)+})$. Here, our *key insight* is that, instead of calculating these bounds from scratch, we can re-use calculations between different simplification levels, in this case, from level s^i. As the bounds from that level are available, we can identify only the incremental part that is "missing" to get from simplification level s^i to s^i+1, and update the existing bounds $\overline{\hat{V}}^{s^i}(b_{\ell+1}^i, \pi^*_{(\ell+1)+}), \underline{\hat{V}}^{s^i}(b_{\ell+1}^i, \pi^*_{(\ell+1)+})$ to recover $\overline{\hat{V}}^{s^i+1}(b_{\ell+1}^i, \pi^*_{(\ell+1)+}), \underline{\hat{V}}^{s^i+1}(b_{\ell+1}^i, \pi^*_{(\ell+1)+})$ exactly. The same argument applies also for bounds over momentary rewards. In Section 6.2.3 we apply this approach to a specific simplification and reward function.

We can repeat iteratively the above process of increasing the simplification level until we can prune all branches but one. This means each subtree will be solved maximum once, per simplification level. Since we assumed the reward bounds converge monotonically to the original reward for the finest level $s = n_{\max}$ (See Fig. 2), from Theorem 1, we are guaranteed to eventually disqualify all sub-optimal branches. Our described approach is summarized in Algs. 1 and 2.

4.1.1 *Illustrative Example*

We now illustrate the described above resimplification strategy in a toy example. Before we start this section, let us clarify that in the example the superscripts are global over the belief tree in contrast to previous section. Consider Fig. 6 and assume the subtrees to b_ℓ^1 were solved using simplification levels that hold $s^2 = s^1+1, s^2 < s^3, s^4$. Further assume the immediate reward simplification is $s = s^1$. According to definitions above this means that for subtree starting at b_ℓ^1 and action a_ℓ^1 the simplification level is $\min\{s^1, s^2\}$ and for action a_ℓ^2 the simplification level is $\min\{s^3, s^4\}$. Now, we consider the case the existing bounds of the subtrees were not tight enough to prune, we adapt simplification level starting from b_ℓ^1 and promote $s \leftarrow s^1+1$. Since $s^1 < s^1+1$ we re-simplify the subtree corresponding to simplification level of s^1 to simplification level s^1+1, i.e. to a finer simplification.

However we do not need to re-simplify subtrees corresponding to s^2, s^3, s^4: The tree corresponding to s^2 is already simplified to the currently desired level; thus we can use its existing bounds. For the two other trees, their current simplification levels, s^3 and s^4, are higher (finer) than the desired s^1+1 level, and since the bounds are tighter as simplification level increases we can use their existing tighter bounds without the need to "go-back" to a coarser level of simplification. If we can now prune one of the actions, we keep pruning up the tree. If pruning is still not possible, we need to adapt simplification again with simplification level s^1+2.

4.1.2 *A Detailed Algorithm Description*

Let us thoroughly describe Alg. 1. We are given a belief tree $\mathbb{T}$. First at the line 10 Alg. 1 recursively descend to the leafs. When the line 11 is hit for the first time the corresponding rewards are set to the initial simplification level or also possible that minimal level of child optimal value bounds is used. In our simulations we used minimal reward level. Further the algorithm calculates bounds over action-value function represented by (45). This happens in line 15 of Alg. 1. The next step is to try to prune all subtrees but one utilizing the Alg. 1. Note, at this point all the subtrees $\mathbb{T}^j$ are already policy trees, namely only a single action emanating from each posterior belief. In there is more that single action left after pruning, at the line 20 the Algorithm 1 calls routine `ResimplifyTree` to initiate **resimplification** for selected subtree corresponding to action a^j. The simplification level of a single step ahead reward is always have to be promoted as we do in line 27. Further, Alg. 1 treats similarly subtrees, if they are present.

4.2 *Resimplification strategy:* `LAZY-Gap`

The `PT` resimplification strategy from previous section assure that no overlap is present (Fig. 3b) at each non-leaf posterior belief and we know the optimal action to take. However, it can inflict a redundant computational burden. We can handle the overlap only at the root of the belief tree and use the bounds over optimal value function according to (43) and (44). Since we already presented the resimplification strategy based on the simplification levels, our second resimplification strategy will be based on the distance between reward bounds. However, the bounds (43) and (44) can be utilized directly also with the resimplifcation strategy based on simplification levels. Yet, this is out of the scope of this paper.

In this section we present a lazy variant of the resimplifcation strategy. In a `LAZY` variant, the overlap is

Algorithm 1 Simplified Information Theoretic Belief Space Planning (SITH-BSP)

procedure SOLVEBELIEFTREE(belief-tree: $\mathbb{T}$)
  if $\mathbb{T}$ is a leaf **then**
    //Corresponds to a single belief node.
    return $0, 0$
  end if
  for all subtrees $\mathbb{T}^j \in \{\mathbb{T}^j\}_{j=1}^{|\mathcal{A}|}$ **do** // Actions
    //Observations
    for all subtrees $\mathbb{T}^{j,i} \in \{\mathbb{T}^{j,i}\}_{i=1}^{n_z}$ **do**
      //Returns Optimal Value bounds and prune suboptimal branches of $\mathbb{T}^{j,i}$.
      SOLVEBELIEFTREE($\mathbb{T}^{j,i}$)
      Set the simplification level of $\underline{\rho}^s(b, a^j, b'^i)$ and $\overline{\rho}^s(b, a^j, b'^i)$ as in (46)
    end for
    Calculate $\underline{\hat{Q}}^{s^j}, \overline{\hat{Q}}^{s^j}$ according to (45)
  end for
  PRUNE($\{\underline{\hat{Q}}^{s^j}, \overline{\hat{Q}}^{s^j}\}_{j=1}^{|\mathcal{A}|}$) // Alg. 2
  while not all subtrees $\mathbb{T}^j \in \{\mathbb{T}^j\}_{j=1}^{|\mathcal{A}|}$ but 1 pruned **do**
    Find minimal simplification level $s_{\min}$ between all $\underline{\hat{Q}}^{s^j}, \overline{\hat{Q}}^{s^j}$ corresponding to **not** pruned $\mathbb{T}^j$
    // Can be more than single subtree
    select subtree $s^j == s_{\min}$
    RESIMPLIFYTREE($\mathbb{T}^j$)
    PRUNE($\{\underline{\hat{Q}}^{s^j}, \overline{\hat{Q}}^{s^j}\}_{j=1}^{|\mathcal{A}|}$) // Alg. 2
  end while
  return optimal action branch that left a^* and $\underline{\hat{Q}}^{s^j}, \overline{\hat{Q}}^{s^j}$.
end procedure
procedure RESIMPLIFYTREE($\mathbb{T}^j$)
  for all subtrees $\mathbb{T}^{j,i} \in \{\mathbb{T}^{j,i}\}_{i=1}^{n_z}$ **do**
    RESIMPLIFYREWARD($\mathbb{T}^j$, b, a^j, b^i) // Alg. 3
    if b^i has children **then**
      // s^i is a simplification level of corresponding optimal value function (policy tree)
      if $s^i \leq s_{\min}$ **then**
        // Alg. 4
        RESIMPLIFYSUBTREE($\mathbb{T}^{j,i}$, b, b^i)
        $s^i \leftarrow s^i + 1$
      end if
    end if
  end for
  $s^j \leftarrow s^j + 1$
end procedure

Algorithm 2 Pruning of trees

procedure PRUNE
  Input: (belief-tree root, b; bounds of root's children, $\{\underline{\hat{Q}}^j, \overline{\hat{Q}}^j\}_{j=1}^{n_a}$) // n_a is the number of child branches (candidate actions) going out of b.
  $\underline{\hat{Q}}^* \leftarrow \max_j \{\underline{\hat{Q}}^j\}_{j=1}^{n_a}$
  for $j \in 1 : n_a$ **do**
    if $\underline{\hat{Q}}^* > \overline{\hat{Q}}^j$ **then**
      prune child j from the belief tree
    end if
  end for
end procedure

Algorithm 3 ResimplifyReward

procedure RESIMPLIFYREWAD($\mathbb{T}^j$, b, a, b')
  Obtain corresponding to the $\mathbb{T}^j$ bounds $\overline{\hat{V}}$, $\underline{\hat{V}}$
  $\overline{\hat{V}} \leftarrow \overline{\hat{V}} - \frac{\overline{\rho}^s(bab')}{n_z}$
  $\underline{\hat{V}} \leftarrow \underline{\hat{V}} - \frac{\underline{\rho}^s(bab')}{n_z}$
  Advance level of simplification of b'
  $\overline{\hat{V}} \leftarrow \overline{\hat{V}} + \frac{\overline{\rho}^s(bab')}{n_z}$
  $\underline{\hat{V}} \leftarrow \underline{\hat{V}} + \frac{\underline{\rho}^s(bab')}{n_z}$
end procedure

Algorithm 4 ResimplifySubtree

procedure RESIMPLIFYSUBTREE($\mathbb{T}^{j,i}$,b b')
  $\overline{\hat{V}}(b) \leftarrow \overline{\hat{V}}(b) - \gamma\frac{\overline{\hat{V}}(b')}{n_z}$
  $\underline{\hat{V}}(b) \leftarrow \underline{\hat{V}}(b) - \gamma\frac{\underline{\hat{V}}(b')}{n_z}$
  RESIMPLIFYTREE($\mathbb{T}^{j,i}$)
  $\overline{\hat{V}} \leftarrow \overline{\hat{V}}(b) + \gamma\frac{\overline{\hat{V}}(b')}{n_z}$
  $\underline{\hat{V}} \leftarrow \underline{\hat{V}}(b) + \gamma\frac{\underline{\hat{V}}(b')}{n_z}$
end procedure

checked solely at the root b_k of the whole belief tree. In this approach three scenarios can be encountered at each belief node.

- The belief node is not root. We bound optimal value according to (43) and (44).
- At the root b_k we shall check for overlap. If no overlap is present ((39) is satisfied) we prune all suboptimal actions according to Alg. 2 and return an optimal action as described in Section 3.4.
- In the presence of an overlap at the root b_k (Eq. (39) is not satisfied), we shall prune actions according to (40) and Alg. 2 and commence a resimplification routine for the non pruned actions based on the resimplification strategy.

Having presented general steps of any LAZY variant of resimplification strategy, we are ready to delve into specific gap driven resimplfication strategy. Let us introduce the following notation

$$G(ha) \triangleq \overline{\hat{Q}}(ha) - \underline{\hat{Q}}(ha). \tag{47}$$

We remind the reader that sometimes, for simplicity of explanation, we will make the gap dependent on belief and an action, and denote $G(ba)$. We use this gap to steer the resimplification procedure towards more promising lace. The lace with actions inducing largest gap (47) at each belief action node along the lace will be selected to resimplification. In fact we use similar gap for value function to select observations along the lace. Now let us proceed to the detailed algorithm description.

Algorithm 5 Lazy Simplified Information Theoretic Belief Space Planning (LAZY-BSP)

1: **procedure** PLAN(belief: b, belief-tree: $\mathbb{T}$)
2: BOUNDOPTIMALVALUE(belief: b, belief-tree: $\mathbb{T}$)
3: $a^* \leftarrow$ ACTIONSELECTION(b, L) // Alg. 6
4: **return** a^*
5: **end procedure**
6: **procedure** BOUNDOPTIMALVALUE(belief-tree: $\mathbb{T}$)
7: **if** $\mathbb{T}$ is a leaf **then**
8: //Corresponds to a single belief node.
9: **return** $0, 0$
10: **end if**
11: **for all** subtrees $\mathbb{T}^j \in \{\mathbb{T}^j\}_{j=1}^{|\mathcal{A}|}$ **do**
12: **for all** subtrees $\mathbb{T}^{j,i} \in \{\mathbb{T}^{j,i}\}_{i=1}^{n_z}$ **do**
13: $\underline{\hat{V}}(b'), \overline{\hat{V}}(b') \leftarrow$ BOUNDOPTIMALVALUE(b, $\mathbb{T}^{j,i}$)
14: Set the simplification level of $\underline{\rho}^s(b, a^j, b'^i)$ and $\overline{\rho}^s(b, a^j, b'^i)$ to coarsest possible
15: **end for**
16: Calculate $\underline{\hat{Q}}^j, \overline{\hat{Q}}^j$
17: **end for**
18: $\underline{\hat{V}}(b) \leftarrow \max_j \{\underline{\hat{Q}}^j\}$
19: $\overline{\hat{V}}(b) \leftarrow \max_j \{\overline{\hat{Q}}^j\}$
20: **return** $\underline{\hat{V}}(b), \overline{\hat{V}}(b)$
21: **end procedure**

4.2.1 A Detailed Algorithm Description This approach is summarized in Alg. 5 When we apply this resimplification strategy, we first use the lowest simplification level for each pair of consecutive beliefs in the given belief tree. In other words, the Alg. 5 first descends to the leaves of the given belief tree. Then it bounds each optimal value function using the initial simplification level using (43) and (44). This initial passage over the given belief tree is enclosed by routine `BoundOptimalValue`. In the procedure `ActionSelection` we increase the simplification level of the reward bounds in the given tree until there is no overlap at the **root**, as in Fig. 3b. In this way, we can prune entire given subtrees at the root, corresponding to candidate actions. The procedure `LazyResimplify` descends back to some leaf through the lace with largest gaps on the way. It select action in line 15. It then select observation/belief according to largest gap of a single step ahead rewards if these rewards are leafs (line 17) or the largest gap of the optimal value function bounds (line 19).

Algorithm 6 Action Selection for Lazy Simplified Information Theoretic Belief Space Planning

1: **procedure** ACTIONSELECTION(belief: b, horizon: L)
2: PRUNE($\{\underline{\hat{Q}}^j, \overline{\hat{Q}}^j\}_{j=1}^{|\mathcal{A}|}$) // Alg. 2
3: $a^\dagger \leftarrow \arg\max_a \underline{\hat{Q}}\left(b, \{a, \pi^*_{(k+1+)}\}\right)$
4: $\tilde{a} \leftarrow \arg\max_{a \in \mathcal{A} \setminus a^\dagger} \overline{\hat{Q}}\left(b, \{a, \pi^*_{(k+1+)}\}\right)$
5: $\Delta \leftarrow \left(\overline{\hat{Q}}(b, \{\tilde{a}, \pi^*_{(k+1+)}\}) - \underline{\hat{Q}}(b, \{a^\dagger, \pi^*_{(k+1+)}\})\right)^+$
6: **while** $\Delta > 0$ **do**
7: $a^* \leftarrow$ LAZYRESIMPLIFY(b, L)
8: **end while**
9: **return** a^*
10: **end procedure**
11: **procedure** LAZYRESIMPLIFY(belief: $b(h)$, depth: d)
12: **if** b is leaf **then**
13: **return** $0, 0$
14: **end if**
15: $\tilde{a} \leftarrow \arg\max_{a \in C(h)} \overline{\hat{Q}}(b, \{a, \pi^*_{(k+1+)}\}) - \underline{\hat{Q}}(b, \{a, \pi^*_{(k+1+)}\})$ // Gap as in (47)
16: **if** $d == 1$ **then**
17: $b' \leftarrow \arg\max_{b' \in C(h\tilde{a})} \overline{\rho}^s(b\tilde{a}b') - \underline{\rho}^s(b\tilde{a}b')$
18: **else**
19: $b' \leftarrow \arg\max_{b' \in C(h\tilde{a})} \overline{\hat{V}}(b') - \underline{\hat{V}}(b')$
20: **end if**
21: RESIMPLIFYREWARD($\mathbb{T}$, b, $\tilde{a}$, b') // Alg. 3
22: $\overline{\hat{Q}}(b, \{\tilde{a}, \pi^*_{(k+1+)}\} \leftarrow \overline{\hat{Q}}(b, \{\tilde{a}, \pi^*_{(k+1+)}\} - \gamma \overline{\hat{V}}(b')$
23: $\underline{\hat{Q}}(b, \{\tilde{a}, \pi^*_{(k+1+)}\} \leftarrow \underline{\hat{Q}}(b, \{\tilde{a}, \pi^*_{(k+1+)}\} - \gamma \underline{\hat{V}}(b')$
24: $\underline{\hat{V}}(b'), \overline{\hat{V}}(b') \leftarrow$ LAZYRESIMPLIFY(b', $d-1$)
25: $\overline{\hat{Q}}(b, \{\tilde{a}, \pi^*_{(k+1+)}\} \leftarrow \overline{\hat{Q}}(b, \{\tilde{a}, \pi^*_{(k+1+)}\} + \gamma \overline{\hat{V}}(b')$
26: $\underline{\hat{Q}}(b, \{\tilde{a}, \pi^*_{(k+1+)}\} \leftarrow \underline{\hat{Q}}(b, \{\tilde{a}, \pi^*_{(k+1+)}\} + \gamma \underline{\hat{V}}(b')$
27: $\underline{\hat{V}}(b) \leftarrow \max_a \{\underline{\hat{Q}}(b, \{a, \pi^*_{(k+1+)}\})\}$
28: $\overline{\hat{V}}(b) \leftarrow \max_a \{\overline{\hat{Q}}(b, \{a, \pi^*_{(k+1+)}\})\}$
29: **return** $\underline{\hat{V}}(b), \overline{\hat{V}}(b)$
30: **end procedure**

5 Adaptive Simplification in the Setting of MCTS

In the previous sections, we described the application of the adaptive simplification paradigm when the belief tree is given or its construction is not coupled with the solution. We now turn to an anytime setting where the belief tree is not given. Instead, the belief tree construction is coupled with the estimation of the action-value function (20) at each belief action node. Such an approach is commonly used in Monte Carlo tree search (MCTS) methods based on an exploration strategy, e.g. Upper Confidence Bound (UCB) as in (16). Our goal is to suggest a resimplifcation strategy so that exactly the same belief tree as without simplification would be constructed. Also the same optimal action is identified with and without simplification. To support general belief-dependent rewards we select PFT-DPW as the baseline, as mentioned in Section 1.1.

Common exploration strategies conform to the structure presented in (38). Without loosing generality we focus on the most advanced, to our knowledge, exploration strategy, named UCB and portrayed by (16).

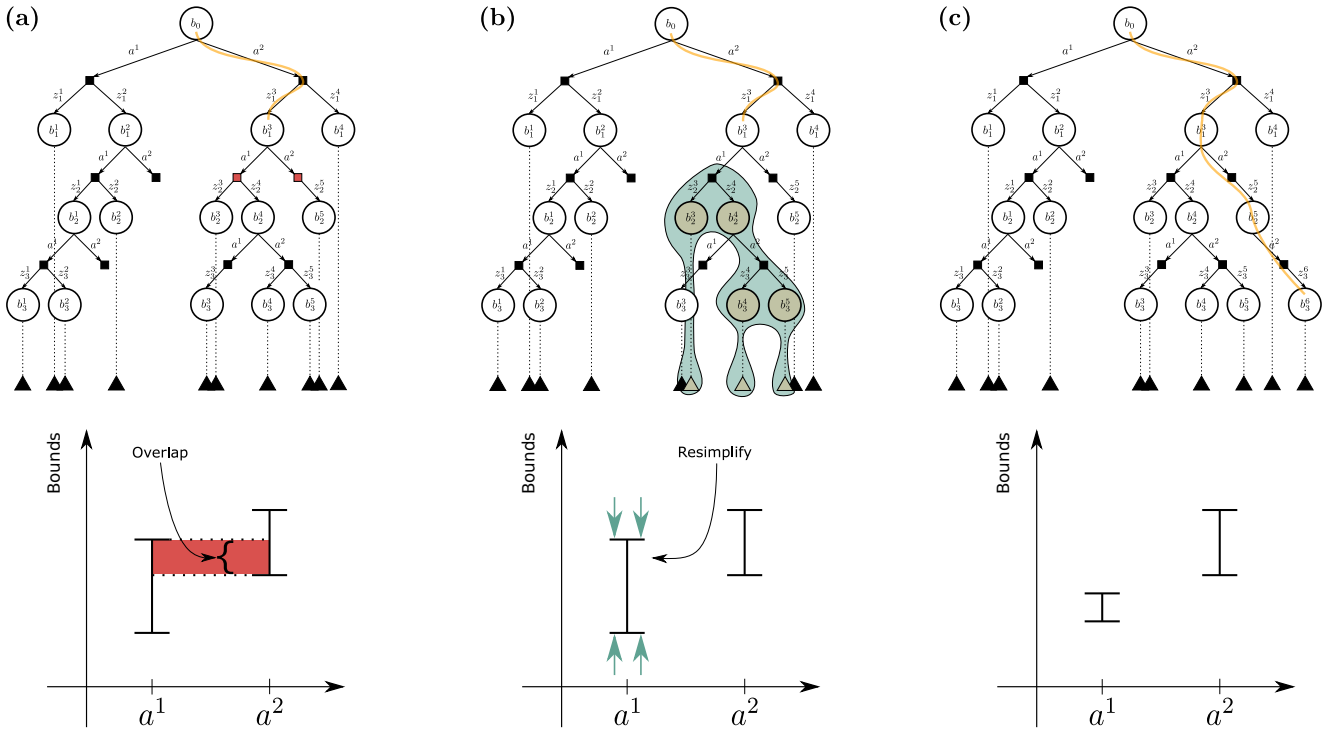


Figure 7. Illustration of our approach. The circles denote the belief nodes, and the rectangles represent the belief-action nodes. Rollouts, emanating from each belief node, are indicated by dashed lines finalized with triangles. **(a)** The simulation starts from the root of the tree, but at node b_1^3 it can not continue due to an overlap of the child nodes (colored red) bounds. **(b)** One of the red colored belief-action nodes is chosen, and resimplification is triggered from it down the tree to the leaves (shaded green area in the tree). The beliefs and rollouts inside the green area (colored by light brown) undergo resimplification if decided so. This procedure results in tighter bounds. **(c)** After the bounds got tighter, nothing prevents the SITH-PFT from continuing down from node b_1^3 guaranteeing the Tree Consistency. If needed, additional resimplifications can be commenced.

5.1 UCB bounds

With this perspicuity in mind, we now introduce bounds over (16)

$$\overline{\mathrm{UCB}}(ha) \triangleq \overline{\hat{Q}}(ha) + c\cdot\sqrt{\log(N(h))/N(ha)}, \tag{48}$$

$$\underline{\mathrm{UCB}}(ha) \triangleq \underline{\hat{Q}}(ha) + c\cdot\sqrt{\log(N(h))/N(ha)}. \tag{49}$$

Similar to the given belief tree setting we now proceed to the explanation how the reward bounds (22) yield (48) and (49).

5.2 Guaranteed Belief Tree Consistency

Since the simplification paradigm substituted UCB (16) by the bounds (48) and (49), the belief tree construction is coupled with these quantities, as opposed to the situation with the given belief tree. If there is an overlap between bounds on UCB for different actions, we can no longer guarantee the same belief tree will be constructed with and without simplification.

In this and the following sections we address this key issue. Specifically, we define the notion of Tree Consistency and prove the equivalence of our algorithm to our baseline PFT-DPW.

Definition 2. Tree consistent algorithms. Imagine two algorithms, constructing a belief tree. Assume every common sampling operation for the two algorithms uses the same seed. The two algorithms are *tree consistent* if two belief trees constructed by the algorithms are identical in terms of actions, observations, and visitation counts.

Our approach relies on a specific procedure for selecting actions within the tree. Since in each simulation the MCTS descends down the tree with a single return lace as in (20), on the way down it requires the action maximizing UCB (16) we shall eliminate overlap at each belief node as described in section 3.4. Further we restate the action selection procedure described in section 3.4 with particular action dependent constant from eq. (38) and (39) rendering the UCB bounds from (48) and (49).

Our action selection is encapsulated by Alg. 8, which is different from the procedure used in PFT-DPW. On top of DPW as in Sunberg and Kochenderfer (2018) with parameters k_a and α_a, instead of selecting an action maximizing the UCB (16), at every belief node we mark as a candidate action the one that maximizes the lower bound $\underline{\mathrm{UCB}}$ as such

$$\tilde{a} = \underset{a\in C(h)}{\arg\max}\ \underline{\mathrm{UCB}}(ha). \tag{50}$$

If $\forall a \neq \tilde{a}, \underline{\mathrm{UCB}}(h\tilde{a}) \geq \overline{\mathrm{UCB}}(ha)$, there is no overlap (Fig. 7 **(c)**) and we can declare that $\tilde{a}$ is identical to a^*, i.e., the action that would be returned by PFT using (16) and the tree consistency has not been affected. Otherwise, the bounds must be tightened, so ensure the tree consistency. We examine the ha siblings of $h\tilde{a}$, which satisfy $a \neq \tilde{a}$: $\underline{\mathrm{UCB}}(h\tilde{a}) < \overline{\mathrm{UCB}}(ha)$ (Fig. 7 **(a)**). Our next step is to tighten the bounds by resimplification (Fig. 7 **(b)**) until there is no overlap using the valid resimplification strategy according to Definition 1.

Algorithm 7 SITH-PFT

1: **procedure** PLAN(belief: b)
2: **for** $i \in 1:n$ or timeout **do**
3: $h \leftarrow \emptyset$
4: SIMULATE(b, $d_{\max}$, h)
5: **end for**
6: **return** ACTIONSELECTION(b, h) // called with nullified exploration constant c
7: **end procedure**
8: **procedure** SIMULATE(belief: b, depth: d, history: h)
9: **if** $d = 0$ **then**
10: **return** 0
11: **end if**
12: $a \leftarrow$ ACTIONSELECTION(b, h)
13: **if** $|C(ha)| \leq k_o N(ha)^{\alpha_o}$ **then**
14: $o \leftarrow$ sample x from b, generate o from (x, a)
15: $b' \leftarrow G_{\mathrm{PF}(m)}(bao)$
16: Calculate initial $\overline{\rho}^s, \underline{\rho}^s$ for b, b' based on $s \leftarrow 1$ // minimal simp. level
17: $C(ha) \leftarrow C(ha) \cup \{(\overline{\rho}^s, \underline{\rho}^s, b', o)\}$
18: $L, U \leftarrow \overline{\rho}^s, \underline{\rho}^s + \gamma$ ROLLOUT(b', hao, $d-1$)
19: **else**
20: $(\overline{\rho}^s, \underline{\rho}^s, b', o) \leftarrow$ sample uniformly from $C(ha)$
21: $L, U \leftarrow \overline{\rho}^s, \underline{\rho}^s + \gamma$ SIMULATE(b', hao, $d-1$)
22: **end if**
23: **if** deepest resimplification depth $< d$ **then** // accounting for updated deeper in the tree bounds. See section 5.3
24: reconstruct $\underline{\hat{Q}}(ha), \overline{\hat{Q}}(ha)$
25: **end if**
26: $N(h) \leftarrow N(h) + 1$
27: $N(ha) \leftarrow N(ha) + 1$
28: $\overline{\hat{Q}}(ha) \leftarrow \overline{\hat{Q}}(ha) + \frac{U - \overline{\hat{Q}}(ha)}{N(ha)}$
29: $\underline{\hat{Q}}(ha) \leftarrow \underline{\hat{Q}}(ha) + \frac{L - \underline{\hat{Q}}(ha)}{N(ha)}$
30: **return** L, U
31: **end procedure**

Algorithm 8 Action Selection for SITH-PFT

1: **procedure** ACTIONSELECTION(b, h)
2: **if** $|C(h)| \leq k_a N(h)^{\alpha_a}$ **then** // action Prog. Widening
3: $a \leftarrow$ NEXTACTION(h)
4: $C(h) \leftarrow C(h) \cup \{a\}$
5: **end if**
6: **while** true **do**
7: Status, $a \leftarrow$ SELECTBEST(b, h)
8: **if** Status **then**
9: break
10: **else**
11: **for all** $b', o \in C(ha)$ **do**
12: RESIMPLIFY(b', hao)
13: **end for**
14: reconstruct $\overline{\hat{Q}}(ha), \underline{\hat{Q}}(ha)$
15: **end if**
16: **end while**
17: return a
18: **end procedure**
19: **procedure** SELECTBEST(b, h)
20: Status $\leftarrow$ true
21: $\tilde{a} \leftarrow \arg\max\{\underline{\mathrm{UCB}}(ha)\}$
22: gap $\leftarrow 0^a$
23: child-to-resimplify $\leftarrow \tilde{a}$
24: **for all** ha children of b **do**
25: **if** $\underline{\mathrm{UCB}}(h\tilde{a}) < \overline{\mathrm{UCB}}(ha) \wedge a \neq \tilde{a}$ **then**
26: Status $\leftarrow$ false
27: **if** $\overline{\hat{Q}}(ha) - \underline{\hat{Q}}(ha) >$ gap **then**
28: gap $\leftarrow \overline{\hat{Q}}(ha) - \underline{\hat{Q}}(ha)$
29: child-to-resimplify $\leftarrow a$
30: **end if**
31: **end if**
32: **end for**
33: **return** Status, child-to-resimplify
34: **end procedure**

Remark: Note that here we cannot use the "lazy variant" from Section 4.2 due to the fact that the MCTS requires selecting an action going down to the tree, see line 12 of Algorithm 7. Therefore, if the UCB bounds do still overlap, we cannot assure that the same acton will be selected as in case of UCB itself.

5.3 A Detailed Algorithm Description

Now we introduce our efficient variant of the Particle Filter Tree (PFT) presented in Sunberg and Kochenderfer (2018). We call our approach Simplified Information-Theoretic Particle Filter Tree (SITH-PFT). SITH-PFT (Alg. 7) incorporates the adaptive simplification into PFT-DPW. We adhere to the conventional notations as in Sunberg and Kochenderfer (2018) and denote by $G_{\mathrm{PF}(m)}(bao)$ a generative model receiving as input the belief b, an action a and an observation o (For clarity we substituted z' by o.), and producing the posterior belief b'. For belief update, we use a particle filter based on n_x state samples. A remarkable property of our efficient variant is the consistency of the belief tree. In other words, PFT and SITH-PFT have the same belief tree constructed with (16), while SITH-PFT enjoys substantial acceleration. By $C(ha)$ we denote the set of the children (posterior beliefs corresponding to the myopic observations) of the belief action node uniquely indexed by the history h with concatenated action a. Line 13 in Alg. 7 is the DPW technique from Sunberg and Kochenderfer (2018) with parameters k_o and α_o. The $N(\cdot)$ is the visitation count of belief or belief action nodes. In MCTS, the Q estimate is assembled by averaging the laces of the returns over simulations see Eq. 20. Each simulation yields a sum of discounted cumulative rewards. Therefore, by replacing the reward with adaptive lightweights bounds (22), we get corresponding discounted cumulative upper and lower bounds over the returns. Averaging the simulations (Alg. 7 lines 28-29), yields the bounds over the action-value function and the UCB bounds used in the routine `ActionSelection()` to be explained in the next paragraph.

Consider a belief-action node ha at level d with $\underline{\hat{Q}}(ha)$, $\overline{\hat{Q}}(ha)$. Suppose the algorithm selects it for bounds narrowing, as described in section 5.2 and Alg. 8 line 7. All tree nodes of which ha is an ancestor, contribute their

immediate $\overline{\rho}^s, \underline{\rho}^s$ bounds to $\hat{\overline{Q}}(ha)$, $\hat{\underline{Q}}(ha)$ computation. Thus, to tighten $\hat{\overline{Q}}(ha), \hat{\underline{Q}}(ha)$, we can potentially choose any candidate node(s) in the subtree of ha. Each child belief node of ha is sent to the resimplification routine (Alg. 8 lines $11-13$), which performs the following tasks. First, it selects the action (Alg. 9 line 7) that will participate in the subsequent resimplification call and sends all its children beliefs nodes to the recursive call further down the tree (Alg. 9 line 8-10). Secondly, It refines the belief node $\overline{\rho}, \underline{\rho}$ according to the specific *resimplification strategy* (Alg. 9 lines $3, 4, 12, 18$). Thirdly, it reconstructs $\hat{\overline{Q}}(ha)$, $\hat{\underline{Q}}(ha)$ once all the child belief nodes of ha have returned from the resimplification routine (Alg. 9 line 11) as we thoroughly explain in the next section. Fourthly, it engages the rollout resimplification routine according to the specific *resimplification strategy* (Alg. 9 lines 4, 13). Upon completion of this resimplification call initiated at ha, we obtain tighter immediate bounds of some of ha descendant belief nodes (including rollouts nodes). Accordingly, appropriate descendant of ha belief-action nodes bounds $(\hat{\overline{Q}}, \hat{\underline{Q}})$ shall be updated.

Many resimplification strategies are possible, below we present our approach. In Section 4.2 we presented a resimplicifation strategy based on gap. Now we adapt it to the MCTS setting.

Algorithm 9 Resimplification

1: **procedure** RESIMPLIFY(b, h)
2: **if** b is a leaf **then**
3: REFINEBOUNDS(b)
4: RESIMPLIFYROLLOUT(b, h)
5: **return**
6: **end if**
7: $\tilde{a} \leftarrow \arg\max_a \{N(ha) \cdot (\hat{\overline{Q}}(ha) - \hat{\underline{Q}}(ha))\}$
8: **for all** $b', o \in C(h\tilde{a})$ **do**
9: RESIMPLIFY(b', $h\tilde{a}o$)
10: **end for**
11: reconstruct $\hat{\overline{Q}}(h\tilde{a}), \hat{\underline{Q}}(h\tilde{a})$
12: REFINEBOUNDS(b)
13: RESIMPLIFYROLLOUT(b, h)
14: **return**
15: **end procedure**
16: **procedure** RESIMPLIFYROLLOUT(b, h)
17: $b^{\text{rollout}} \leftarrow$ find weakest link in rollout
18: REFINEBOUNDS(b^{rollout})
19: **end procedure**
20: **procedure** REFINEBOUNDS(b)
21: if (51) holds for b, refine its $\overline{\rho}^{s+1}, \underline{\rho}^{s+1}$ and promote its simplification level
22: **end procedure**

5.4 Specific Resimplification Strategy: `PT-Gap`

In this section, we explain the resimplification procedure in more detail. In particular we present a specific resimplification strategy and further show that this strategy is valid according to Definition 1. When some sibling belief action nodes have overlapping bounds (Fig. 3a, Fig. 7), we strive to avoid tightening them all at once since fewer resimplifications lead to greater acceleration (speedup). Thus, we choose a single ha-node that causes the largest "gap", denoted by G, between its bounds (see Alg. 8 lines 24-30), where G is defined by (47). Further, we tighten the bounds down the branch of the chosen node (see Alg. 8 lines 11-13) for each member of $C(ha)$, the set of children of ha. Since the bounds converge to the actual reward (Assumption 2) we can guarantee that Alg. 8 will pick a single action after a finite number of resimplifications; thus, tree consistency is assured.

Specifically, we decide to refine $\overline{\rho}^s, \underline{\rho}^s$ of a belief node indexed by h' at depth d' within the subtree starting from a belief action node indexed by ha at depth d when

$$\gamma^{d-d'} \cdot (\overline{\rho}^s - \underline{\rho}^s) \geq \frac{1}{d} G(ha), \tag{51}$$

where $G(ha)$ corresponds to the gap (47) of the belief-action node ha that initially triggered resimplification in Alg. 8 line 24.

The explanation of resimplification strategy based on (51) is rather simple. The right hand side of (51) is the mean gap per depth/level in the sub-tree with ha as its root and spreading downwards to the leaves. Naturally, some of the nodes in this subtree have $\overline{\rho}^s - \underline{\rho}^s$ above or equal to the mean gap and some below. We want to locate and refine all those above or equal to it. For the left side of (51); the rewards are accumulated and discounted according to their depth. Thus, we must account for the relative discount factor. Note that the depth identified with the root is the horizon $d_{\max} = L$, as seen in Alg. 7 line 4, and the leaves are distinguished by depth $d = 0$. For each rollout originating from a tree belief node, we find the rollout node with the largest $\overline{\rho} - \underline{\rho}$ satisfying (51) term locally in the rollout and resimplify it (Alg. 9 lines 4,13). To choose the action to continue resimplification down the tree, we take the action corresponding to the belief action node with the largest gap, weighted by its visitation count (Alg. 9 line 7). With this strategy, we aim to keep the belief tree at the lowest possible simplification level while maintaining belieftree consistency.

If the action selection procedure triggers resimplification, it modifies the bounds through the tree. Since the resimplification works recursively, it reconstructs the belief-action node bounds coming back from the recursion (Alg. 9 line 11). Similarly, the action dismissal procedure reconstructs $\hat{\overline{Q}}$ and $\hat{\underline{Q}}$ of the belief-action node at which the action dismissal is performed (Alg. 8 line 14). Moreover, on the way back from the simulation, we shall update the ancestral belief-action nodes of the tree. Specifically, we need to reconstruct each $\hat{\overline{Q}}$ and $\hat{\underline{Q}}$ that is higher than the deepest starting point of the resimplification (Alg. 7 line 23-25). The reconstruction is essentially a double loop. To reconstruct $\hat{\overline{Q}}(ha), \hat{\underline{Q}}(ha)$ we first query for all belief children nodes hao. We then query all belief-action nodes that are children to the hao, i.e. $haoa'$. The possibly modified immediate bounds $\underline{\rho}$ and $\overline{\rho}$ are taken from hao nodes and the $\hat{\overline{Q}}(\cdot)$, $\hat{\underline{Q}}(\cdot)$ bounds are taken from the $haoa'$ nodes. Importantly, each of the bounds is weighted according to the proper visitation count.

5.5 Guarantees

In this section we first show that the resimplification strategy suggested in the previous section is valid.

Lemma 1. Validity of the suggested resimplification strategy. *The resimplification strategy presented in Section 5.4 promotes the simplification level of at least one reward in the rollout or belief tree. Alternatively, all the rewards are at the maximal simplification level $n_{\max}$. In other words the suggested resimplifcation strategy is valid.*

We provide the complete proof in Appendix 11.2. Having proved the validity of the suggested resimplification strategy, we proceed to the monotonicity and convergence of UCB bounds from (48) and (49).

Lemma 2. Monotonicity and convergence of UCB bounds. *The UCB bounds are monotonic as a function of the number of resimplifications and after at most $n_{\max} \cdot M$ resimplifications we have that*

$$\overline{\mathrm{UCB}}(ha) = \underline{\mathrm{UCB}}(ha) = \mathrm{UCB}(ha) \tag{52}$$

We provide the proof in Appendix 11.3. Now, using Lemma 2, we prove that SITH-PFT (Alg. 7) yields the same belief tree and the same best action as PFT.

Theorem 2. *SITH-PFT and PFT are Tree Consistent Algorithms for* **any** *valid resimplification strategy.*

Theorem 3. *SITH-PFT provides the same solution as PFT for* **any** *valid resimplification strategy.*

We provide the full proofs of Theorems 2 and 3 in Appendix 11.4 and 11.5, respectively. We showed that for any valid resimplification strategy SITH-PFT is guaranteed to construct the same belief tree as PFT and select the same best action at the root. From Lemma 1, our resimplification strategy is valid. Thus, we achieved the desired result.

6 Specific Simplification and Information-theoretic Bounds

In this section we focus on a specific simplification in the context of a continuous state space and nonparametric beliefs represented by n_x weighted particles,

$$b \triangleq \{w^i, x^i\}_{i=1}^{n_x}. \tag{53}$$

Suggested Simplification: Given the belief representation (53), the simplified belief is a subset of n_x^s particles, sampled from the original belief, where $n_x^s \leq n_x$. More formally:

$$b_k^s \triangleq \left\{(x_k^i, w_k^i) \middle| i \in A_k^s \subseteq \{1, 2, \ldots, n_x\}, |A_k^s| = n_x^s\right\}, \tag{54}$$

where A_k^s is the set of particle indices comprising the simplified belief b_k^s for time k.

Increasing the level of simplification is done *incrementally*. Specifically, when resimplification is carried out, new indices are drawn from the sets $\{1, 2, \ldots, n_x\} \setminus A_k^s$ and and included to the set A_k^s. This operation promotes the simplification level to $s+1$ and defines A_k^{s+1}.

6.1 Novel Bounds Over Differential Entropy Estimator

As one of our key contributions, we now derive novel analytical bounds for the differential entropy estimator from Boers et al. (2010). These bounds can then be used within our general simplification framework presented in the previous sections. To calculate differential entropy

$$\mathcal{H}(b(x_k)) \triangleq -\int b(x_k) \cdot \log\left(b(x_k)\right) \mathrm{d}x_k,$$

one must have access to the manifold representing the belief. In a nonparametric setting this manifold is out of reach. We have to resort to approximations. Several approaches exist. One of them is using Kernel Density Estimation (KDE) as done, e.g., by Fischer and Tas (2020). Here, however, we consider the method proposed by Boers et al. (2010). This method builds on top of usage of motion and observation models such that

$$\hat{\mathcal{H}}(b_k, a_k, z_{k+1}, b_{k+1}) \triangleq \log\left[\sum_{i=1}^{n_x} \mathbb{P}_O(z_{k+1}|x_{k+1}^i) w_k^i\right] - \tag{55}$$
$$-\sum_{i=1}^{n_x} w_{k+1}^i \cdot \log\left[\mathbb{P}_O(z_{k+1}|x_{k+1}^i) \sum_{j=1}^{n_x} \mathbb{P}_T(x_{k+1}^i|x_k^j, a_k) w_k^j\right].$$

One can observe this method requires access to samples representing both b_k and b_{k+1}; thus, this corresponds to an information-theoretic reward of the form $\rho^I(b_k, a_k, z_{k+1}, b_{k+1})$. Note that as explained in Section 3 such a reward is tied to b_{k+1}.

For the sake of clarity and to remove unnecessary clutter we apply an identical simplification described by (54) to both beliefs b_k and b_{k+1}. The simplification indices for both beliefs are defined by A_{k+1}^s. However this is not an inherent limitation. One can easily maintain two sets of indices so as the theory presented below is developed to this more general setting. Moreover, as mentioned in Section 3, we have the same belief b_{k+1} also participating in $\rho^I(b_{k+1}, a_{k+1}, z_{k+2}, b_{k+2})$. In this reward, the simplification indices for b_{k+1} will according to A_{k+2}^s (and not according to A_{k+1}^s).

Utilizing the chosen simplification (54), we now introduce the following upper and lower bounds on (55).

Theorem 4. Adaptive bounds on differential entropy estimator. *The estimator (55) can be bounded by*

$$\begin{aligned} &\ell(b_k, a_k, z_{k+1}, b_{k+1}; A_k^s, A_{k+1}^s) \leq \\ &\leq -\hat{\mathcal{H}}(b_k, a_k, z_{k+1}, b_{k+1}) \leq \\ &\leq u(b_k, a_k, z_{k+1}, b_{k+1}; A_k^s, A_{k+1}^s), \end{aligned} \tag{56}$$

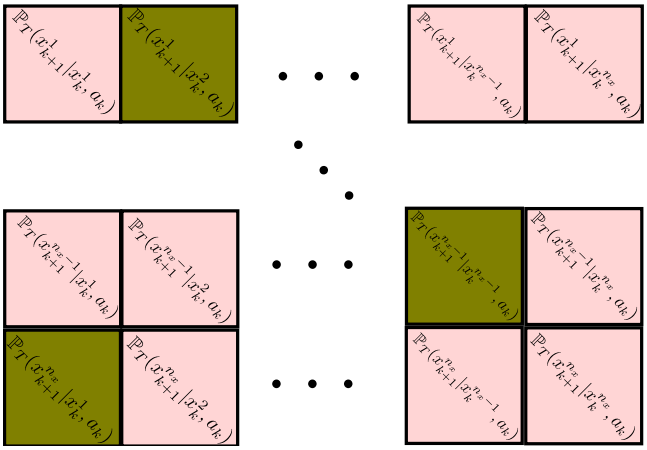


Figure 8. Schematic visualization of calculations reuse principle in bounds. We select **columns** using indexes from set A^s_k and rows by A^s_{k+1}. We marked by **olive** color resulting constituents of the bounds.

where

$$u \triangleq -\log\left[\sum_{i=1}^{n_x}\mathbb{P}_O(z_{k+1}|x^i_{k+1})w^i_k\right] + \tag{57}$$

$$+\sum_{i\notin A^s_{k+1}} w^i_{k+1}\cdot\log\left[m\cdot\mathbb{P}_O(z_{k+1}|x^i_{k+1})\right] +$$

$$+\sum_{i\in A^s_{k+1}} w^i_{k+1}\cdot\log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1})\sum_{j=1}^{n_x}\mathbb{P}_T(x^i_{k+1}|x^j_k,a_k)w^j_k\right]$$

$$\ell \triangleq -\log\left[\sum_{i=1}^{n_x}\mathbb{P}_O(z_{k+1}|x^i_{k+1})w^i_k\right] + \tag{58}$$

$$+\sum_{i=1}^{n_x} w^i_{k+1}\cdot\log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1})\sum_{j\in A^s_k}\mathbb{P}_T(x^i_{k+1}|x^j_k,a_k)w^j_k\right]$$

and where superscript s is the discrete level of simplification $s\in\{1,2,\ldots,n_{\max}\}$, $m\triangleq\max_{x',x,a}\mathbb{P}_T(x'|x,a)$ and A^s_k, $A^s_{k+1}\subseteq\{1,2,\ldots,n_x\}$.

See proof in Appendix 11.6. Theorem 4 accommodates different sets $A^s_k\neq A^s_{k+1}$. These sets denote sets of particle indices from b_k and b_{k+1} for simplification level s. In general, each of these sets can have its own simplification level. However, this is out of the scope of this paper. Here, both sets A^s_k, A^s_{k+1} have the same simplification level, as well as the number of levels. Yet, the number of particles at each level can vary between A^s_k and A^s_{k+1}. Each subsequent level (low to high) defines a larger set of indices such that higher levels of simplification (i.e. more samples) correspond to tighter and lower levels of simplification correspond to looser bounds. Note that the bounds (57) and (58) actually use the original and simplified beliefs so it settles with Eqs. (21) and (22).

Importantly, by caching the shared calculations of both bounds in the same time instance, we never repeat the calculation of these values and obtain maximal speedup. Without compromising on the solution's quality we are accelerating the online decision making process.

6.2 Bounds Properties and Analysis

We now turn to analysis of the bounds and investigation of their properties. Allow us to start from computational complexity. We then examine monotonicity and convergence of the bounds and reuse of calculations.

6.2.1 Computational complexity Eqs. (57) and (58) suggest that the bounds are cheaper to calculate than $\hat{\mathcal{H}}$ from (55), with complexity of $O(n^s_x\cdot n_x)$ instead of $O(n^2_x)$, where $n^s_x\triangleq|A^s_k|\equiv|A^s_{k+1}|$. Altogether, time saved for all belief nodes in the tree will result in the total speedup of our approach.

6.2.2 Monotonicy and Convergence

Theorem 5. Monotonicity and convergence. *The bounds from (56) are monotonic (Assumption 1) and convergent (Assumption 2) to (55).*

See proof in Appendix 11.7. Finally, bounding (55) using Theorem 4 corresponds, in our general framework from Section 3, to (21).

6.2.3 Re-use of Calculations The bounds can be tightened on demand incrementally without an overhead. Moving from simplification level s to level $s+1$, corresponds to adding some m additional particles to b^s to get b^{s+1}. For bounds calculation, we store the highlighted elements of the matrix in Fig. 8. This allows us to reuse the calculations when promoting the simplification level and **between the lower and the upper bounds** in a particular time index. Namely, after a few bounds-contracting iterations they are just the reward itself and the entire calculation is roughly time-equivalent to calculating the original reward. This will happen in a worst-case scenario.

We provide the theoretical time complexity analysis using the specific bounds (from Section 6.1) in Appendix 11.8. Now we are keen to present our simulations.

7 Adaptation Overhead

Whereas the bounds presented in Section 6 are incremental repeated resimplifications may lead to actually slower decision-making. This overhead is caused by additional algorithmics introduced by the resimplification routine. We can anticipate such scenarios when the actions are symmetrical in terms of the reward. However, as we empirically observed and will shortly present in the next section, in the setting of given belief tree the cases where the simplification is beneficial prevail. Especially in the LAZY variant since there the Alg. 5 engages resimplification routine only at the root of the belief tree.

In the setting of MCTS the situation is slightly more complicated. In UCB we cannot prune actions for eternity but only dismiss up until the next arrival to the belief node. This is because when MAB (defined in Section 2.3.3) converges it switches the current best action with arrivals to the belief node; such a behavior necessitates our simplification approach to tighten the bounds for many candidate actions. As a result in a MCTS setting we obtain less speedup than in the setting of a given belief tree considering LAZY variant (Alg. 5). Nevertheless in some problems the simplification approach is invaluable,

as for example, in the problem described in Section 8.1.3 and investigated in Section 8.3.5. Importantly, we can further accelerate resimplification routines by parallelization. However, this is out of the scope of this paper. All our implementations are single threaded.

8 Simulations and Results

We evaluate our proposed framework and approaches in simulation considering the setting of nonparametric fully continuous POMDP. Our implementation is built upon the JuliaPOMDP package collection (Egorov et al. 2017). For our simulations, we used a 16 cores 11th Gen Intel(R) Core(TM) i9-11900K with 64 GB of RAM working at 3.50GHz.

First, we study empirically the specific simplification and bounds from Section 6 and show that they become tighter as the number of particles increases. We, then benchmark our algorithms for planning in the setting of a given belief tree (Section 4) and in an anytime MCTS setting (Section 5). In the former setting, we compare SITH-BSP and LAZY-BSP against Sparse Sampling (Kearns et al. 2002). In an anytime MCTS setting, we compare SITH-PFT with PFT-DPW (Sunberg and Kochenderfer 2018) and IPFT (Fischer and Tas 2020). This performance evaluation is conducted considering three problems, as discussed next.

8.1 Problems under Consideration

We proceed to the description of the evaluated problems. In two first problems the immediate reward for b' is

$$\rho(b,a,z',b') = -(1-\lambda)\mathop{\mathbb{E}}_{x'\sim b'}\Big[r(x')\Big] - \lambda\hat{\mathcal{H}}(b,a,z',b'). \quad (59)$$

8.1.1 Continuous Light Dark Our first problem is *2D continuous Light-Dark problem*. The robot starts at some unknown point $x_0 \in \mathbb{R}^2$. In this world, there are spatially scattered beacons with known locations. Near the beacons, the obtained observations are less "noisy". The robot's mission is to get to the goal located at the upper right corner of the world. The state dependent reward in this problem is $r(x) = -\|x - x^{\text{goal}}\|_2^2$. The initial belief is $b_0 = \mathcal{N}(\mu_0, I \cdot \sigma_0)$, where we select $x_0 = \mu_0$ for actual robot initial state. The motion and observation models are

$$\mathbb{P}_T(x'|x,a) = \mathcal{N}(x + a, I \cdot \sigma_T), \quad (60)$$

and

$$O = \mathbb{P}_O(z|x) = \mathcal{N}(x - x^b, I \cdot \sigma_O \cdot \max\{d(x), d_{\min}\}), \quad (61)$$

respectively, where $d(x)$ is the ℓ^2 distance from robot's state x to the nearest beacon with known location denoted by x^b, and $d_{\min}$ is a tuneable parameter.

8.1.2 Target Tracking Our second problem is *2D continuous Target Tracking*. In this problem we have a moving target in addition to the agent. In this problem the belief is maintained over both positions, the agent and the target. The state dependent reward in this problem is $r(x) = -\|x^{\text{agent}} - x^{\text{target}}\|_2^2$. The motion model of the target and the agent follows

$$\mathbb{P}_T(\cdot|x,a) = \mathcal{N}(x^{\text{agent}} + a^{\text{agent}}, \Sigma_T) \cdot \mathcal{N}(x^{\text{target}} + a^{\text{target}}, \Sigma_T),$$

where by x we denote the concatenated $\{x^{\text{agent}}, x^{\text{target}}\}$. For target actions we use a circular buffer with $\{\uparrow, \uparrow, \leftarrow\}$ action sequence of unit length motion primitives. For simplicity we assume that in inference as well as in the planning session the agent knows the target action sequence. The observation model is also the multiplication of the observation model from the previous section with the additional observation model due to a moving target. Thus, the overall observation model is

$$\mathbb{P}_O(\cdot|x;\{x^{b,i}\}_{i=1}) = \mathcal{N}(x^{\text{agent}}, \Sigma_O(x^{\text{agent}};\{x^{b,i}\}_{i=1}))\cdot \\ \cdot\mathcal{N}(x^{\text{agent}} - x^{\text{target}}, \Sigma_O(x^{\text{agent}}, x^{\text{target}})),$$

where $\Sigma_O(x^{\text{agent}};\{x^{b,i}\}_{i=1})$ conforms to the observation model covariance described in Section 8.1.1 and

$$\Sigma_O(x^{\text{agent}}, x^{\text{target}}) = \quad (62)$$
$$\begin{cases} \sigma_T^2 I \|x^{\text{agent}} - x^{\text{target}}\|_2, \text{ if } \|x^{\text{agent}} - x^{\text{target}}\|_2 \geq d_{\min} \\ \sigma_O^2 I, \text{else} \end{cases}.$$

Before the planning experiments we study of the entropy estimators and the bounds presented in Theorem 4.

8.1.3 Safe Autonomous Localization Our third problem is a variation of the problem presented in Section 8.1.1. Here we change the reward to be the combination of localization reward and safety reward (Zhitnikov and Indelman 2022a)

$$\rho(b,a,z',b') = \overbrace{-\hat{\mathcal{H}}(b,a,z',b')}^{\text{localization reward}} + \\ + \underbrace{s\Big(2\cdot\mathbf{1}_{\{\mathrm{P}(\{x'\in\mathcal{X}^{\text{safe},\prime}\}|b')\geq\delta\}}(b') - 1\Big)}_{\text{safety reward}}. \quad (63)$$

Such a safety reward divides the candidate actions into two sets, the safe set and the unsafe. If the safety parameter s is sufficiently large to assure that safe action is selected, these two sets are detached enough in terms of safety reward and the unsafe set is substantially inferior such that there is no point to calculate localization reward precisely over this set of actions. There, we can, without any harm for decision-making outcome, substitute differential entropy by the bounds at the low simplification levels. This aspect makes the simplification paradigm invaluable.

8.2 Entropy Estimators and Bounds Study

In this section, we experiment with a passive case of the continuous 2D Light Dark problem from Section 8.1.1. Our goal is to study the various entropy estimators and our derived bounds from Section 6 over the estimator developed in Boers et al. (2010). In this study, we manually supply the robot with an action sequence to conduct. This results in a single lace of the beliefs corresponding to observations that the robot actually obtained by executing a given externally action sequence. We also provide some attempt in this section to compare estimated reward with the exact analytical counterpart.

Over this sequence of the beliefs, at each time instance of the sequence we calculate minus differential entropy estimator (information) in four ways. The first is the Boers estimator (Boers et al. 2010) and our bounds from

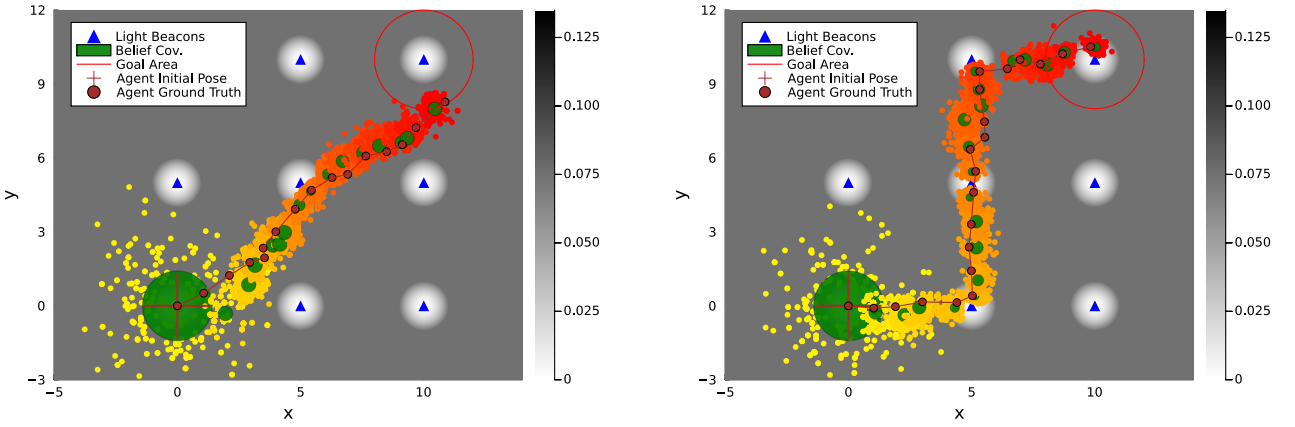


Figure 9. The plot shows the evolution of belief in terms of sets of particles along the actual trajectory of the robot. The color of the particles from yellow to red illustrate the evolution of the belief over time. The green ellipses represent the parametric Gaussian belief covariances obtained from update by Kalman filter. The canvas color here is $\sigma_O = \sigma_T = 0.075$ as in equations (60) and (61) respectively. **(a)** Our first scenario. **(b)** Our second scenario.

Theorem 4. The second is KDE approximation as done by Fischer and Tas (2020). The third is the naive calculation of discrete entropy over the the particles weights: $\hat{\mathcal{H}}(b) = -\sum_i w^i \cdot \log w^i$. The fourth estimator is analytical and it requires additional explanation. If we make an unrealistic assumption that robot's ground truth state from which the observation has been taken is known, plug it into the covariance matrix of (61) and set prior belief to be Gaussian; the motion and observation models met all the requirement for the exact update by Kalman Filter (linear additive models). For the proof see Thrun et al. (2005). In this case the belief stays Gaussian and the differential entropy has closed form solution.

We have two scenarios. In the first scenario, the robot moves diagonally to the goal using a unit length action $\nearrow$ (Fig. 9a) fifteen times. Along the way, it passes close-by two beacons. Consequentially, the robot's information about its state peaks twice. In our second scenario the robot moves five times to the right $\rightarrow$ followed by ten times $\uparrow$ and again five times to the right $\rightarrow$ (Fig. 9b).

The prior belief in this setting follows a Gaussian distribution $b_0 = \mathcal{N}\left(\begin{pmatrix}0.0\\0.0\end{pmatrix}, \begin{pmatrix}2.0 & 0.0\\0.0 & 2.0\end{pmatrix}\right)$, the motion and observation models parameters are $\sigma_O = \sigma_T = 0.075, d_{\text{min}} = 0.0001$. The number of unsimplified belief weighted particles is $n_x = 300$. For creating initial weighted particles we use the following proposal

$$
\begin{aligned}
q = 0.25 \cdot \mathcal{N}\left(\begin{pmatrix}0.0\\1.0\end{pmatrix}, \begin{pmatrix}2.0 & 0.0\\0.0 & 0.2\end{pmatrix}\right) + \\
+0.25 \cdot \mathcal{N}\left(\begin{pmatrix}1.0\\0.0\end{pmatrix}, \begin{pmatrix}2.0 & 0.0\\0.0 & 0.2\end{pmatrix}\right) + \\
+0.25 \cdot \mathcal{N}\left(\begin{pmatrix}-1.0\\0.0\end{pmatrix}, \begin{pmatrix}2.0 & 0.0\\0.0 & 0.2\end{pmatrix}\right) + \\
+0.25 \mathcal{N}\left(\begin{pmatrix}1.0\\-1.0\end{pmatrix}, \begin{pmatrix}2.0 & 0.0\\0.0 & 0.2\end{pmatrix}\right).
\end{aligned}
$$

The initial weights are the ratio $w(x) = \frac{b_0(x)}{q(x)}$.

To examine the bounds monotonical convergence with a growing number of simplified belief particles we plot the bounds (57) and (58) for minus entropy estimator (55) alongside estimators described above for the entire robot trajectory of the beliefs.

The results for the first and second scenarios are provided in Figs. 10 and 11, respectively. For both scenarios we observe that the bounds become tighter as the number of particles of simplified belief n_x^s increases. We also witness that all estimators vary but the overall trend is similar, putting aside the discrete entropy over the weights. The discrete entropy over the weights fails to adequately represent the uncertainty of the belief. This is an anticipated result. Let us proceed to the planning experiments.

8.3 *Planning*

In this section we study and benchmark our efficient planning algorithms. In our algorithms 1 and 5 the tree is build by SS (Kearns et al. 2002) such that the given belief tree is obtained when the algorithm descends to the leafs. We first compare Alg. 1 and 5 versus SS. We then proceed to simulations in an anytime MCTS setting.

For all further experiments, the belief is approximated by a set of n_x weighted samples as in (53). The robot does replanning after each executed action.

8.3.1 *Acceleration measures*

Let us begin this section by describing our measures of acceleration. We report planning time speedup in terms of saved accesses to particles.

The following speedup is based on the final number of simplified beliefs particles required for planning

$$
\frac{\sum_i \left(n_{i,x}^2 - n_{i,x}^s n_{i,x}\right)}{\sum_i n_{i,x}^2} \cdot 100, \tag{64}
$$

where the summation is over the future posterior beliefs in all the belief trees in a number of a consecutive planning sessions in particular scenario. Eq. (64) measures relative speedup without time spent on resimplifications. It is calculated at the end of several consecutive planning sessions. To calculate speedup according to (64) one shall pick up the **final** number of particles of simplified belief $n_{i,x}^s$

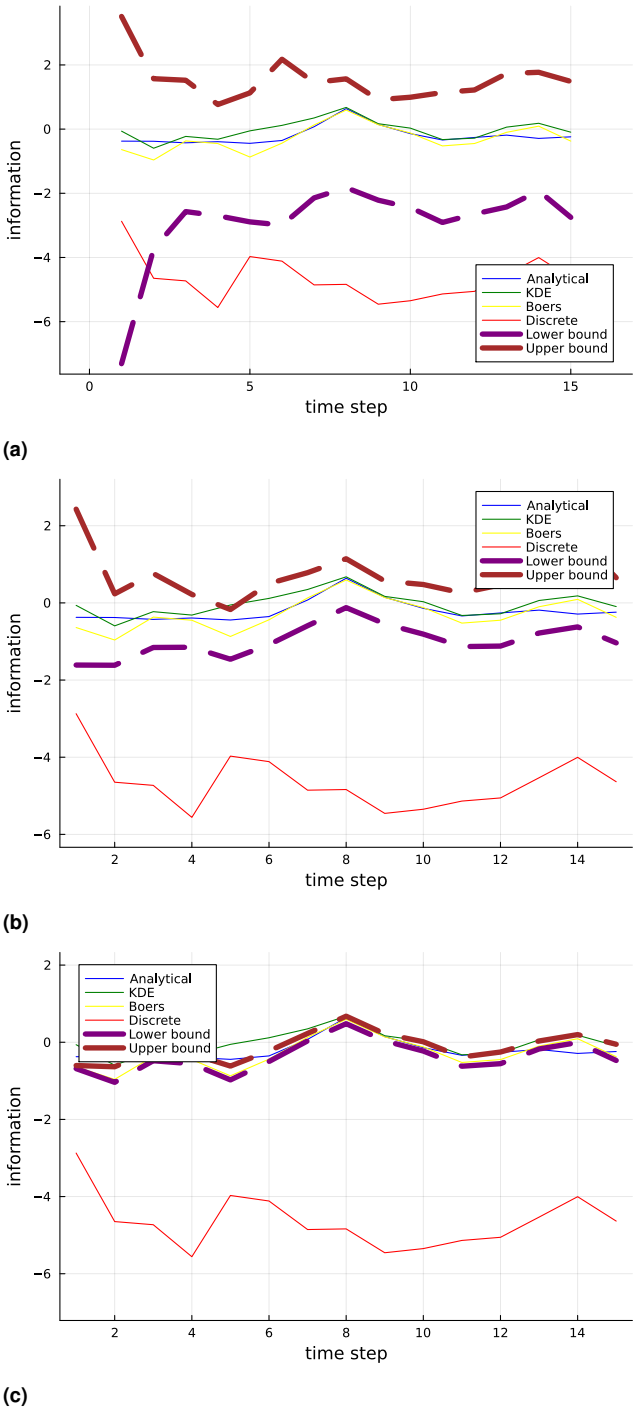


Figure 10. Bounds convergence for our first scenario $n_x = 300$ **(a)** $n_x^s = 30$ particles **(b)** $n_x^s = 150$ particles **(c)** $n_x^s = 270$ particles.

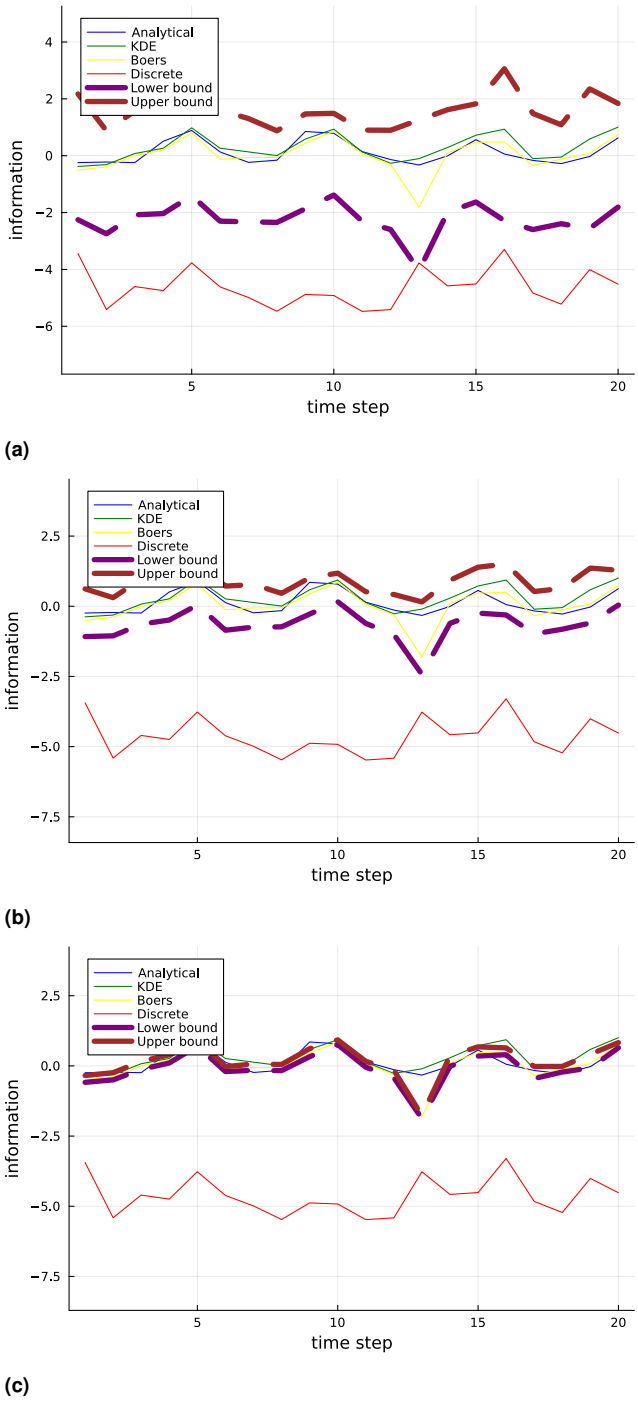


Figure 11. Bounds convergence for our second scenario $n_x = 300$ **(a)** $n_x^s = 30$ **(b)** $n_x^s = 150$ particles **(c)** $n_x^s = 270$ particles.

used for the simplified reward for each belief node i, sum over all the nodes of the belief trees (given or constructed on the fly) from planning sessions, make a calculation portrayed by (64). Importantly, acceleration measure (64) assumes that time of evaluating the motion and observation models do not vary from one evaluation to another. If the number of belief particles is not not depending on the belief ($n_{i,x} = n_x$) we can further simplify the (64) to

$$\frac{\sum_i \left(n_x - n_{i,x}^s\right)}{\sum_i n_x} \cdot 100. \tag{65}$$

To calculate planning time speedup we use the following metric

$$\frac{t_{\text{baseline}} - t_{\text{our}}}{t_{\text{baseline}}} \cdot 100. \tag{66}$$

If the quantities (64) and (66) are identical we can conclude that there will be no overhead from resimplifications and adapting the bounds. Note also that in the first place it is not clear how many particles n_x for belief representation to take. The number of particles n_x shall be as large as possible due to fact that we do not know when the belief represented

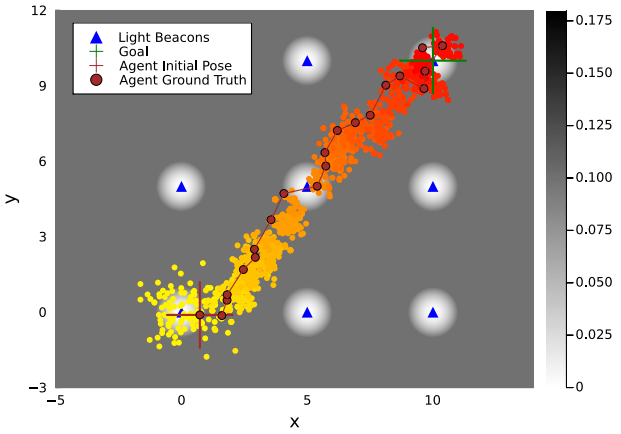


Figure 12. Exemplary 2D Light Dark problem planning scenario. Here we present the first trial of configuration $\lambda = 0.5$ of Table 1.

by weighted particles will converge to the corresponding theoretical belief.

To thoroughly study the acceleration yielded by our simplification paradigm we calculate total speedup over a number of the consecutive planning sessions in terms of particles in accordance to (64) and in terms of time in accordance to (66).

8.3.2 Results for 2D Continuous Light-Dark in the Setting of a Given Belief Tree We start from the problem described in Section 8.1.1. Our action space is constituted by motion primitives of unit length $\mathcal{A} = \{\rightarrow, \nearrow, \uparrow, \nwarrow, \leftarrow, \swarrow, \downarrow, \searrow\}$. In this problem the selected parameters are $\sigma_T = \sigma_O = 0.1$, $d_{\min} = 0.0001$, $\gamma = 0.95$. We simulate 15 trials of 20 consecutive alternating planning and action execution sessions. Fig. 12 shows an exemplary trial of 20 executions of the best action identified by the robot.

We investigate the influence of the parameter λ on speedup in Table 1 and the impact of changing the number of particles in Table 2. In both tables we see the particles speedup (column 4) and the time speedup (column 5). As expected with increasing values of λ (column 3) the speedup diminishes. LAZY-BSP (Alg. 5) produces larger speedup in terms of particles (column 4) and time (column 5) than SITH-BSP (Alg. 1). All three algorithms always selected the same optimal action. We observe that the return is always identical (column 9). Significant time speedup is obtained in the range of $35\% - 70\%$ for LAZY-BSP depending on the values of λ. For the SITH-BSP we see less time speedup ranging from 65% to 10% with increasing λ.

In all tables the number of motion and observation model calls does not include belief update calls but only the calls for reward or bounds calculation. The number of accesses to the observation model is always the same for all three algorithms (column 8). This agrees with the structure of the bounds (57) and (58). For the baseline SS, up to rounding errors, the number of motion model accesses, as we anticipated, is the squared number of unsimplified belief particles multiplied by number belief nodes in the tree minus one for root belief, multiplied by number of planning sessions (column 7 in the tables). This is in agreement with (55). Also, for all three algorithms the number of accesses to the observation model was the number of particles of unsimplified belief minus one for root belief, multiplied by the number of belief nodes in the tree, multiplied by the number of planning sessions.

We see that, while having larger particle speedup (column 3), LAZY-BSP makes more resimplification calls (column 6) than SITH-BSP. Observing the histograms of simplification levels in Fig. 13, we understand that LAZY variant of resimplification strategy leads to lower simplification levels of the rewards at the deepest level of a given belief tree. This was expected since the rewards at the upper levels of the belief tree participate in more laces and therefore their simplification level is promoted more times (See Alg. 5). In addition at the lowest levels reside more beliefs and corresponding rewards. This fact is corroborated by Table 3 where we witness that LAZY-BSP yields more beliefs, in the given in belief tree, with lower simplification levels than SITH-BSP.

8.3.3 Results for 2D Continuous Target Tracking in the Setting of a Given Belief Tree Our action space is $\mathcal{A} = \{\rightarrow, \nearrow, \uparrow, \nwarrow, \leftarrow, \swarrow, \downarrow, \searrow, \text{Null}\}$, where action Null means that agent doesn't take any action. In this problem we selected the parameters to be $d_{\min} = 0.0001$, $\Sigma_T = I \cdot \sigma_T$ where $\sigma_T = 0.1$ and $\sigma_O = 0.1$, $\gamma = 0.95$.

We simulate 15 trials of 15 consecutive alternating planning sessions and the executions by the robot of the selected optimal action. Fig. 14 shows an exemplary trial. We show the agent particles in Fig. 14a and the target particles in Fig. 14b. Similar to the previous section, we study speedup with growing λ in Table 4 and as function of various amounts of belief particles in Table 5. Again we observe that speedup diminishes with growing λ; LAZY-BSP (Alg. 5) produces a larger speedup in terms of particles (column 4) and time (column 5) than SITH-BSP (Alg. 1); accesses to motion and observation models are as expected; the return is identical for three algorithms.

In Fig. 15, which is associated with Table 6, we observe that to select an optimal action LAZY-BSP leaves more beliefs with lower simplification levels at the bottom of the given belief tree and produces more beliefs with lower simplification levels than SITH-BSP. A significant time speedup is obtained in the range of $30\% - 70\%$ for LAZY-BSP depending on the values of λ. For the SITH-BSP we see less time speedup ranging from 60% to 2% with increasing λ. The same best action was identified by SITH-BSP, LAZY-BSP and SS in all cases. Interestingly, in configuration $n_x = 350$ of Table. 5, for the first time we obtained that time speedup (66) is larger than particle speedup (64). This points to the fact that this run was so long due to large number of unsimplified belief particles $n_x = 350$ so that the time of access to motion and observation models varied.

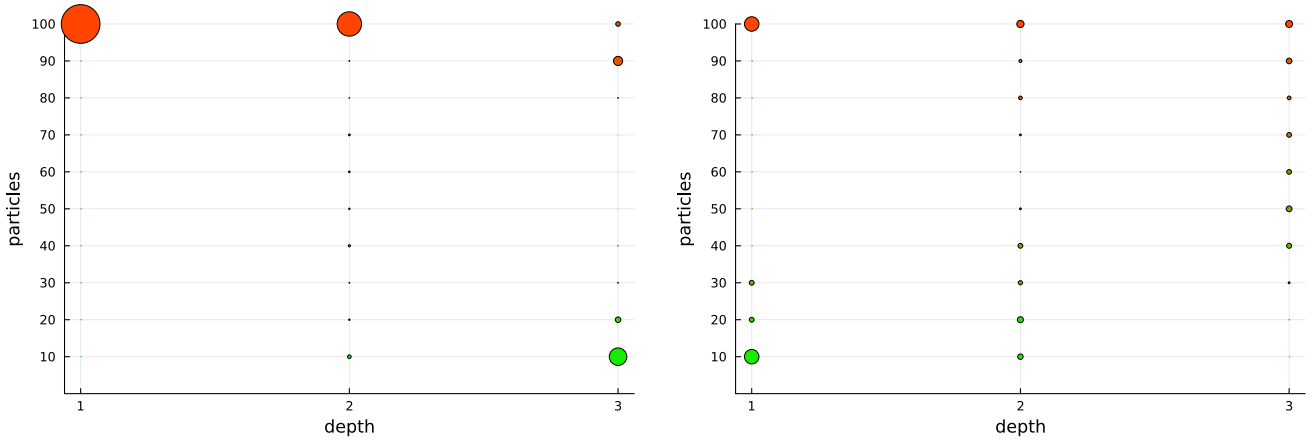
8.3.4 Experiments with MCTS In an anytime setting of MCTS we focus on the $2D$-continuous light dark problem from Section 8.1.1. We place a single "light beacon" in the continuous world. Here we changed the reward. The agent's goal is to get to location $(0, 0)$ and execute the terminal action - Null. Executing it within a radius of 0.5 from $(0, 0)$ will give the agent a reward of 200, and executing it outside the radius will yield a negative reward of -200. For all other actions the multi-objective reward function is $\rho(b, a, z, b') = -\mathop{\mathbb{E}}_{x\sim b'}[\|x\|_2] - \hat{\mathcal{H}}(b, a, z, b')$. The agent can move in eight evenly spread directions $\mathcal{A} = \{\rightarrow, \nearrow, \uparrow, \nwarrow, \leftarrow, \swarrow, \downarrow, \searrow$

Table 1. This table shows cumulative results of 20 consecutive alternating planning and execution sessions of the Continuous Light Dark problem averaged over 15 trials. Each planning session creates a single belief tree to perform a search for optimal action. This given belief tree has 4809 belief nodes. Overall, in 20 planning sessions, we have 96180 belief nodes. The horizon in each planning session is $L = 3$. The number of observations sampled from each belief action node is $n_z^1 = 1$, $n_z^2 = 3$, $n_z^3 = 3$ at the corresponding to superscripts depths $1, 2, 3$. This table examines the influence of various values of λ.

BSP Alg.	n_x	λ	particles speedup (64)	time speedup (66)	resimpl. calls (recursive)	motion model calls	obs. model calls	return ($\hat{V}$)
Alg 1 SITH	100	0.1	78.76 ± 0.20	64.44 ± 1.51	$2.05 \cdot 10^5 \pm 0.05 \cdot 10^5$	$3.13 \cdot 10^8 \pm 0.02 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-115.49 ± 16.58
Alg 5 LAZY			85.46 ± 1.22	71.59 ± 1.52	$10.71 \cdot 10^5 \pm 4.61 \cdot 10^5$	$2.38 \cdot 10^8 \pm 0.13 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-115.49 ± 16.58
SS						$9.62 \cdot 10^8 \pm 0.0$	$9.62 \cdot 10^6 \pm 0.0$	-115.49 ± 16.58
Alg 1 SITH	100	0.2	68.82 ± 0.32	53.59 ± 2.05	$3.36 \cdot 10^5 \pm 0.06 \cdot 10^5$	$4.22 \cdot 10^8 \pm 0.03 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-103.51 ± 14.91
Alg 5 LAZY			80.09 ± 1.52	65.01 ± 1.88	$25.65 \cdot 10^5 \pm 6.17 \cdot 10^5$	$3.01 \cdot 10^8 \pm 0.18 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-103.51 ± 14.91
SS						$9.62 \cdot 10^8 \pm 0.0$	$9.62 \cdot 10^6 \pm 0.0$	-103.51 ± 14.91
Alg 1 SITH	100	0.3	58.33 ± 0.52	42.76 ± 2.96	$4.13 \cdot 10^5 \pm 0.05 \cdot 10^5$	$5.40 \cdot 10^8 \pm 0.01 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-91.86 ± 13.88
Alg 5 LAZY			74.85 ± 2.63	58.94 ± 3.04	$42.66 \cdot 10^5 \pm 9.80 \cdot 10^5$	$3.59 \cdot 10^8 \pm 0.29 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-91.86 ± 13.88
SS						$9.62 \cdot 10^8 \pm 0.0$	$9.62 \cdot 10^6 \pm 0.0$	-91.86 ± 13.88
Alg 1 SITH	100	0.4	45.66 ± 0.83	29.33 ± 4.78	$4.70 \cdot 10^5 \pm 0.04 \cdot 10^5$	$6.84 \cdot 10^8 \pm 0.08 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-80.44 ± 11.77
Alg 5 LAZY			69.94 ± 1.89	53.85 ± 2.56	$59.05 \pm 10^5 \pm 8.76 \cdot 10^5$	$4.16 \cdot 10^8 \pm 0.22 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-80.44 ± 11.77
SS						$9.62 \cdot 10^8 \pm 0.0$	$9.62 \cdot 10^6 \pm 0.0$	-80.44 ± 11.77
Alg 1 SITH	100	0.5	34.46 ± 0.79	18.98 ± 4.16	$5.27 \cdot 10^5 \pm 0.05 \cdot 10^5$	$7.92 \cdot 10^8 \pm 0.01 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-66.3 ± 8.0
Alg 5 LAZY			63.6 ± 2.23	46.67 ± 2.81	$81.48 \cdot 10^5 \pm 8.52 \cdot 10^5$	$4.87 \cdot 10^8 \pm 0.24 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-66.3 ± 8.0
SS						$9.62 \cdot 10^8 \pm 0.0$	$9.62 \cdot 10^6 \pm 0.0$	-66.3 ± 8.0
Alg 1 SITH	100	0.6	25.09 ± 0.89	12.05 ± 4.83	$5.85 \cdot 10^5 \pm 0.05 \cdot 10^5$	$8.64 \cdot 10^8 \pm 0.05 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-55.36 ± 6.93
Alg 5 LAZY			56.32 ± 2.72	38.45 ± 3.65	$113.26 \cdot 10^5 \pm 11.45 \cdot 10^5$	$5.71 \cdot 10^8 \pm 0.28 \cdot 10^8$	$9.62 \cdot 10^6 \pm 0.0$	-55.36 ± 6.93
SS						$9.62 \cdot 10^8 \pm 0.0$	$9.62 \cdot 10^6 \pm 0.0$	-55.36 ± 6.93

Table 2. This table shows cumulative results of 20 consecutive alternating planning and execution sessions averaged over 15 trials of Continuous Light Dark problem. The given belief tree in a single planning session has 4809 belief nodes. Overall, in 20 planning sessions, we have 96180 belief nodes. The horizon in each planning session is $L = 3$. The number of observations sampled from each belief action node is $n_z^1 = 1$, $n_z^2 = 3$, $n_z^3 = 3$ at the corresponding to superscripts depths $1, 2, 3$. In this table we examine influence of various number of belief particles.

BSP Alg.	n_x	λ	particles speedup (64)	time speedup (66)	resimpl. calls (recursive)	motion model calls	obs. model calls	return ($\hat{V}$)
Alg 1 SITH	200	0.5	34.1 ± 0.8	25.01 ± 5.11	$5.30 \cdot 10^5 \pm 0.04 \cdot 10^5$	$31.80 \cdot 10^8 \pm 0.25 \cdot 10^8$	$19.24 \cdot 10^6 \pm 0.0$	-69.36 ± 7.95
Alg 5 LAZY			64.0 ± 2.98	51.71 ± 4.9	$83.95 \cdot 10^5 \pm 10.24 \cdot 10^5$	$19.35 \cdot 10^8 \pm 1.29 \cdot 10^8$	$19.24 \cdot 10^6 \pm 0.0$	-69.36 ± 7.95
SS						$38.47 \cdot 10^8 \pm 0.0$	$19.24 \cdot 10^6 \pm 0.0$	-69.36 ± 7.95
Alg 1 SITH	300	0.5	33.84 ± 0.83	18.67 ± 2.02	$5.30 \cdot 10^5 \pm 0.04 \cdot 10^5$	$71.74 \cdot 10^8 \pm 0.58 \cdot 10^8$	$28.85 \cdot 10^6 \pm 0.0$	-68.29 ± 8.42
Alg 5 LAZY			63.39 ± 3.44	47.34 ± 3.72	$84.09 \cdot 10^5 \pm 10.66 \cdot 10^5$	$43.91 \cdot 10^8 \pm 3.09 \cdot 10^8$	$28.85 \cdot 10^6 \pm 0.0$	-68.29 ± 8.42
SS						$86.56 \cdot 10^8 \pm 0.0$	$28.85 \cdot 10^6 \pm 0.0$	-68.29 ± 8.42
Alg 1 SITH	400	0.5	33.97 ± 0.85	25.34 ± 3.44	$6.65 \cdot 10^5 \pm 0.06 \cdot 10^5$	$181.50 \cdot 10^8 \pm 1.41 \cdot 10^8$	$54.51 \cdot 10^6 \pm 0.0$	-67.92 ± 11.52
Alg 5 LAZY			66.06 ± 2.3	53.74 ± 3.4	$106.90 \cdot 10^5 \pm 15.73 \cdot 10^5$	$105.35 \cdot 10^8 \pm 5.75 \cdot 10^8$	$54.51 \cdot 10^6 \pm 0.0$	-67.92 ± 11.52
SS						$218.05 \cdot 10^8 \pm 0.0$	$54.51 \cdot 10^6 \pm 0.0$	-67.92 ± 11.52



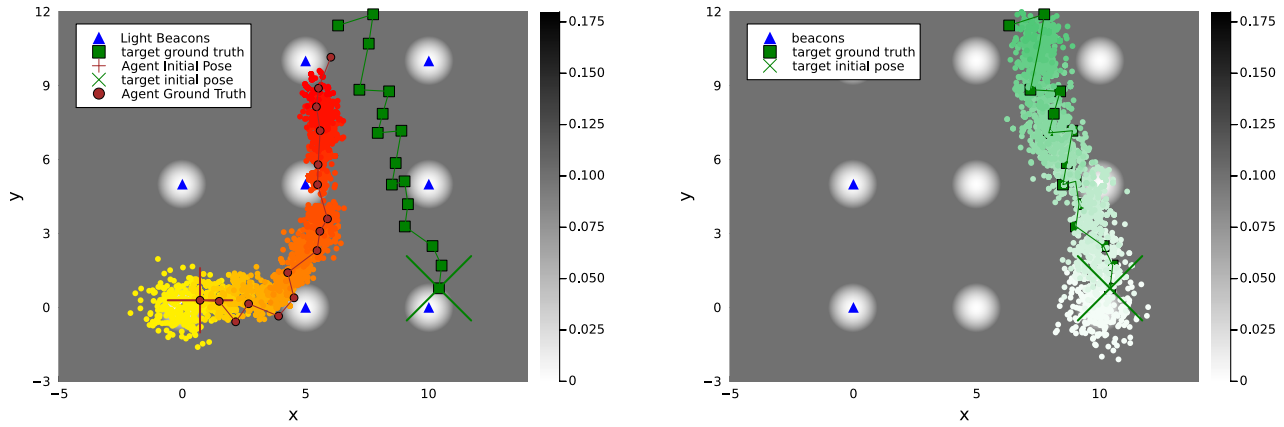
(a) **(b)**

Figure 13. Simplification levels at each depth of the given belief tree of Light Dark Problem (Section 8.1.1) after determining best action for one of the planning sessions. Here we present planning session 6 of the first trial of configuration $\lambda = 0.5$ of Table 1. The radius of circles represents the fraction of all nodes at a particular depth that have a particular simplification level. This figure is associated with Table 3. **(a)** LAZY-SITH-BSP Alg 5 **(b)** SITH-BSP Alg 1.

, Null$\}$. Motion and observation models, and the initial belief are $\mathbb{P}_T(\cdot|x,a) = \mathcal{N}(x + a, \Sigma_T)$, $\mathbb{P}_O(z|x) = \mathcal{N}(x, \min\{1, \| x - x^b \|_2^2\} \cdot \Sigma_O)$, $b_0 = \mathcal{N}(x_0, \Sigma_0)$ respectively. x^b is the 2D location of the beacon and all covariance matrices

Table 3. This table displays the numbers of the beliefs at each simplification level in a given tree after the identification of optimal action at the root b_k. Here we investigate Light Dark problem and belief tree as in Fig. 13. The given belief tree has 4809 belief nodes.

BSP Alg.	n_x	n_z^1	n_z^2	n_z^3	λ	L	simpl. level, particles									
							$s{=}1$ $n_x^s{=}10$	$s{=}2$ $n_x^s{=}20$	$s{=}3$ $n_x^s{=}30$	$s{=}4$ $n_x^s{=}40$	$s{=}5$ $n_x^s{=}50$	$s{=}6$ $n_x^s{=}60$	$s{=}7$ $n_x^s{=}70$	$s{=}8$ $n_x^s{=}80$	$s{=}9$ $n_x^s{=}90$	$s{=}10$ $n_x^s{=}100$
Alg 5 LAZY	100	1	3	3	0.5	3	2103	666	91	47	14	10	15	88	1094	680
Alg 1 SITH							30	61	241	618	696	567	576	465	684	870



(a) **(b)**

Figure 14. In this illustration we show second trial of Table. 5, configuration $n_x = 250$. The canvas color here is $\sigma_O = \sigma_T = 0.1$. **(a)** Agent particles **(b)** Target Particles.

Table 4. This table shows cumulative results of 15 consecutive alternating planning and action execution sessions averaged over 15 trials of Continuous Target Tracking problem. The given in a single planning session belief tree has 6814 belief nodes. Overall, in 15 planning sessions, we have 102210 belief nodes. The horizon in each planning session is $L = 3$. The number of observations sampled from each belief action node is $n_z^1 = 1$, $n_z^2 = 3$, $n_z^3 = 3$ at corresponding to superscripts depths $1, 2, 3$. In this table we examine influence of various values of λ.

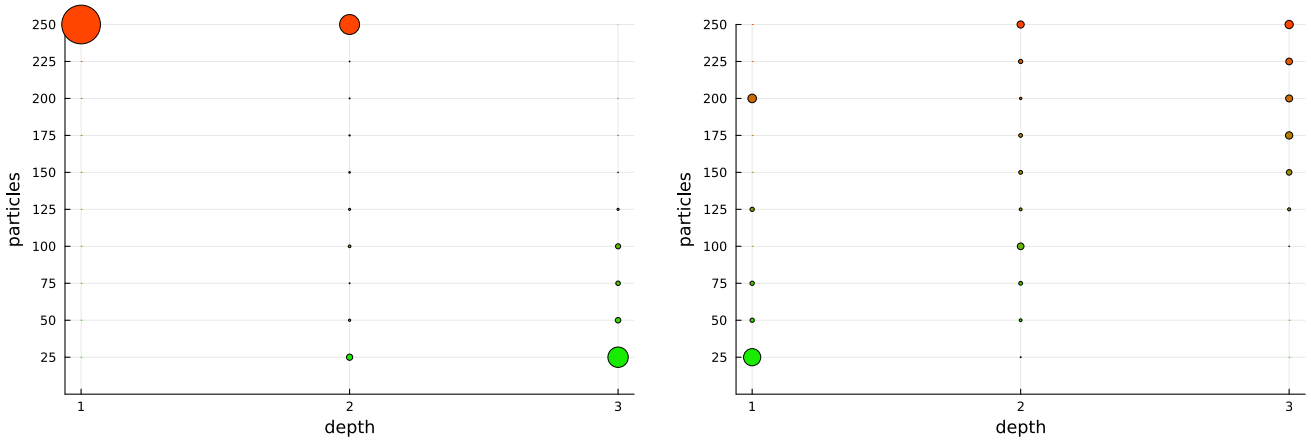
BSP Alg.	n_x	λ	particles speedup (64)	time speedup (66)	resimpl. calls (recursive)	motion model calls	obs. model calls	return ($\hat{V}$)
Alg 1 SITH	100	0.1	77.43 ± 0.26	60.3 ± 2.21	$1.69 \cdot 10^5 \pm 0.04 \cdot 10^5$	$3.48 \cdot 10^8 \pm 0.03 \cdot 10^8$	$10.22 \cdot 10^6 \pm 0.0$	-79.87 ± 9.69
Alg 5 LAZY			86.97 ± 1.28	71.18 ± 2.42	$7.44 \cdot 10^5 \pm 3.09 \cdot 10^5$	$2.32 \cdot 10^8 \pm 0.16 \cdot 10^8$	$10.22 \cdot 10^6 \pm 0.0$	-79.87 ± 9.69
SS						$10.22 \cdot 10^8 \pm 0.0$	$10.22 \cdot 10^6 \pm 0.0$	-79.87 ± 9.69
Alg 1 SITH	100	0.2	64.64 ± 0.57	46.39 ± 2.27	$2.60 \cdot 10^5 \pm 0.04 \cdot 10^5$	$5.03 \cdot 10^8 \pm 0.07 \cdot 10^8$	$10.22 \cdot 10^6 \pm 0.0$	-73.38 ± 9.8
Alg 5 LAZY			83.52 ± 1.7	67.24 ± 2.62	$16.52 \cdot 10^5 \pm 5.56 \cdot 10^5$	$2.75 \cdot 10^8 \pm 0.22 \cdot 10^8$	$10.22 \cdot 10^6 \pm 0.0$	-73.38 ± 9.8
SS						$10.22 \cdot 10^8 \pm 0.0$	$10.22 \cdot 10^6 \pm 0.0$	-73.38 ± 9.8
Alg 1 SITH	100	0.3	49.57 ± 0.93	29.25 ± 2.39	$3.14 \cdot 10^5 \pm 0.05 \cdot 10^5$	$6.86 \cdot 10^8 \pm 0.10 \cdot 10^8$	$10.44 \cdot 10^6 \pm 0.0$	-66.29 ± 9.3
Alg 5 LAZY			79.83 ± 2.55	63.34 ± 3.45	$26.61 \cdot 10^5 \pm 8.41 \cdot 10^5$	$3.21 \cdot 10^8 \pm 0.30 \cdot 10^8$	$10.44 \cdot 10^6 \pm 0.0$	-66.29 ± 9.3
SS						$10.22 \cdot 10^8 \pm 0.0$	$10.44 \cdot 10^6 \pm 0.0$	-66.29 ± 9.3
Alg 1 SITH	100	0.4	35.75 ± 1.09	14.45 ± 2.85	$3.61 \cdot 10^5 \pm 0.06 \cdot 10^5$	$8.33 \cdot 10^8 \pm 0.09 \cdot 10^8$	$10.44 \cdot 10^6 \pm 0.0$	-59.99 ± 8.05
Alg 5 LAZY			74.38 ± 3.5	55.69 ± 4.38	$42.74 \cdot 10^5 \pm 12.16 \cdot 10^5$	$3.90 \cdot 10^8 \pm 0.38 \cdot 10^8$	$10.44 \cdot 10^6 \pm 0.0$	-59.99 ± 8.05
SS						$10.22 \cdot 10^8 \pm 0.0$	$10.44 \cdot 10^6 \pm 0.0$	-59.99 ± 8.05
Alg 1 SITH	100	0.5	25.51 ± 1.04	6.44 ± 2.49	$4.05 \cdot 10^5 \pm 0.06 \cdot 10^5$	$9.18 \cdot 10^8 \pm 0.08 \cdot 10^8$	$10.44 \cdot 10^6 \pm 0.0$	-53.15 ± 7.03
Alg 5 LAZY			67.76 ± 3.88	47.94 ± 5.08	$63.18 \pm 15.71 \cdot 10^5$	$4.75 \cdot 10^8 \pm 0.44 \cdot 10^8$	$10.44 \cdot 10^6 \pm 0.0$	-53.15 ± 7.03
SS						$10.22 \cdot 10^8 \pm 0.0$	$10.44 \cdot 10^6 \pm 0.0$	-53.15 ± 7.03
Alg 1 SITH	100	0.6	18.06 ± 1.0	2.63 ± 2.32	$4.43 \cdot 10^5 \pm 0.06 \cdot 10^5$	$9.65 \cdot 10^8 \pm 0.06 \cdot 10^8$	$10.44 \cdot 10^6 \pm 0.0$	-46.97 ± 7.14
Alg 5 LAZY			59.53 ± 3.78	38.14 ± 4.69	$89.27 \cdot 10^5 \pm 15.03 \cdot 10^5$	$5.77 \cdot 10^8 \pm 0.43 \cdot 10^8$	$10.44 \cdot 10^6 \pm 0.0$	-46.97 ± 7.14
SS						$10.22 \cdot 10^8 \pm 0.0$	$10.44 \cdot 10^6 \pm 0.0$	-46.97 ± 7.14

are diagonal (i.e. $\Sigma = I \cdot \sigma^2$). We selected the following parameters $x_0 = \begin{pmatrix} -5.5 \\ 0.0 \end{pmatrix}, \Sigma_0 = \begin{pmatrix} 0.2 & 0.0 \\ 0.0 & 0.2 \end{pmatrix}, \sigma_T = \sigma_O = 0.075$. We experiment with 10 different configurations (rows of Table 7) that differ in n_x (number of particles), L (MCTS simulation depth), and #iter (number of MCTS simulation iterations per planning session). Each scenario comprises 10 planning sessions, i.e. the agent performs up to 10 planning action-executing iterations. The scenario stops if the best action determined in planning is Null. We repeat each experiment 25 times. In each such repetition we run PFT-DPW and SITH-PFT with the same seed and calculate the relative time speedup in percentage according to (66) where $t_{\mathrm{PFT-DPW}}$ and $t_{\mathrm{SITH-PFT}}$ are running times of a baseline and our methods respectively.

In all different configurations, we obtained significant time speedup of approximately 20% while achieving the exact same solution compared to PFT. In Table 7 we report the

Table 5. This table shows cumulative results of 15 consecutive alternating planning and execution sessions averaged over 15 trials of Continuous Target Tracking problem. The given belief tree has 6814 belief nodes. Overall, in 15 planning sessions, we have 102210 belief nodes. The horizon in each planning session is $L = 3$. The number of observations sampled from each belief action node is $n_z^1 = 1, n_z^2 = 3, n_z^3 = 3$ at corresponding to superscripts depths $1, 2, 3$. In this table we examine various numbers of belief particles.

BSP Alg.	n_x	λ	particles speedup (64)	time speedup (66)	resimpl. calls (recursive)	motion model calls	obs. model calls	return ($\hat{V}$)
Alg 1 SITH	150	0.5	25.19 ± 0.94	8.72 ± 2.4	$4.03 \cdot 10^5 \pm 0.03 \cdot 10^5$	$20.71 \cdot 10^8 \pm 0.15 \cdot 10^8$	$15.33 \cdot 10^6 \pm 0.0$	-54.0 ± 8.16
Alg 5 LAZY			68.36 ± 2.66	50.23 ± 3.2	$63.14 \cdot 10^5 \pm 9.15 \cdot 10^5$	$10.55 \cdot 10^8 \pm 0.65 \cdot 10^8$	$15.33 \cdot 10^6 \pm 0.0$	-54.0 ± 8.16
SS						$22.10 \cdot 10^8 \pm 0.0$	$15.33 \cdot 10^6 \pm 0.0$	-54.0 ± 8.16
Alg 1 SITH	250	0.5	23.87 ± 0.98	11.01 ± 3.93	$4.11 \cdot 10^5 \pm 0.05 \cdot 10^5$	$58.10 \cdot 10^8 \pm 0.40 \cdot 10^8$	$25.55 \cdot 10^6 \pm 0.0$	-55.57 ± 9.59
Alg 5 LAZY			66.18 ± 3.35	51.51 ± 3.83	$70.02 \cdot 10^5 \pm 12.74 \cdot 10^5$	$30.79 \cdot 10^8 \pm 2.37 \cdot 10^8$	$25.55 \cdot 10^6 \pm 0.0$	-55.57 ± 9.59
SS						$63.88 \cdot 10^8 \pm 0.0$	$25.55 \cdot 10^6 \pm 0.0$	-55.57 ± 9.59
Alg 1 SITH	350	0.5	23.95 ± 1.07	40.18 ± 10.29	$4.11 \cdot 10^5 \pm 0.03 \cdot 10^5$	$113.81 \cdot 10^8 \pm 0.89 \cdot 10^8$	$35.77 \cdot 10^6 \pm 0.0$	-55.62 ± 8.73
Alg 5 LAZY			66.36 ± 2.58	67.17 ± 4.86	$69.40 \cdot 10^5 \pm 10.08 \cdot 10^5$	$60.19 \cdot 10^8 \pm 3.62 \cdot 10^8$	$35.77 \cdot 10^6 \pm 0.0$	-55.62 ± 8.73
SS						$125.21 \cdot 10^8 \pm 0.0$	$35.77 \cdot 10^6 \pm 0.0$	-55.62 ± 8.73

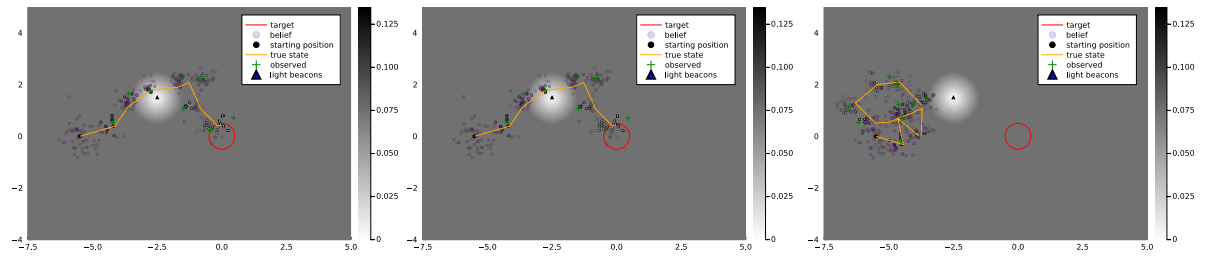


(a) **(b)**

Figure 15. Simplification levels at each depth of the given belief tree of Target Tracking problem (Section 8.1.2) after determining best action for one of the planning sessions.. Here we present planning session 6 of the first trial of configuration $n_x = 250$ of Table 5. The radius of circles represent the fraction of all nodes at particular depth that have a particular simplification level. This figure is associated with Table 6. **(a)** LAZY-SITH-BSP Alg 5 **(b)** SITH-BSP Alg 1.

Table 6. This table displays the numbers of the beliefs at each simplification level in given tree after the identification of optimal action at the root b_k. Here we investigate Target Tracking problem and belief tree as in Fig. 15. The size of given belief tree is 6814 belief nodes.

BSP Alg.	n_x	n_z^1	n_z^2	n_z^3	λ	L	simpl. level, particles									
							s=1 n_x^s=25	s=2 n_x^s=50	s=3 n_x^s=75	s=4 n_x^s=100	s=5 n_x^s=125	s=6 n_x^s=150	s=7 n_x^s=175	s=8 n_x^s=200	s=9 n_x^s=225	s=10 n_x^s=250
Alg 5 LAZY	250	1	3	3	0.5	3	3487	949	776	884	379	106	48	34	11	139
Alg 1 SITH							10	19	46	144	538	966	1266	1208	1164	1452



(a) SITH-PFT **(b)** PFT-DPW **(c)** IPFT

Figure 16. 2D Continuous Light Dark. The agent starts from an initial unknown location and is given an initial belief. The goal is to get to location $(0, 0)$ (circled in red) and execute the terminal action. Near the beacon (white light) the observations are less noisy. We consider multi-objective function, accounting for the distance to the goal and the differential entropy approximation (with the minus sign for reward notation). Executing the terminal action inside the red circle gives the agent a large positive reward but executing it outside it, will yield a large negative reward.

Table 7. Time speedup (66) obtained SITH-PFT versus PFT-DPW. The rows are different configurations of the number of belief particles n_x, maximal tree depth L, and the number of iterations per planning session. In all simulations SITH-PFT and PFT-DPW declared *identical* actions as optimal and exhibited *identical* belief trees in terms of connectivity and visitation counts.

(n_x, L, #iter.)	mean $\pm$std	max.	min.
(50, 30, 200)	19.35 ± 6.34	30.17	7.99
(50, 50, 500)	17.43 ± 5.4	33.49	10.72
(100, 30, 200)	21.97 ± 8.74	49.24	7.36
(100, 50, 500)	22.54 ± 6.33	36.09	13.65
(200, 30, 200)	26.27 ± 9.36	42.43	11.17
(200, 50, 500)	26.17 ± 7.64	44.31	14.43
(400, 30, 200)	21.88 ± 8.47	37.04	10.34
(400, 50, 500)	21.71 ± 6.01	32.69	9.67
(600, 30, 200)	20.27 ± 7.38	32.95	8.77
(600, 50, 500)	19.93 ± 6.48	31.26	6.49

Table 8. Total runtime of 25 repetitions of two algorithms.

(n_x, L, #iter.)	Algorithm	tot. plan. time [sec]
(50, 30, 200)	PFT-DPW	49.7
	SITH-PFT	40.25
(50, 50, 500)	PFT-DPW	125.05
	SITH-PFT	103.71
(100, 30, 200)	PFT-DPW	185.47
	SITH-PFT	145.08
(100, 50, 500)	PFT-DPW	460.29
	SITH-PFT	357.57
(200, 30, 200)	PFT-DPW	709.66
	SITH-PFT	526.18
(200, 50, 500)	PFT-DPW	1755.08
	SITH-PFT	1298.86
(400, 30, 200)	PFT-DPW	2672.56
	SITH-PFT	2099.0
(400, 50, 500)	PFT-DPW	6877.24
	SITH-PFT	5403.91
(600, 30, 200)	PFT-DPW	6335.09
	SITH-PFT	5056.96
(600, 50, 500)	PFT-DPW	15682.47
	SITH-PFT	12602.09

mean and standard error of (66) as well as maximum and minimum value. Remarkably, we observe that we never slowdown the PFT-DPW with SITH-PFT. We also present total running times of 25 repetitions of at most 10 (the simulation stops if best identified action is Null) planning sessions in Table 8. Note that we divided the total planning time by the number of planning sessions in each repetition.

An illustration of evaluated scenario can be found in Fig. 16. Note that SITH-PFT (Fig. 16a) yields an identical to PFT solution (Fig. 16b) while IPFT demonstrates a severely degraded behavior. We remind the purpose of our work is to speedup the PFT approach when coupled with information-theoretic reward. Since the two algorithms produce identical belief trees and action at the end of each planning session, there is no point reporting the algorithms *identical* performances (apart from planning time).

8.3.5 Localization with Collision Avoidance Solved by MCTS In this section, we investigate the application of three algorithms, IPFT (Fischer and Tas 2020), PFT-DPW (Sunberg and Kochenderfer 2018) and our SITH-PFT encapsulated by Alg. 7. The algorithmic implementation of IPFT boils down to making more simulations inside IPFT with substantially less number of belief particles subsampled from root belief.

Further, we discuss the quality and speed of IPFT. Representation of the belief with a tiny amount of particles induces larger error in differential entropy estimation and other parts of the reward function such as, for example, soft safety reward component in (63). The authors of (Fischer and Tas 2020) claim that IPFT averages differential entropies calculated from tiny subsets subsampled from the particle belief. However, observing the `SIMULATE` routine (similar to our in Alg. 7) in (Fischer and Tas 2020), we see that in practice this average is obtained through more simulations, starting from a new subsample from the root belief, with less number of particles, thereby averaging entropies calculated from different beliefs with less number of particles, but same history of actions and observations. The parameter K in (Fischer and Tas 2020) in practice is the visitation count $N(b)$ of each belief in the belief tree. There is no direct control of this parameter. In other words, to make a proper comparison we shall increase the number of `SIMULATE` calls inside IPFT by a factor $K = {}^{n_x}/_{m}$ where m is the size of the subsample from a belief represented by n_x particles. In such a way in both belief trees, built by IPFT and PFT-DPW, there are the same number of total particles. This is in contrast to using the same number of calls to `SIMULATE` in both trees. If the number of calls to `SIMULATE` is the same the number of particles in the tree built by IPFT will be much smaller than in the tree built by PFT-DPW. Do note that we cannot assure that the same K will be for each future history due to the exploratory nature of MCTS.

The speed of IPFT is linked with the rollout policy of MCTS. As we mentioned above, when the belief is represented by particles we know that asymptotically when the number of particles tends to infinity this representation converges to the theoretical belief for any given belief (Crisan and Doucet 2002). Therefore, we shall take as many particles as possible for the belief representation. Given that the size of subsample m in IPFT does not change, this will increase the parameter K and therefore slowdown IPFT. Because when the new belief node is expanded in the belief tree there is always a rollout initiated, a more complex rollout policy will slowdown IPFT more, yet, this is ultimately the question of how big the parameter K is.

As we observe in Fig. 17, IPFT is less accurate compared to PFT-DPW and SITH-PFT in spite of a much larger number of calls to `SIMULATE` routine compared to PFT-DPW and SITH-PFT. Clearly, better localization is closer to the beacons. In Fig. 17a we see that more trajectories went to completely different from beacons directions as opposed to Fig. 17c and Fig. 17b displaying identical results. From Fig. 18a we conclude that in 10 from 15 trials the information reward obtained in execution of the optimal action returned by IPFT was inferior to the corresponding reward obtained by SITH-PFT and PFT-DPW. From Fig. 18b we see that

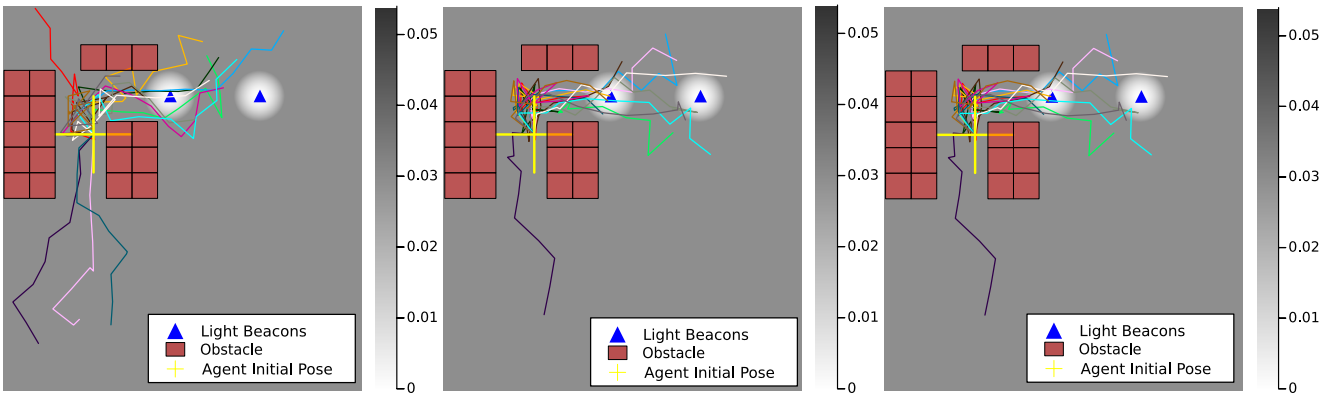


Figure 17. The plot shows 15 differently colored robot trajectories. Each such trajectory comprises ten time steps. In each such step the robot performs re-planning and executes the best action selected by an appropriate BSP algorithm. The color of each trajectory matches planning with the same seed in each plot. The canvas color here is $\sigma_O = \sigma_T = 0.03$ as in equations (60) and (61) respectively. The parameters are $n_x = 300$, $m = 20$, number of calls to SIMULATE of IPFT is 4500, the number of calls to SIMULATE of PFT-DPW and SITH-PFT is 300. In such a setting the constructed belief trees by these methods have the same number of total samples (see Section 8.3.5 for details). **(a)** Safe IPFT. **(b)** Safe SITH-PFT (Alg 7), **(c)** Safe PFT-DPW.

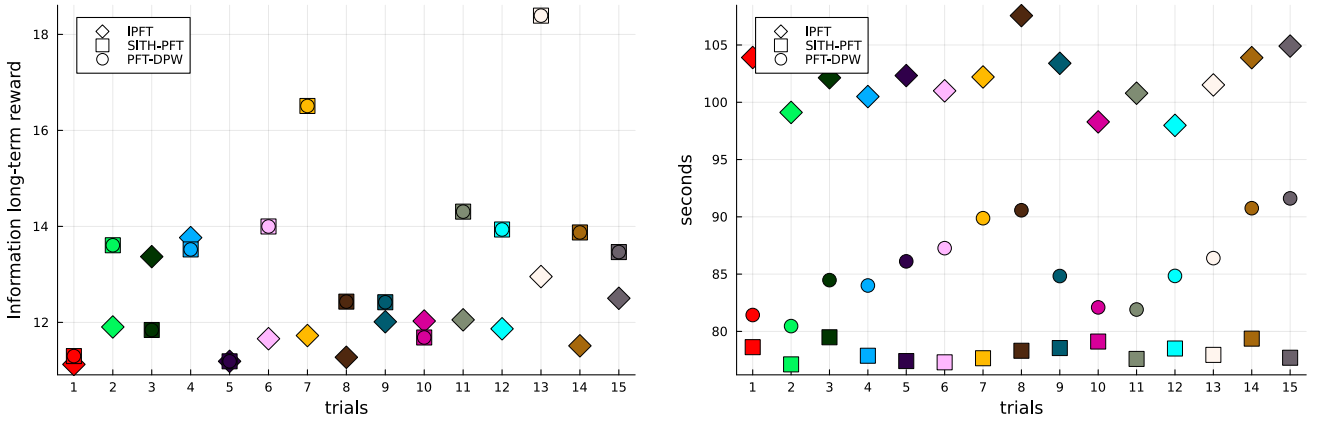


Figure 18. This plot is associated with Fig. 17. Each color matches the corresponding trajectory in Fig. 17. The parameters are $n_x = 300$, $m = 20$, $K = \frac{300}{20} = 15$ number of calls to SIMULATE of IPFT is 4500, the number of calls to SIMULATE of PFT-DPW and SITH-PFT is 300. In such a setting the constructed belief trees by these methods have the same number of total samples (see Section 8.3.5 for details). **(a)** Cumulative information reward as in (63) in the execution of the trajectory. Here the SITH-PFT curve and the PFT-DPW curve overlap. This is because the rewards are identical since the same best action is calculated by SITH-PFT and PFT-DPW; **(b)** Average planning times of 10 planning sessions in each trial.

IPFT is slowest from the three algorithms while SITH-PFT (Alg. 7) is the fastest *in all trials*.

8.4 Discussion

Although the speedup was significant and steady for all simulations, we did not observe growth in speed-up with growth of number of belief particles in any simulation. This can be explained by the fact that increasing number of particles of the belief (n_x) changes also the bounds because the parameter n_x is present in the bounds as well. The limitation of our approach is that it leans on converging bounds, which are not trivial to derive and specific for a particular reward function. In addition, it requires slightly more caching than the baseline. Our simplification approach may still be ill-timed, since the resimplifications take an additional toll in terms of running time.

9 Conclusions

We contributed a rigorous provable theory of adaptive multilevel simplification that accelerates the solution of belief-dependent fully continuous POMDP. Our theory always identifies the same optimal action or policy as the unsimplified analog. Our theoretical approach receives as input adaptive bounds over the belief-dependent reward. Using the suggested theory and any bounds satisfying stated conditions we formulated three algorithms, considering a given belief tree and an anytime MCTS setting. We also contributed a specific simplification for nonparametric beliefs represented by weighted particles and derived novel bounds over a differential entropy estimator. These bounds are computationally cheaper than the latter. Our experiments demonstrate that our algorithms are paramount in terms of computation time while guaranteed to have the same performance as the baselines. In the setting of the given

belief tree, we achieved a speedup up to 70%. In an anytime MCTS setting, our algorithm enjoyed the speedup of 20%.

10 Funding

This research was supported by the Israel Science Foundation (ISF) and by a donation from the Zuckerman Fund to the Technion Artificial Intelligence Hub (Tech.AI).

11 APPENDIX

11.1 Proof for Theorem 1

To shorten the notations we prove the theorem for value function under arbitrary policy. Note that by substituting the policy $\pi_{(\ell)+}$ by $\{\pi_\ell(b_\ell), \pi^*_{(\ell+1)+}\}$ where $a_\ell = \pi_\ell(b_\ell)$ we always can obtain action-value function. Without loosing generality assume the resimplification hits an arbitrary belief action node. The new upper bound will be

$$\overline{\hat{V}}(b_\ell, \pi_{\ell+}) + \frac{1}{M}\left(\underbrace{\overline{\Delta}^{s+1}(b,a,b') - \overline{\Delta}^{s}(b,a,b')}_{\leq 0}\right) \leq \overline{\hat{V}}(b_\ell, \pi_{\ell+}) \tag{67}$$

The new lower bound will be

$$\underline{\hat{V}}(b_\ell, \pi_{\ell+}) - \frac{1}{M}\left(\underbrace{\underline{\Delta}^{s+1}(b,a,b') - \underline{\Delta}^{s}(b,a,b')}_{\leq 0}\right) \geq \underline{\hat{V}}(b_\ell, \pi_{\ell+}) \tag{68}$$

where $M = n_z^d$ depending on the depth d of resimplified reward bound. Moreover if the inequalities involving increments are strict $\overline{\Delta}^{s}(b,a,b') > \overline{\Delta}^{s+1}(b,a,b')$ and $\underline{\Delta}^{s}(b,a) > \underline{\Delta}^{s+1}(b,a,b')$ also the retracting the bounds over Value function inequalities are strict. In case of MCTS, we have that $M = \frac{N(ha)}{N(h')}$ where history ha corresponds to b_ℓ and action a, and h' corresponds to b'. ■

11.2 Proof of Lemma 1

Recall that the bounds $\overline{\rho}, \underline{\rho}$ of belief nodes and "weakest link" rollout nodes are refined when the inequality (51) is encountered.

Assume in contradiction that the resimplification strategy does not promote any reward level and $G(ha) > 0$. This means that $G(ha)/d > 0$ and for all reward bounds the inequality $\gamma^{d-d'} \cdot (\overline{\rho} - \underline{\rho}) < \frac{1}{d} G(ha)$. This is not possible since $G(ha)/d$ is the mean gap with respect to simulations of MCT and the depth of the belief tree, multiplied by the appropriate discount factor, over all the nodes that are the descendants to ha . See equation (33). ■

11.3 Proof of Lemma 2

Observe that

$$\overline{\mathrm{UCB}}(ha) - \underline{\mathrm{UCB}}(ha) = \overline{\hat{Q}}(ha) - \underline{\hat{Q}}(ha). \tag{69}$$

We already proved the desired for $\overline{\hat{Q}}(ha), \underline{\hat{Q}}(ha)$ in Theorem 1. Using the convergence $\underline{\hat{Q}}(\cdot) = \hat{Q}(\cdot) = \overline{\hat{Q}}(\cdot)$ we obtain

$$\begin{aligned} &\underline{\hat{Q}}(\cdot) + c \cdot \sqrt{\log(N(h))/N(ha)} = \\ &\hat{Q}(\cdot) + c \cdot \sqrt{\log(N(h))/N(ha)} = \\ &\overline{\hat{Q}}(\cdot) + c \cdot \sqrt{\log(N(h))/N(ha)}. \end{aligned} \tag{70}$$

The proof is completed. ■

11.4 Proof of Theorem 2

We provide proof by induction on the belief tree structure.
Base: Consider an initial given belief node b_0. No actions have been taken and no observations have been made. Thus, both the PFT tree and the SITH-PFT tree contain a single identical belief node, and the claim holds.
Induction hypothesis: Assume we are given two identical trees with n nodes, generated by PFT and SITH-PFT. The trees uphold the terms of **Definition 2**.
Induction step: Assume in contradiction that different nodes were added to the trees in the next simulation (expanding the belief tree by one belief node by definition). Thus, we got different trees.
Two scenarios are possible:

Case 1. The same action-observation sequence $a_0, z_1, a_1, z_2 ... a_m$ was chosen in both trees, but different nodes were added.

Case 2. Different action-observation sequences were chosen for both trees, and thus, we got different trees structure.

Since the Induction hypothesis holds, the last action a_m was taken from the same node denoted h' shared and identical to both trees. Next, the same observation model is sampled for a new observation, and a new belief node is added with a rollout emanating from it. The new belief nodes and the rollout are identical for both trees since both algorithms use the same randomization seed and observation and motion models. Case 2 must be true since we showed Case 1 is false. There are two possible scenarios such that different action-observation sequences were chosen:

Case 2.1. At some point in the actions-observations sequence, different observations z_i, z_i' were chosen.

Case 2.2. At some point in the actions-observations sequence, PFT chose action $a^\dagger$ while SITH-PFT chose a different action, $\tilde{a}$, or got stuck without picking any action.

Case 2.1 is not possible since if new observations were made, they are the same one by reasons contradicting Case 1. If we draw existing observations (choose some observation branch down the tree) the same observations are drawn since they are drawn with the same random seed and from the same observations "pool". It is the same "pool" since the Induction hypothesis holds. Case 2.2 must be true since we showed Case 2.1 is false, i.e., when both algorithms are at the identical node denoted as h PFT chooses action $a^\dagger$, while SITH-PFT chooses a different action, $\tilde{a}$, or even got stuck without picking any action. Specifically, PFT chooses action $a^\dagger = \arg\max_a \mathrm{UCB}$ and SITH-PFT's candidate action is $\tilde{a} = \arg\max_{a \in \mathcal{A}} \underline{\mathrm{UCB}}(ha)$. Three different scenarios are possible:

Case 2.2.1. the $\overline{\text{UCB}}, \underline{\text{UCB}}$ bounds over $h\tilde{a}$ were tight enough and $\tilde{a}$ was chosen such that $a^\dagger \neq \tilde{a}$.

Case 2.2.2. SITH-PFT is stuck in an infinite loop. It can happen if the $\overline{\text{UCB}}, \underline{\text{UCB}}$ bounds over $h\tilde{a}$, and at least one of its sibling nodes ha, are not tight enough. However, all tree nodes are at the maximal simplification level. Hence, resimplification is triggered over and over without it changing anything.

Case 2.2.1 is not possible as the bounds are analytical (always true) and converge to the actual reward ($\underline{\text{UCB}} = \text{UCB} = \overline{\text{UCB}}$) for the maximal simplification level. Case 2.2.2 is not possible. If the bounds are not close enough to make a decision, resimplification is triggered. Each time some ha node - sibling to $h\tilde{a}$ and maybe even $h\tilde{a}$ itself is chosen in *SelectBest* to over-go resimplification. According to lemmas 1 and 2, after some finite number of iterations for all of the sibling ha nodes (including $h\tilde{a}$) it holds $\underline{\text{UCB}}(ha) = \text{UCB}(ha) = \overline{\text{UCB}}(ha)$ and some action can be picked. If different actions have identical values we choose one by the same rule UCB picks actions with identical values (e.g. lower index/random). Since Case 2.2.2 is false, after some finite number of resimplification iterations, SITH-PFT will stop with bounds sufficient enough to make a decision; as Case 2.2.1 is false it holds that $a^\dagger = \tilde{a}$. Thus we get a contradiction and the proof is complete. ■

11.5 Proof of Theorem 3

Since same tree is built according to Theorem 2, the only modification is the final criteria at the end of the planning session at the root of the tree: $a^* = \arg\max_a Q(ha)$. Note we can set the exploration constant of UCB to $c = 0$ and we get that UCB is just the Q function. Thus if the bounds are not tight enough at the root to decide on an action, resimplification will be repeatedly called until SITH-PFT can make a decision. The action will be identical to the one chosen by UCB at PFT from similar arguments in the proof of Theorem 2. Note that additional final criteria for action selection could be introduced, but it would not matter as tree consistency is kept according to Theorem 2 and the bounds converge to the immediate rewards and Q estimations. ■

11.6 Proof for Theorem 4

Let us first prove that $u + \hat{\mathcal{H}} \geq 0$. the It holds

$$u + \hat{\mathcal{H}} = \sum_{i \notin A^s_{k+1}} w^i_{k+1} \cdot \log\left[m \cdot \mathbb{P}_O(z_{k+1}|x^i_{k+1})\right] + \tag{71}$$

$$\sum_{i \in A^s_{k+1}} w^i_{k+1} \cdot \log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \sum_{j=1}^{n_x} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k\right] -$$

$$\sum_{i=1}^{n_x} w^i_{k+1} \cdot \log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \sum_{j=1}^{n_x} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k\right] =$$

The Eq. (71) equals to

$$\sum_{i \notin A^s_{k+1}} w^i_{k+1} \cdot \log\left[m \cdot \mathbb{P}_O(z_{k+1}|x^i_{k+1})\right] -$$

$$\sum_{i \notin A^s_{k+1}} w^i_{k+1} \cdot \log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \sum_{j=1}^{n_x} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k\right]$$

Fix arbitrary index $i \notin A^s_{k+1}$. The log is monotonically increasing function so it is left to prove that

$$m\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \geq \mathbb{P}_O(z_{k+1}|x^i_{k+1}) \sum_{j=1}^{n_x} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k$$

If $\mathbb{P}_O(z_{k+1}|x^i_{k+1}) = 0$, we finished. Assume $\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \neq 0$. Recalling the definition $m \triangleq \max_{x',a} \mathbb{P}_T(x'|x,a)$, it holds that

$$\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \sum_{j=1}^{n_x} \max_{x_{k+1}, x_k, a_k} \mathbb{P}_T(x_{k+1}|x_k, a_k) w^j_k \geq \mathbb{P}_O(z_{k+1}|x^i_{k+1}) \sum_{j=1}^{n_x} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k. \tag{72}$$

We reached the desired result. Now let us show the second part $\ell + \hat{\mathcal{H}} \leq 0$. Observe, that

$$0 \geq \ell + \hat{\mathcal{H}} = \tag{73}$$

$$\sum_{i=1}^{n_x} w^i_{k+1} \log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \sum_{j \in A^s_k} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k\right] -$$

$$\sum_{i=1}^{n_x} w^i_{k+1} \log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \sum_{j=1}^{n_x} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k\right]$$

Select arbitrary index i. We shall prove that

$$\log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \sum_{j \in A^s_k} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k\right] -$$

$$\log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \sum_{j=1}^{n_x} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k\right] \leq 0.$$

Again use that log is monotonically increasing and assume that $\mathbb{P}_O(z_{k+1}|x^i_{k+1}) \neq 0$. We have that

$$\sum_{j \in A^s_k} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k - \sum_{j=1}^{n_x} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k = \tag{74}$$

$$- \sum_{j \notin A^s_k} \mathbb{P}_T(x^i_{k+1}|x^j_k, a_k) w^j_k \leq 0$$

■

11.7 Proof for Theorem 5

We first prove that

$$\overline{\Delta}^s(b, a, b') \geq \overline{\Delta}^{s+1}(b, a, b') \geq 0. \tag{75}$$

Recall that from the previous proof equation (71)

$$\overline{\Delta}^s(b,a,b') = \sum_{i\notin A^s_{k+1}} w^i_{k+1}\log\left[m\cdot\mathbb{P}_O(z_{k+1}|x^i_{k+1})\right] - \sum_{i\notin A^s_{k+1}} w^i_{k+1}\log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1})\sum_{j=1}^{n_x}\mathbb{P}_T(x^i_{k+1}|x^j_k,a_k)w^j_k\right]. \tag{76}$$

Suppose we promote the simplification level. Without loss of generality assume that $A^{s+1}_{k+1} = A^s_{k+1}\cup\{q\}$. From the above we conclude that $q\notin A^s_{k+1}$

$$\overline{\Delta}^{s+1}(b,a,b') = \overline{\Delta}^s(b,a,b') - w^q_{k+1}\left(\log\left[m\cdot\mathbb{P}_O(z_{k+1}|x^q_{k+1})\right] - \log\left[\mathbb{P}_O(z_{k+1}|x^q_{k+1})\sum_{j=1}^{n_x}\mathbb{P}_T(x^q_{k+1}|x^j_k,a_k)w^j_k\right]\right) \tag{77}$$

It is left to prove that

$$m\cdot\mathbb{P}_O(z_{k+1}|x^q_{k+1}) \geq \geq \mathbb{P}_O(z_{k+1}|x^q_{k+1})\sum_{j=1}^{n_x}\mathbb{P}_T(x^q_{k+1}|x^j_k,a_k)w^j_k \tag{78}$$

We already done that in previous theorem. Now we prove the second part

$$\underline{\Delta}^s(b,a,b') \geq \underline{\Delta}^{s+1}(b,a,b') \geq 0. \tag{79}$$

The next equation is the minus equation (73)

$$\underline{\Delta}^s(b,a,b') = \sum_{i=1}^{n_x} w^i_{k+1}\log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1})\sum_{j=1}^{n_x}\mathbb{P}_T(x^i_{k+1}|x^j_k,a_k)w^j_k\right] - \sum_{i=1}^{n_x} w^i_{k+1}\log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1})\sum_{j\in A^s_k}\mathbb{P}_T(x^i_{k+1}|x^j_k,a_k)w^j_k\right] \tag{80}$$

Assume again without loosing generality that $A^{s+1}_k = A^s_k\cup\{q\}$. In that case

$$\underline{\Delta}^s(b,a,b') - \underline{\Delta}^{s+1}(b,a,b') = \tag{81}$$

$$-\sum_{i=1}^{n_x} w^i_{k+1}\log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1})\sum_{j\in A^s_k}\mathbb{P}_T(x^i_{k+1}|x^j_k,a_k)w^j_k\right] \tag{82}$$

$$+\sum_{i=1}^{n_x} w^i_{k+1}\log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1})\sum_{j\in A^{s+1}_k}\mathbb{P}_T(x^i_{k+1}|x^j_k,a_k)w^j_k\right]. \tag{83}$$

Select arbitrary index i. We got back to end to previous theorem. Note that by definition the bounds are convergent since we are using all the particles. To see it explicitly suppose that $\{i\notin A^s_{k+1}\} = \emptyset$ and $\{i\notin A^s_k\} = \emptyset$. We have that

$$\overline{\Delta}^s(b,a,b') = \underline{\Delta}^s(b,a,b') = 0. \tag{84}$$

This concludes the proof. ■

11.8 Bounds time complexity analysis

We turn to analyze the time complexity of our method using the chosen bounds (57) and (58). We assume the significant bottleneck is querying the motion $\mathbb{P}_T(x'|x,a)$ and observation $\mathbb{P}_O(z|x)$ models respectively. Assume the belief is approximated by a set of n_x weighted particles,

$$b = \{x^i, w^i\}_{i=1}^{n_x}. \tag{85}$$

Consider the Boers et al. (2010) differential entropy approximation for belief at time $k+1$,

$$\hat{\mathcal{H}}(b_k,a_k,z_{k+1},b_{k+1}) \triangleq \underbrace{\log\left[\sum_{i=1}^{n_x}\mathbb{P}_O(z_{k+1}|x^i_{k+1})w^i_k\right]}_{(a)} + \tag{86}$$

$$\underbrace{\sum_{i=1}^{n_x} w^i_{k+1}\cdot\log\left[\mathbb{P}_O(z_{k+1}|x^i_{k+1})\sum_{j=1}^{n_x}\mathbb{P}_T(x^i_{k+1}|x^j_k,a_k)w^j_k\right]}_{(b)}. \tag{87}$$

Denote the time to query the observation and motion models a single time as t_{obs}, t_{mot} respectively. It is clear from (85), (86) (term a) and, (87) (term b) that:

$$\forall b \text{ as in (85) } \Theta(\hat{\mathcal{H}}(b)) = \Theta(n_x\cdot t_{obs} + n_x^2\cdot t_{mot}). \tag{88}$$

Since we share calculation between the bounds, the bounds' time complexity, for some level of simplification s, is:

$$\Theta(\ell^s + u^s) = \Theta(n_x\cdot t_{obs} + n_x^s\cdot n_x\cdot t_{mot}), \tag{89}$$

where n_x^s is the size of the particles subset that is currently used for the bounds calculations, e.g. $n_x^s = |A^s|$ (A^s is as in (57) and (58)) and ℓ^s, u^s denotes the immediate upper and lower bound using simplification level s. Further, we remind the simplification levels are discrete, finite, and satisfy

$$s\in\{1,2,\ldots,n_{\max}\},\quad \ell^{s=n_{\max}} = -\hat{\mathcal{H}} = u^{s=n_{\max}}. \tag{90}$$

Now, assume we wish to tighten ℓ^s, u^s and move from simplification level s to $s+1$. Since the bounds are updated incrementally (as introduced by Sztyglic and Indelman (2022)), when moving from simplification level s to $s+1$ the only additional data we are missing are the new values of the observation and motion models for the newly added particles. Thus, we get that the time complexity of moving from one simplification level to another is:

$$\Theta(\ell^s + u^s \to \ell^{s+1} + u^{s+1}) = \Theta((n_x^{s+1} - n_x^s)\cdot n_x\cdot t_{mot}), \tag{91}$$

where $\Theta(\ell^s + u^s \to \ell^{s+1} + u^{s+1})$ denotes the time complexity of updating the bounds from one simplification level to the following one. Note the first term from (89), $n_x\cdot t_{obs}$, is not present in (91). This term has nothing to do with simplification level s and it is calculated linearly over all particles n_x. Thus, it is calculated once at the beginning (initial/lowest simplification level).

We can now deduce using (89) and (91)

$$\Theta(\ell^{s+1} + u^{s+1}) = \\ \Theta(\ell^s + u^s) + \Theta(\ell^s + u^s \rightarrow \ell^{s+1} + u^{s+1}). \tag{92}$$

Finally, using (88), (89), (90), (91), and (92), we come to the conclusion that if at the end of a planning session, a node's b simplification level was $1 \leq s \leq n_{\max}$ than the time complexity saved for that node is

$$\Theta((n_x - n_x^s) \cdot n_x \cdot t_{mot}). \tag{93}$$

This makes perfect sense since if we had to resimplify all the way to the maximal level we get $s = n_{\max} \Rightarrow n_x^{s=n_{\max}} = n_x$ and by substituting $n_x^s = n_x$ in (93) we saved no time at all.

To conclude, the total speedup of the algorithm is dependent on how many belief nodes' bounds were not resimplified to the maximal level. The more nodes we had at the end of a planning session with lower simplification levels, the more speedup we get according to (93).

References

Araya M, Buffet O, Thomas V and Charpillet F (2010) A pomdp extension with belief-dependent rewards. In: *Advances in Neural Information Processing Systems (NIPS)*. pp. 64–72. 2

Auer P, Cesa-Bianchi N and Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2): 235–256. 6

Auger D, Couetoux A and Teytaud O (2013) Continuous upper confidence trees with polynomial exploration–consistency. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part I 13*. Springer, pp. 194–209. 6

Barenboim M and Indelman V (2022) Adaptive information belief space planning. In: *the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI)*. 3

Barenboim M and Indelman V (2023) Online pomdp planning with anytime deterministic guarantees. In: *Advances in Neural Information Processing Systems (NIPS)*. 3

Boers Y, Driessen H, Bagchi A and Mandal P (2010) Particle filter based entropy. In: *2010 13th International Conference on Information Fusion*. pp. 1–8. DOI:10.1109/ICIF.2010.5712013. 2, 4, 5, 19, 21, 32

Burgard W, Fox D and Thrun S (1997) Active mobile robot localization. In: *Intl. Joint Conf. on AI (IJCAI)*. Citeseer, pp. 1346–1352. 2

Crisan D and Doucet A (2002) A survey of convergence results on particle filtering for practitioners. *IEEE Trans. Signal Processing* . 2, 28

Dressel L and Kochenderfer MJ (2017) Efficient decision-theoretic target localization. In: Barbulescu L, Frank J, Mausam and Smith SF (eds.) *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS 2017, Pittsburgh, Pennsylvania, USA, June 18-23, 2017*. AAAI Press, pp. 70–78. 2

Egorov M, Sunberg ZN, Balaban E, Wheeler TA, Gupta JK and Kochenderfer MJ (2017) Pomdps. jl: A framework for sequential decision making under uncertainty. *The Journal of Machine Learning Research* 18(1): 831–835. 21

Elimelech K and Indelman V (2022) Simplified decision making in the belief space using belief sparsification. *The International Journal of Robotics Research* 41(5): 470–496. 2, 3

Farhi E and Indelman V (2021) ix-bsp: Incremental belief space planning. *arXiv preprint arXiv:2102.09539* . 11

Farhi EI and Indelman V (2019) ix-bsp: Belief space planning through incremental expectation. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 11

Fehr M, Buffet O, Thomas V and Dibangoye J (2018) rho-pomdps have lipschitz-continuous epsilon-optimal value functions. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N and Garnett R (eds.) *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., pp. 6933–6943. 2

Fischer J and Tas OS (2020) Information particle filter tree: An online algorithm for pomdps with belief-based rewards on continuous domains. In: *Intl. Conf. on Machine Learning (ICML)*. Vienna, Austria. 2, 3, 19, 21, 22, 28

Garg NP, Hsu D and Lee WS (2019) Despot-α: Online pomdp planning with large state and observation spaces. In: *Robotics: Science and Systems (RSS)*. 1, 3

Hoerger M and Kurniawati H (2021) An on-line pomdp solver for continuous observation spaces. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, pp. 7643–7649. 3

Hoerger M, Kurniawati H, Bandyopadhyay T and Elfes A (2020) Linearization in motion planning under uncertainty. In: *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*. Springer, pp. 272–287. 3

Hoerger M, Kurniawati H and Elfes A (2019) Multilevel monte-carlo for solving pomdps online. In: *Proc. International Symposium on Robotics Research (ISRR)*. 3

Hollinger GA and Sukhatme GS (2014) Sampling-based robotic information gathering algorithms. *Intl. J. of Robotics Research* : 1271–1287. 2

Indelman V, Carlone L and Dellaert F (2015) Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments. *Intl. J. of Robotics Research* 34(7): 849–882. 2

Kearns M, Mansour Y and Ng AY (2002) A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine learning* 49(2): 193–208. 3, 11, 21, 22

Kitanov A and Indelman V (2024) Topological belief space planning for active slam with pairwise gaussian potentials and performance guarantees. *Intl. J. of Robotics Research* 43(1): 69–97. DOI:10.1177/02783649231204898. 2, 3

Kochenderfer M, Wheeler T and Wray K (2022) *Algorithms for Decision Making*. MIT Press. 1, 3, 10

Kocsis L and Szepesvári C (2006) Bandit based monte-carlo planning. In: *European conference on machine learning*. Springer, pp. 282–293. 3, 6

Kopitkov D and Indelman V (2017) No belief propagation required: Belief space planning in high-dimensional state spaces via factor graphs, matrix determinant lemma and re-use of calculation. *Intl. J. of Robotics Research* 36(10): 1088–1130. 2

Kopitkov D and Indelman V (2019) General purpose incremental covariance update and efficient belief space planning via factor-graph propagation action tree. *Intl. J. of Robotics Research* 38(14): 1644–1673. 2

Kurniawati H, Hsu D and Lee WS (2008) SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In: *Robotics: Science and Systems (RSS)*. 1

Lev-Yehudi I, Barenboim M and Indelman V (2024) Simplifying complex observation models in continuous pomdp planning with probabilistic guarantees and practice. In: *AAAI Conf. on Artificial Intelligence*. 3

Munos R (2014) *From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning*. 6

Papadimitriou C and Tsitsiklis J (1987) The complexity of Markov decision processes. *Mathematics of operations research* 12(3): 441–450. 1

Pathak S, Thomas A and Indelman V (2018) A unified framework for data association aware robust belief space planning and perception. *Intl. J. of Robotics Research* 32(2-3): 287–315. 2

Platt R, Tedrake R, Kaelbling L and Lozano-Pérez T (2010) Belief space planning assuming maximum likelihood observations. In: *Robotics: Science and Systems (RSS)*. Zaragoza, Spain, pp. 587–593. 2

Shienman M and Indelman V (2022) Nonmyopic distilled data association belief space planning under budget constraints. In: *Proc. of the Intl. Symp. of Robotics Research (ISRR)*. 3

Silver D and Veness J (2010) Monte-carlo planning in large pomdps. In: *Advances in Neural Information Processing Systems (NIPS)*. pp. 2164–2172. 1, 3

Smith T and Simmons R (2004) Heuristic search value iteration for pomdps. In: *Conf. on Uncertainty in Artificial Intelligence (UAI)*. pp. 520–527. 1, 3

Spaan MT, Veiga TS and Lima PU (2015) Decision-theoretic planning under uncertainty with information rewards for active cooperative perception. *Autonomous Agents and Multi-Agent Systems* 29(6): 1157–1185. 2

Stachniss C, Grisetti G and Burgard W (2005) Information gain-based exploration using Rao-Blackwellized particle filters. In: *Robotics: Science and Systems (RSS)*. pp. 65–72. 2

Sunberg Z and Kochenderfer M (2018) Online algorithms for pomdps with continuous state, action, and observation spaces. In: *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28. 1, 3, 6, 16, 17, 21, 28

Sutton RS and Barto AG (2018) *Reinforcement learning: An introduction*. MIT press. 6

Sztyglic O and Indelman V (2022) Speeding up online pomdp planning via simplification. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 3, 4, 32

Thrun S, Burgard W and Fox D (2005) *Probabilistic Robotics*. The MIT press, Cambridge, MA. 22

Van Den Berg J, Patil S and Alterovitz R (2012) Motion planning under uncertainty using iterative local optimization in belief space. *Intl. J. of Robotics Research* 31(11): 1263–1278. 2

Walsh T, Goschin S and Littman M (2010) Integrating sample-based planning and model-based reinforcement learning. In: *AAAI Conf. on Artificial Intelligence*, volume 24. 3

Ye N, Somani A, Hsu D and Lee WS (2017) Despot: Online pomdp planning with regularization. *JAIR* 58: 231–266. 1, 3

Zhitnikov A and Indelman V (2022a) Risk aware adaptive belief-dependent probabilistically constrained continuous pomdp planning. *arXiv preprint arXiv:2209.02679* . 21

Zhitnikov A and Indelman V (2022b) Simplified risk aware decision making with belief dependent rewards in partially observable domains. *Artificial Intelligence, Special Issue on "Risk-Aware Autonomous Systems: Theory and Practice"* . 3, 7

Zhitnikov A and Indelman V (2024) Simplified continuous high dimensional belief space planning with adaptive probabilistic belief-dependent constraints. *IEEE Trans. Robotics* . 3

Chapter 4

Unpublished Material

Safe Belief-dependent Probabilistically-constrained and Chance-constrained Continuous Approximate Adaptive POMDP Planning*

Andrey Zhitnikov[1], Vadim Indelman[2]
[1]Technion Autonomous Systems Program (TASP)
[2]Department of Aerospace Engineering
Technion - Israel Institute of Technology, Haifa 32000, Israel
andreyz@campus.technion.ac.il, vadim.indelman@technion.ac.il

August 13, 2024

Abstract

Although safety is fundamental to an online operating agent, it has received less attention in the challenging continuous domain and under partial observability. This paper presents a novel formulation and solution for risk-averse belief-dependent probabilistically-constrained continuous POMDP. We tackle a demanding setting of belief-dependent reward and constraint operators. Our Probabilistic Constraint is belief-dependent and has two conditions. The internal condition thresholds the belief-dependent operator given a future possible history of actions and observations simulated in a planning session, while the external one operates on the level of histories and thresholds the probability that the internal condition is satisfied, stemming from the distribution of future histories (decision epochs). We rigorously analyze our formulation versus the Chance Constraint in the Closed and Open Loop setting. In the Closed Loop setting, we revealed that Chance Constraint is a special case of our Probabilistic Constraint. In the Open Loop setting, the two approaches are essentially different since the Chance Constraint enforces the condition over the underlying MDP. In contrast, Probabilistic Constraint do not assume complete observability in planning session. Moreover, the Chance Constraint does not accommodate general belief-dependent operators. We uplift the chance-constrained approach to continuous environments and belief-dependent rewards. For probabilistically-constrained planning, we contribute adaptive, in terms of observation episodes laces and beliefs within the lace, algorithms. For chance-constrained planning, we contribute an adaptive, with respect to state trajectories and states within the trajectory, algorithm. All our proposed algorithms can be used with parametric and nonparametric beliefs represented by particles and in continuous domains in terms of states and observations. The simulations demonstrate that in the setting of policies (Closed Loop), our Probabilistic Constraint allows much faster evaluation compared to the chance-constrained formulation, with the same performance in terms of collisions. In the setting of static action sequences (Open Loop), we show that the two formulations yield a very similar number of collisions, but Chance Constraint appears to be faster than Probabilistic Constraint.

Keywords— Decision making under Uncertainty, Belief Space Planning, Belief-dependent POMDP, Planning with Incomplete Information, Belief-dependent rewards, Belief-dependent Probabilistic Constraints

1 Introduction and Related Work

Decision making under uncertainty in partially observable domains is a key capability for reliable autonomous agents. Commonly, the basis of the State Of The Art (SOTA) algorithms in such a setting is the Partially Observable Markov Decision Process (POMDP). The robot does not have access to the POMDP state. Instead, it maintains a distribution, named the (posterior) belief, over the state given all its current information, namely, the history of actions and the observations alongside the prior belief. The decision maker shall maintain and reason about the evolution of the belief within the planning phase. At the same time, the robot's online goal is to find an optimal action for its current belief. Unfortunately, an exact solution of POMDP is unfeasible [27]. A critical limitation of the

*This work was partially supported by the Israel Science Foundation (ISF).

classical POMDP formulation is the assumption that the belief-dependent reward is nothing more than the expectation of state-dependent reward with respect to the corresponding belief [20]. Another limiting assumption in many SOTA algorithms is the discrete domain, e.g., discrete state and observation spaces [32], [37]. The main problem in continuous and infinite or large discrete spaces is that one can not go over the entire observation space and calculate the probabilities of the observations as in [30]. Therefore, all tabular methods are inappropriate for these domains. Instead one needs to resort to sampling from the likelihood of observations given the previous belief and an action. Then one samples from continuous density the same observation can be received with probability zero such that the extension of tabular methods, e.g. [30], to continuous domains requires clarification. In this work, we aim to tackle this crucial gap. Specifically, our theory and algorithms accommodate continuous domains in terms of state and observation spaces.

1.1 Belief-dependent Rewards

Augmenting POMDP with general belief-dependent rewards is a long-standing problem. Unraveling it would allow information-theoretic rewards, which are extremely important in numerous problems in Artificial Intelligence (AI) and Robotics, such as Autonomous Exploration, Informative Planning, Information Gathering [38], Belief Space Planning (BSP) [16], and active Simultaneous Localization and Mapping (SLAM) [28]. The belief-dependent reward formulation is known as ρ-POMDP [2], [9]. Earlier techniques focused on offline solvers and extended α-vectors approach to piecewise linear and convex [2], [8] or Lipschitz-continuous rewards [9]. These extended solvers are also limited to discrete domains in terms of states and observations.

Another way to incorporate general belief-dependent rewards is to reformulate POMDP as Belief-MDP (BMDP) and use more recent online solvers designed for MDP. These algorithms build a belief tree and are suitable for continuous domains and challenging nonparametric beliefs represented by particles. Seminal approaches in this category are Sparse Sampling (SS) [18], Monte Carlo Tree Search (MCTS) [33], and its efficient, simplified variant [39]. Such an MCTS running on BMDP is called a Particle Filter Tree with Double Progressive Widening (PFT-DPW) [33]. Progressive Widening handles the problem of shallow trees in continuous setting arising due to the inability to sample twice the same action and observation.

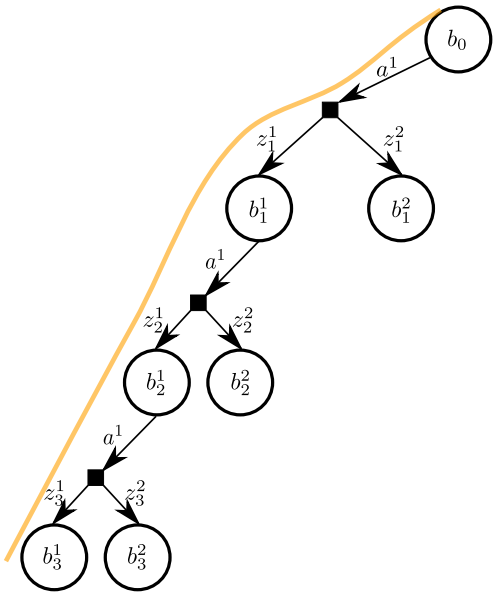


Fig. 1: Illustration of a belief tree. The thick yellow lace depicts a sequence of beliefs $(b_0, b_1^1, b_2^1, b_3^1)$ generated by a corresponding sequence of observations under some policy (Section 3).

1.2 Constrained POMDPs

In this section, we begin by describing the Chance Constraint (CC) and then we talk about other formulations of constraints for POMDP.

Chance-constrained Approaches The motivation to add the CC is to introduce the notion of *risk* and *safety* into the problem. Initially, the planning community focused on collision avoidance, formulating it as a CC. For example, [6] converts the belief tree to a graph in belief space. This conversion utilizes a stabilizing controller to shift the expectation of the posterior belief to the nominal value. Further, it uses a Kalman filter on top of linearized additive models with Gaussian noise. In this case, the covariance matrix of the belief does not depend on the actual observation but depends on the covariance of the observation model. The authors assume that the map has areas of large and low variance of measurement noise. Over the path in the graph, [6] suggests a way to track the state estimate which is the expectation of the posterior belief. The state estimate has variability because the stabilizing controller is imperfect and has an error. All in all, the CC in [6] is calculated with respect to the application of the controller at each time instance. While being seminal and important [6] suffers from many limiting assumptions. Let us mention a few. The variability of the state estimate is taken into account only in the CC and not in the reward. The dependence of the observation model covariance matrix on the robot state is not considered in the belief update. The approach does not apply to general beliefs represented by particles and applies only to Gaussian beliefs and additive linear observation models with Gaussian noise not depending on the robot state. The linearity of the models can be handled by linearization. Nevertheless, this work can be applied in continuous domains.

The authors of [30] present another approach to CC for discrete state and observation spaces and the setting of deterministic policies. The paper [30] introduces the algorithm RAO* which uses admissible heuristics for the action value function (Q-function) in the belief space. This aspect is problematic with **general belief-dependent rewards**. However, this is not the main point of our work. Moreover, the RAO* algorithm prunes not feasible actions using only a necessary condition of the feasibility of CC (See Appendix E). In other words, the CC may be violated, but the action

has not been pruned. In fact, this is one of our main points in this paper. Since the condition is only necessary, after pruning it is still required to verify the feasibility of each kept action. This significantly complicates the solution and inflicts unnecessary computational burden. It would be much easier if after pruning we would know that remained actions satisfy the constraint. Another aspect of RAO* is disregarding the discrepancy in the belief definition for the CC and the reward calculations. This claim will be clear shortly. Although this discrepancy is taken into account in observation likelihood, with respect to beliefs themselves such a discrepancy is ignored in [30].

The CC formulation appearing in[30] was also extended to the notion of durative actions [19]. The paper [19] presents algorithms solely for discrete domains. Although it is interesting to extend to continuous domains the Stochastic duration with percentile risk criteria and Chance-constrained duration, we left it for future work. These formulations appear close to our cumulative form of inner constraint, to be defined shortly.

Another recent paper [26] utilizes CC within the MCTS. They train Neural Networks (NN) for an initial estimate of Execution Risk (ER) [30] under a stochastic future policy. Utilizing adaptive conformal inference, they have an adaptive approach for future CC thresholds to assure feasibility at the corresponding future belief node. Their threshold is adaptive with respect to each update of the estimator of execution risk. If the current action under consideration is not feasible with regard to updated execution risk, they widen and, if feasible, tighten the threshold. Still, it is not ensured that it will be a feasible action at the root due to the usage of the recursive Bellman optimality principle. Moreover, since the planning sessions have recursive dependence, the CC not enforced from each belief in the belief tree with the same threshold is suboptimal. When the Bellman recursive approach is used, CC requires a rule for determining future CC thresholds to assure at least a single feasible, with respect to CC at the root, action. In contrast, our approach handles that by design as we will further see.

Other Formulations of the Constraint Some works consider an averaged cumulative constraint [36],[21]. Another interesting and related work [3] suggests ϵ-shadows. That work assumes a fully observable deterministic robot motion over the map populated by uncertain obstacles. The robot receives a stream of observations from the obstacles and reasons about obstacle locations using geometric confidence intervals named ϵ-shadows. Unlike that paper, we operate in belief space and model decision-making under uncertainty as POMDP. Moreover, similar to ϵ-shadows, we can inflate obstacles, to increase robustness. Recently, the shielded POMDP formulations have appeared [1], [25]. While appearing similar to our formulation with multiplicative form, to be defined shortly, these works are in discrete domains in terms of states and observations. Moreover, we contribute a rigorous formulation of the problem and the constraints, both in continuous domains, using the indicator function and relevant sets.

Analogous to the situation with belief-dependent rewards, reformulation of POMDP as BMDP can possibly allow employing approaches designed for probabilistically-constrained MDP [12], [10]. However, the theory presented in these papers does not apply to parametric or nonparametric BMDP due to various assumptions made by the authors. This is one of the gaps we aim to fill in this work. Typically algorithms designed for general beliefs represent the belief as a set of particles and use a Particle Filter (PF) [35] for nonparametric Bayesian update. In this work, we assume the setting of nonparametric beliefs, although our formulations also support a parametric setting.

To conclude, the closest to our formulation is CC. Therefore, our comparison will be centered around Probabilistic Constraint (PC) versus CC. In this paper, we ask the question of how the CC is different from our novel PC.

1.3 CC Accommodation to Belief-dependent Rewards and Continuous Domains

Having established that the closest to our formulation is CC, we now turn to the question of employing existing chance-constrained methods in continuous domains and in conjunction with belief-dependent rewards. The work [6] is by definition in continuous domains with belief-dependent reward. However, the belief shall be parametric Gaussian to employ the Kalman Filter. The approach presented in [30] is a tabular method; therefore, it is not clear how to extend it to continuous domains. Moreover, the belief-dependent reward will complicate the finding of the heuristics for the objective. The work [26] can be used with belief-dependent rewards and in continuous domains. This is, however, an MCTS based method with learned components and a stochastic future policy. Here we focus on deterministic policies, similar to [30]. Moreover, the convergence of MCTS with unbounded rewards is under the question mark, whereas all the algorithms presented in this paper converge as the number of simulated observation episodes and belief particles grows.

1.4 Comparison to Chance Constraint

Since one of our goals in this paper is to compare in terms of quality and celerity our suggested PC with CC, we now outline the prominent aspects of both approaches. In this paper, we focus on safety aspects. Most works tackling constrained online planning under uncertainty, in this context, utilize the chance-constrained formulation [6], [30]. This formulation is regarded as SOTA. By design, the CC is defined over the future states [6] or trajectories [30] considered in the planning session, given the belief at the beginning of the planning session and the candidate policy. In contrast, we develop our PC on the level of posterior beliefs. This, by definition, allows the utilization of general belief-dependent operators. As we further show in Section 2.3, the CC formulation from [6] is a particular case of the one used in [30].

In a fully observable setting of MDP, where the action is predefined or is a function of the state as in QMDP [20], CC motivation is clear. It thresholds the probability that future trajectories will be safe. This probability is accessible using the motion model. However, under partial observability (POMDP), generally, the action is a function of the belief (in case of deterministic policies). Therefore, there are two cases to consider here, Closed Loop (CL) where one deals with policies and the Open Loop (OL) where one deals with predefined candidate action sequences.

Closed Loop CC In this setting, the likelihood of future trajectories implicitly depends on future observations/beliefs. As we show in Section 5, transferring CC from MDP to POMDP is essentially the **averaging** the probabilities of the safe event given future posterior belief, when the belief is defined in a way that accounts for the safe events in the past. In addition, CC does not accommodate general belief-dependent operators. This, however, can be relaxed using our reformulation, as we will further see in Section 5.

The authors of [30] enforce the CC with different threshold levels starting from each non-terminal belief in the belief tree. In [30] only at the root, the CC is enforced with the given in planning session threshold. Inflicting the CC (with a shorter horizon due to the finite depth of the belief tree) from each non-terminal belief is essential in the case of candidate policies. If the CC is imposed only at the root of the belief tree, due to Bellman optimality down the tree, it is very hard to obtain a feasible action at the root. We verified this in simulations. One way or another some thresholds of CC enforced from future beliefs are required. It will be apparent later that, if we enforce CC with the same threshold from each belief and until the predefined horizon, then implicitly CC thresholds non-terminal posterior beliefs with a possibly much larger threshold, but not larger than one. Last but not least, by examining the CC on the level of posterior beliefs, we observe that only the safe portion of the belief is pushed forward to the future time with action and observation. In other words, chance-constrained POMDP has disparate definitions of future beliefs and different distributions of future observations for rewards and the CC. We delve into this aspect in Section 5. This fact significantly complicates the algorithmics in discrete and continuous spaces and renders the chance-constrained approach computationally intense. Moreover, the [30] disregards the mismatch of the definition of beliefs in the CC belief tree and the reward belief tree. It shall be noted that the discrepancy in the likelihood of observations is taken into account by [30] but in discrete spaces only. We extend the treatment of the discrepancy of the likelihood of the observations to **continuous spaces** (See Alg. 3) and add the treatment of the mismatch in the beliefs. We extensively debate this claim in the paper.

	PC	CC
CL ($\pi_{k:k+L-1}$)	faster	slower
OL ($a_{k:k+L-1}$)	slower	faster

Fig. 2: Illustration of celerity of suggested approaches.

Closed Loop PC Instead of enforcing the CC from each belief in the belief tree and until the horizon, we apply a general belief-dependent operator on each corresponding belief. As we will further see, in the stiffest outer threshold case (ϵ=0, Alg. 1), due to its recursive nature, our constraint is automatically enforced from each belief in the search tree exhibiting optimal substructure [13] property similar to CC. By definition, in our setting, the unsafe portion of the belief is also updated with action and observation, such that unsafe states can be pushed forward in time if such an action is not discarded. This way, we have an identical distribution of future observations for belief-dependent rewards and constraints as well as the definition of the beliefs themselves. As we further see in simulations in Section 8, this is highly beneficial in terms of time efficiency.

Open Loop CC and PC In the case of predefined static action sequences, CC thresholds the probability to be safe of possible POMDP future trajectories of states. Due to the fact that the candidate action sequence is predefined, one can constrain the MDP distribution of trajectories by applying a motion model on particles sampled from prior belief. Instead of a safe trajectory of the future states in CC (Alg. 6), we in PC have a safe trajectory of future beliefs (Alg. 2). The major difference is that in chance-constrained formulation one constrains possible MDP states assuming perfect observability and we, in PC, constrain the beliefs without the usage of state selected for future observation creation.

To conclude this section, in Fig. 2, we pictorially summarize the celerity of PC and CC approaches in the setting of CL (policies) and OL (static action sequences).

1.5 Contributions

In this paper, we innovate a technique to enhance continuous belief-dependent POMDP with a belief-dependent Probabilistic Constraint (PC). Our constraint is two-staged. We have an internal threshold applied to the belief-dependent operator (given a history) and an external threshold used for the probability originating from future observations episodes (histories). Surprisingly, in Section 5.7 we unveiled that in the CL setting our PC generalizes CC. We extensively study the interplay between PC and CC in Section 5. Moreover, a general belief-dependent PC was not studied nor proposed. Nevertheless, such a constraint is of the highest importance. For instance, as we discuss in [38], such a formulation can be used to determine when to stop exploration, e.g. in an active SLAM context, which is an open problem currently [7], [28]. In the context of safety belief-dependent operators such that Value at Risk (VaR) and Conditional Value at Risk

(CVaR) quantify what happens in case of the collision, in other words measure how bad the collision will be. Moreover, the CC itself is a belief-dependent operator (Section 5.7).

The preceding discussion leads us to the contributions of this paper. We detail them below in the order they appear in the article.

- Firstly, in Section 3.1, we formulate a risk-averse belief-dependent Probabilistically Constrained continuous POMDP. Averaging the state-dependent reward/constraint to obtain the belief-dependent reward/constraint is a severe hindrance that we relax. We are unaware of prior works addressing POMDP with risk-averse belief-dependent constraints. In particular, our probabilistic belief-dependent constraint supports risk-averse operators, such as CVaR, and leads to a novel safety constraint formulation.
- Secondly, on top of our probabilistic formulation, in Section 4.1.3, we contribute a novel, efficient actions-pruning mechanism. SOTA pruning technique proposed by [30] constitutes only a necessary condition such that it is possible that after pruning, actions violating the CC are kept in the belief tree. Therefore, the feasibility of CC has to still be inspected for each not-pruned action. On the contrary, our pruning condition is necessary and sufficient. No additional checks are needed after the pruning of the belief tree is complete.
- In Section 4.2, we contribute algorithms for online solutions of Probabilistically constrained belief-dependent POMDP in continuous domains. Our algorithms are adaptive given a budget of observation episodes laces (Fig. 1) and beliefs within the lace to expand in the belief tree. In other words, we provide a way to guide the belief tree construction while planning. Our framework is universal for challenging continuous domains and can be applied in nonparametric and parametric settings. We innovate algorithms for CL setting with policies as well as for OL setting with candidate action sequences.
- Another contribution on our end is a rigorous analysis of our probabilistic formulation versus chance-constrained in Section 5. Despite recent algorithmic developments [30], there has been relatively little effort devoted to the theoretical aspects of Chance-constrained continuous belief-dependent POMDP. Surprisingly, in Section 5.7 we obtained that in the CL setting, CC is a specific case of our PC when the belief-dependent operator is CC itself. It shall be noted as a contribution that we spotted the fact that belief shall be defined differently within CC. To the best of our knowledge, no paper addresses this discrepancy.
- We uplift a chance-constrained solver to continuous domains in terms of states and observations and general belief-dependent rewards through Importance Sampling (IS) in Section 6.
- In an OL setting, we contribute an adaptive, in terms of trajectories and states, algorithm (Alg. 6) for chance-constrained continuous ρ-POMDP. This algorithm can be used with exceptionally long horizons and a high dimensional setting.
- We present a detailed and comprehensive study of nonparametric collision avoidance.

1.6 Paper Layout

The rest of this paper is organized as follows. We start from preliminaries in Section 2. We then define our novel framework in Section 3, and give relevant examples of possible constraints in Section 3.3. Next, in Section 4 we adaptively evaluate the PC while constructing the belief tree and present online algorithms (Section 4.2) for our novel formulation. Further, we rigorously analyze the conventional CC in Section 5 and compare it to our PC. Finally, in Section 6 we introduce online solvers for chance-constrained POMDP in a continuous setting augmented with belief-dependent rewards. Section 7 is devoted to the objective modification. Eventually, Section 8 shows simulations and results. The conclusions and final remarks are presented in Section 9. To allow fluid reading, we placed the proofs for all theorems and lemmas, and additional in-depth discussions in the appendix.

2 Preliminaries

We now turn to the definition of Belief-dependent POMDP known as ρ-POMDP. We then discuss existing CC formulations in the setting of POMDP. Let us start with notations.

2.1 Notations

By the letter $\mathbb{P}$, we denote the Probability Density Function (PDF), and by P the probability. By lowercase letters, we denote the random quantities or the realizations depending on the context. For any set A, the $\mathbf{1}_A(\cdot)$ is the indicator function defined as $\mathbf{1}_A(\square)=1$ iff $\square\in A$ and $\mathbf{1}_A(\square)=0$ iff $\square \notin A$. To rephrase that, the indicator function of the set A is the Iverson bracket of the property of belonging to A; that is, $\mathbf{1}_A(\square)=[\square\in A]$. The values in $\square$ can be replaced by one of the respective options. The $\mathbf{1}_A$ is the Bernoulli random variable such that $\mathrm{P}(A)=\mathrm{P}(\mathbf{1}_A=1)$. For better readability we will use interchangeably the notation of $\mathrm{P}(A)$ and $\mathrm{P}(\mathbf{1}_A=1)$. By $\hat{\square}$, we denote estimated values.

2.2 Belief-dependent POMDP (ρ-POMDP)

The ρ-POMDP is formed by a tuple $\langle \mathcal{X}, \mathcal{A}, \mathcal{Z}, \mathrm{T}, \mathrm{O}, \rho, \gamma, b_0 \rangle$ where $\mathcal{X}, \mathcal{A}, \mathcal{Z}$ denote state, action, and observation spaces with $x \in \mathcal{X}$, $a \in \mathcal{A}$, $z \in \mathcal{Z}$ the momentary state, action, and observation, respectively. $\mathrm{T}(x', a, x) \triangleq \mathbb{P}_{\mathrm{T}}(x'|x, a)$ is a stochastic transition model from the past state x to the subsequent x' through action a, $\mathrm{O}(z, x) \triangleq \mathbb{P}_{\mathrm{O}}(z|x)$ is the stochastic observation model, $\gamma \in (0, 1]$ is the discount factor, b_0 is the belief over the initial state (prior), and ρ is the belief-dependent reward operator. Let $h_k \triangleq \{b_0, a_{0:k-1}, z_{1:k}\}$ be a history, of actions $a_{0:k-1}$ and observations $z_{1:k}$ alongside the prior belief b_0, obtained by the agent up to time instance k. The posterior belief b_k is a shorthand for the PDF of the state given all information up to the current time index $b_k(x_k) \triangleq \mathbb{P}(x_k|h_k)$. Similar to $b(x)$, we, sometimes, will write $b(h)$ to index the position in the belief tree by h. Importantly, the belief can be switched with history when conditioned upon. When the agent performs an action and receives an observation, it shall update its belief from b to b', such that $b' = \psi(b, a, z')$. The exact Bayesian belief update reads

$$\mathbb{P}(x'|b, a, z') = \frac{\mathbb{P}(z'|b,a,x')\mathbb{P}(x'|b,a)}{\mathbb{P}(z'|b,a)} = \frac{\mathbb{P}_{\mathrm{O}}(z'|x') \int_{\xi} \mathbb{P}_{\mathrm{T}}(x'|\xi,a) b(\xi) \mathrm{d}\xi}{\int_{\xi'} \mathbb{P}_{\mathrm{O}}(z'|\xi') \int_{\xi} \mathbb{P}_{\mathrm{T}}(\xi'|\xi,a) b(\xi) \mathrm{d}\xi \mathrm{d}\xi'}. \tag{1}$$

In our context, it will be a PF since we focus on the setting of nonparametric beliefs. However, this is not an inherent limitation of our approach. Any belief update method would be suitable. The policy in this paper is a deterministic mapping, indexed by the time instances, from belief to action to be executed $\pi_k : \mathcal{B} \mapsto \mathcal{A}$, where $\mathcal{B}$ is the space of all the beliefs taken into account in the problem. The policy for L consecutive steps ahead is denoted by $\pi_{k:k+L-1}$ and means the sequence of functions $(\pi_\ell)_{\ell=k}^{k+L-1}$. Sometimes we will omit the time indices for clarity and write π or $\pi_{(k+1)+}$. We hope the time indices will be evident from the context.

When an information-theoretic reward, for instance, Information Gain (IG), is introduced to the problem, the reward can assume the following form $\rho(b, a, b') = (1-\lambda)\rho^x(b, a, b') + \lambda \rho^I(b, b')$. In this case, it is a function of two subsequent in time beliefs and an action in between. Note that in the setting of nonparametric beliefs, we shall resort to sampling approximations using m_x samples of the belief. Such a reward is comprised of the expectation over the state and action dependent reward

$$\rho^x(b, a, b') = \mathbb{E}_{x \sim b}[r(x, a)] \approx {}^1\!/_{m_x} \textstyle\sum_{i=1}^{m_x} r(x^i, a), \quad \text{or} \quad \rho^x(b, a, b') = \mathbb{E}_{x' \sim b'}[r(a, x')] \approx {}^1\!/_{m_x} \textstyle\sum_{i=1}^{m_x} r(a, x'^{,i}), \tag{2}$$

weighted by $1-\lambda$ and the information-theoretic reward $\rho^I(\cdot)$ weighted by λ, which in general can be dependent on consecutive beliefs and the elements relating them (e.g. IG estimator from the particle based belief [5]). The online decision making goal at time instance k is to find an action a_k to execute, maximizing the action value function

$$Q^\pi(b_k, a_k; \rho) = \mathbb{E}_{z_{k+1}}\big[\rho(b_k, a_k, b_{k+1}) + V^\pi(b_{k+1}; \rho) \big| b_k, a_k\big], \tag{3}$$

where π is the execution policy or belief tree policy and the value function

$$V^\pi(b_k; \rho) = \mathbb{E}_{z_{k+1:k+L}}\big[\textstyle\sum_{\ell=k}^{k+L-1} \rho(b_\ell, \pi_\ell(b_\ell), b_{\ell+1}) \big| b_k, \pi\big], \tag{4}$$

is expected cumulative reward under the particular policy π. For better readability we explicitly denote the dependence of (3) and (4) on belief-dependent operator ρ. In the online decision making the future belief tree policy $\pi_{(k+1)+}$ is calculated as part of the decision making process. We denote the best future policy by $\pi^*_{(k+1)+}$. The online best current time policy is given by $\pi_k(b_k) = \arg\max_{a_k \in \mathcal{A}} Q^{\pi^*}(b_k, a_k; \rho)$. Further, with slight abuse of notation, to properly denote place of estimates $\hat{Q}$ in belief tree we switch to dependence on history. In this paper we also consider the OL setting. In this setting instead of policy (4) depends on static action sequence of the length L denoted by $a_{k:k+L-1}$ or in short a_{k+}. In this case, we will denote the value as $V(b_k, a_{k+}; \rho) = \mathbb{E}_{z_{k+1:k+L}}\big[\sum_{\ell=k}^{k+L-1} \rho(b_\ell, a_\ell, b_{\ell+1}) \big| b_k, a_{k+}\big]$. The future belief simulated in planning session is defined as

$$b_\ell(x_\ell) = \mathbb{P}(x_\ell|b_k, a_{k:\ell-1}, z_{k+1:\ell}) = \mathbb{P}(x_\ell|h_\ell) = \mathbb{P}(x_\ell|b_{\ell-1}, a_{\ell-1}, z_\ell), \tag{5}$$

where b_k is the input to the planner. Moreover, we define a propagated belief b'^- as the belief b after the robot performed an action a and before it received and observation.

$$b^-_\ell(x_\ell) = \mathbb{P}(x_\ell|b_k, a_{k:\ell-1}, z_{k+1:\ell-1}) = \mathbb{P}(x_\ell|h_{\ell-1}, a_{\ell-1}) = \mathbb{P}(x_\ell|b_{\ell-1}, a_{\ell-1}). \tag{6}$$

Having presented a fundamental stochastic process, we make an overview of the existing CC formulations.

2.3 Chance-constrained ρ-POMDP

The CC in [6] can be written as

$$\Big(\textstyle\prod_{\ell=k}^{k+L} \mathbf{1}_{\{\mathrm{P}(\mathbf{1}_{\{x_\ell \in \mathcal{X}_\ell^{\mathrm{safe}}\}} = 1 | b_k, \pi_{k:\ell-1}) \geq \delta\}}(b_k, \pi)\Big) = 1, \tag{7}$$

where $\mathcal{X}_\ell^{\text{safe}}$ is the safe part of the robot workspace. Whenever, (7) equals to one, the CC is satisfied. To avoid confusion let us reiterate that the notation of the set $\{\mathrm{P}(\mathbf{1}_{\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}}=1|b_k,\pi)\geq\delta\}$ is a shorthand for $\{b_k,\pi{:}\mathrm{P}(\mathbf{1}_{\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}}=1|b_k,\pi)\geq\delta\}$. To look at the indicator from a slightly different angle we also can write $\mathbf{1}_{\{\mathrm{P}(\mathbf{1}_{\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}}=1|b_k,\pi_{k:\ell-1})\geq\delta\}}(\mathrm{P}(\mathbf{1}_{\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}}=1|b_k,\pi_{k:\ell-1}))$. For a thorough discussion about the Indicator variable please refer to Section 2.1. In each time index ℓ, (7) is concerned to be safe solely at the current time index. As we further show, due to the **dependence of the policy on the beliefs**, each indicator in the product in Eq. (7) thresholds the averaged probability of safe event $\{x\in\mathcal{X}^{\text{safe}}\}$ given posterior belief at a corresponding time index. The controller used in [6] can be seen as tweaking the belief update operator ψ. It of course improves the situation by moving the expectation of the sibling beliefs to be closer to each other. The authors of [30] utilize the following form of CC

$$\mathbf{1}_{\left\{\mathrm{P}(\mathbf{1}_{\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}}=1|b_k,\pi_{k:k+L-1})\geq\delta\right\}}(b_k,\pi)=1, \tag{8}$$

where $\tau_k\triangleq x_{k:k+L}$ is the trajectory of the current and future states. The CC represented by (8) has never been investigated, to the best of our knowledge, in a continuous setting and in conjunction with belief-dependent rewards. Observe also that the probability in (8) can be written as $\mathrm{P}(\{x_k\in\mathcal{X}_k^{\text{safe}}\}|b_k)\prod_{\ell=k}^{k+L}\mathrm{P}(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}|b_k,\pi,\bigcap_{i=k}^{\ell-1}\{x_i\in\mathcal{X}_i^{\text{safe}}\})$, using the chain rule backward in time, leading to the conclusion that the probability in (8) is more meaningful than probabilities in (7) because (7) discards the dependence on safe events. If $\big(\mathrm{P}(\{x_k\in\mathcal{X}_k^{\text{safe}}\}|b_k)\prod_{\ell=k}^{k+L}\mathrm{P}(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}|b_k,\pi,\bigcap_{i=k}^{\ell-1}\{x_i\in\mathcal{X}_i^{\text{safe}}\})\big)\geq\delta$, each multiplicand is larger or equal δ and this is exactly the condition (7) without condition on safe events.

After scrutinizing into existing CC formulations in the POMDP setting, we are ready to introduce our novel two-staged approach.

3 Introducing Probabilistic Belief-dependent Constraints

Further we formulate the problem and in due course give examples of possible belief-dependent constraints.

3.1 Problem Formulation

We add to the ρ-POMDP tuple, described in Section 2.2, an additional belief-dependent operator ϕ and obtain

$$\langle \underbrace{\mathcal{X}}_{\text{continuous}}, \mathcal{A}, \underbrace{\mathcal{Z}}_{\text{continuous}}, \mathrm{T}, \mathrm{O}, \underbrace{\rho}_{\substack{\text{belief}\\ \text{dependent}}}, \underbrace{\phi}_{\substack{\text{belief}\\ \text{dependent}}}, \gamma, b_0\rangle.$$

Next, we introduce a new problem with the following objective, in the setting of policies,

$$a_k^*\in\arg\max_{a_k\in\mathcal{A}}Q^{\pi^*}(b_k,a_k;\rho)\quad\text{subject to} \tag{9}$$

$$\mathrm{P}\big(\underbrace{c(b_{k:k+L};\phi,\delta)=1}_{\substack{\text{inner}\\ \text{constraint}}}|b_k,a_k,\pi^*_{k+1:k+L-1}\big)\geq 1-\epsilon,\ \texttt{PC} \tag{10}$$

where $c\in\{0,1\}$ is a Bernoulli random variable. By π^* we denote the belief tree policy defined by the planning algorithm. The operators $\rho(\cdot)$ and $\phi(\cdot)$ are general and belief-dependent. Note that one can select the same operator for both. Further, we will regard the PC (10) as the outer or external constraint operating on the level of distribution of future histories. It requires two parameters, ϵ, and δ. The former, $\epsilon\in[0,1)$, is the probability margin within which we permit to the future contingencies, rendered by possible future observations episodes generating the beliefs (see Fig. 1), violate the inner constraint, in other words, to be unprofitable or unsafe. The parameter δ is the margin for some particular episode of the beliefs $b_{k:k+L}$. With the probability of at least $1-\epsilon$, we want the received sequence of the current and future posterior beliefs $b_{k:k+L}$ to fulfill the inner constraint. The inner or internal constraint $c(b_{k:k+L};\phi,\delta)=1$ can be of two forms, the cumulative and the multiplicative

$$c(b_{k:k+L};\phi,\delta)\triangleq\mathbf{1}_{\left\{b_{k:k+L}:b_{k:k+L}\in\mathcal{B}_{k:k+L},\left(\sum_{\ell=k}^{k+L-1}\phi(b_\ell,b_{\ell+1})\right)>\delta\right\}}(b_{k:k+L}),\quad\texttt{cumulative flavor} \tag{11}$$

$$c(b_{k:k+L};\phi,\delta)\triangleq\prod_{\ell=k}^{k+L}\mathbf{1}_{\{b_\ell:b_\ell\in\mathcal{B}_\ell,\phi(b_\ell)\geq\delta\}}(b_\ell),\quad\texttt{multiplicative flavor} \tag{12}$$

where ϕ denotes a general belief-dependent operator and $\mathcal{B}_\ell$ is a space of reachable from b_k beliefs. Further, for clarity we define

$$A_\ell^\delta\triangleq\{b_\ell{:}b_\ell\in\mathcal{B}_\ell,\phi(b_\ell)\geq\delta\}. \tag{13}$$

Note that in the inner constraint, the policy is realized since the beliefs are given as input. Let us interpret the two forms, (11) and (12), of the inner constraint in (10). The first form (11) is formulated with respect to a cumulative value of the operator ϕ along a sequence of beliefs generated by a sequence of possible future observations episode. In this

form, we permit the immediate value of the operator ϕ to deviate but the cumulative value shall fulfill the inequality (11). In contrast, (12) states that every value of ϕ in the sequence of the beliefs shall fulfill the inequality (12), meaning to be larger than or equal to δ. Both formulations are novel, to the best of our knowledge. Furthermore, the form of (11) is motivated by the long-standing question of stopping exploration [7]. The form of (12) is motivated by safety, e.g. collision avoidance, and is the subject of our interest in this paper. When the problem (9) is augmented with the PC (10), ideally, every selection of the action following the future policy $\pi^*_{(k+1)+}$ shall take into account the outer constraint (10) at the root of the belief tree. Note that a particular case of candidate policy is a predefined static action sequence $\pi_{k:k+L-1} \equiv a_{k:k+L-1}$. In the OL setting our objective is

$$a^*_{k+} \in \arg\max_{a_{k+}\in\mathcal{A}_k} V(b_k, a_{k:k+L-1}; \rho) \quad \text{subject to} \tag{14}$$

$$\mathrm{P}\left(c(b_{k:k+L}; \phi, \delta) = 1 \mid b_k, a_{k:k+L-1}\right) \geq 1 - \epsilon, \tag{15}$$

where $\mathcal{A}_k$ is the space of candidate action sequences returned by an external process. We explain it more thoroughly in Simulations Section 8.4.

3.2 Dependence of the Inner Threshold on History

Let us clarify that in our approach the δ is constant and defined per planning session. This is equivalent to saying that $\delta(h_k)$ is a function of a history given in a planning session, namely h_k corresponding to $b_k(h_k)$, and the future thresholds are set according to the following rule

$$\delta(h_\ell a_\ell z_{\ell+1}) \triangleq \delta(h_\ell). \tag{16}$$

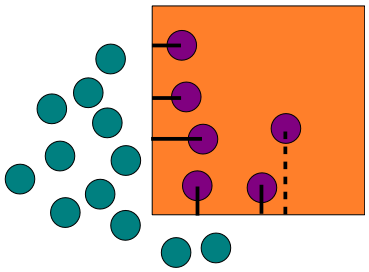


Fig. 3: Visualization of CVaR safety in the setting of a given $\mathcal{Y}^i$ (white space). The teal particles have zero distance to safe space. The purple particles have distances marked by black thick lines. For $6/18 \leq \alpha \leq 1$ the VaR will be zero and CVaR will average all the distances from purple particles marked by the black lines. If $\alpha = 1/18$, the CVaR is equal to the distance marked by the dashed line.

3.3 Possible Constraints

Subsequent to the formulation of the problem, in due course, we focus on several possible operators ϕ applicable for the inner constraint in (10)($c(b_{k:k+L}; \phi, \delta)$=1) of both forms (11) and (12).

One important example is a safety constraint, e.g., collision avoidance or energy consumption. In general form, utilizing our formulation, it would be

$$\mathrm{P}\big(\mathbf{1}_{\{b_{k:k+L}\in\mathcal{B}^{\text{safe}}_{k:k+L}\}} = 1 | b_k, \pi_{k+1:k+L-1}, a_k\big) \geq 1 - \epsilon, \tag{17}$$

where $\mathcal{B}^{\text{safe}}_{k:k+L}$ is the space of safe belief sequences (will be defined shortly) starting at time index k and of the length L. It holds that $\mathcal{B}^{\text{safe}}_{k:k+L} \subseteq \mathcal{B}_{k:k+L}$. To relate to (10), in (17): $c(b_{k:k+L}) \triangleq \mathbf{1}_{\{b_{k:k+L}\in\mathcal{B}^{\text{safe}}_{k:k+L}\}}(b_{k:k+L})$. The safeness of a sequence of beliefs $b_{k+1:k+L}$ can be defined in various ways. One possibility is

$$\phi(b_\ell(h_\ell)) \triangleq \mathrm{P}\big(\{x_\ell \in \mathcal{X}^{\text{safe}}_\ell\} | b_\ell(h_\ell)\big) = \mathbb{E}\big[\mathbf{1}_{\{x_\ell\in\mathcal{X}^{\text{safe}}_\ell\}} | b_\ell(h_\ell)\big] = \mathbb{E}\big[\mathbf{1}_{\{x_\ell\in\mathcal{X}^{\text{safe}}_\ell\}} | h_\ell\big]. \tag{18}$$

Note, in (18) belief dependent operator is actually history dependent. Contingent upon (18) the random variable distinguishing the safe and dangerous event is

$$\mathbf{1}_{\{b_{k:k+L}\in\mathcal{B}^{\text{safe}}_{k:k+L}\}} \triangleq \prod_{\ell=k}^{k+L} \mathbf{1}_{\{b_\ell : b_\ell \in \mathcal{B}_\ell, \mathrm{P}(\mathbf{1}_{\{x_\ell\in\mathcal{X}^{\text{safe}}_\ell\}} = 1 | b_\ell) \geq \delta\}}, \tag{19}$$

where $\mathcal{X}^{\text{safe}}_\ell$ is the safe space, which generally can be time-dependent dynamic environment, e.g., due to moving obstacles in the context of collision avoidance.

Another possibility is to use the Value at Risk (VaR) or Conditional Value at Risk (CVaR) operator for collision avoidance as $-\phi$ (minus sign is needed merely to maintain $\geq$ in (12)). We define the deviation of the robot's position from the safe region $\mathcal{Y}^i$ considering the obstacle i as follows $\mathrm{dist}(x, \mathcal{Y}^i) \triangleq \min_{y\in\mathcal{Y}^i} \|x - y\|_2$. Note that $\bigcap_{i=1}^M \mathcal{Y}^i \triangleq \mathcal{X}^{\text{safe}}$ for M obstacles. The following belief-dependent constraint $\mathrm{VaR}^b_\alpha[\mathrm{dist}(x, \mathcal{Y}^i)] \leq \delta$ ensures the safety, where the belief is over the agent pose x, such that $x \sim b$. Now the event to be safe is

$$\mathbf{1}_{\{b_{k:k+L}\in\mathcal{B}^{\text{safe}}_{k:k+L}\}} \triangleq \prod_{\ell=k}^{k+L} \prod_{\ell=1}^{M} \mathbf{1}_{\{-\mathrm{VaR}^{b_\ell}_\alpha[\mathrm{dist}(x_\ell, \mathcal{Y}^i_\ell)] \geq \delta\}}, \tag{20}$$

where $\mathrm{VaR}_\alpha[\mathrm{dist}(x, \mathcal{Y}^i)]$ at confidence level α is the $1-\alpha$ quantile of $\mathrm{dist}(x, \mathcal{Y}^i)$, namely,

$$\mathrm{VaR}^b_\alpha[\mathrm{dist}(x, \mathcal{Y}^i)] \triangleq \min\{\xi | \mathrm{P}(\mathrm{dist}(x, \mathcal{Y}^i) \leq \xi) \geq 1 - \alpha\}. \tag{21}$$

The value $\mathrm{VaR}_\alpha^b[\mathrm{dist}(x,\mathcal{Y}^i)]$ is the minimal value such that with probability at least $1-\alpha$ the deviation from the safe space considering one obstacle is smaller than or equal it. Another possibility is to use the condition $\mathrm{CVaR}_\alpha^b[\mathrm{dist}(x,\mathcal{Y}^i)]\leq\delta$. In this case we have that

$$\mathbf{1}_{\left\{b_{k:k+L}\in\mathcal{B}^{\mathrm{safe}}_{k:k+L}\right\}}\triangleq\prod_{\ell=k}^{k+L}\prod_{\ell=1}^{M}\mathbf{1}_{\left\{-\mathrm{CVaR}_\alpha^{b_\ell}[\mathrm{dist}(x_\ell,\mathcal{Y}_\ell^i)]\geq\delta.\right\}}. \tag{22}$$

Let us explain the meaning of such a constraint. For the obstacle i, by definition

$$\mathrm{CVaR}_\alpha^b[\mathrm{dist}(x,\mathcal{Y}^i)]\triangleq\mathbb{E}[\mathrm{dist}(x,\mathcal{Y}^i)|\{x:\mathrm{dist}(x,\mathcal{Y}^i)>\mathrm{VaR}_\alpha[\mathrm{dist}(x,\mathcal{Y}^i)]\}].$$

The CVaR is taking the average of the unsafe tail. Meaning if the unsafe tail is extremely unsafe but with low probability, such a constraint will catch that (See Fig. 3). The CVaR operator quantifies how bad the collision will be. The inner constraint of PC formulated with probabilities(19) as well as CC [30] are unable to distinguish such a behavior.

Note that the VaR and CVaR operators cannot be represented by the expectation operator with respect to the belief as in (18). Therefore, these are general belief-dependent constraints operators and *not supported* by existing constrained POMDP approaches. The distribution over the unsafe part of the beliefs is inaccessible. We note that such a constraint was suggested by [29] in the MDP setting and by [11], in the setting of randomly moving obstacles. However, [11] assumes deterministic motion and observation models, and not the general POMDP setting considered herein. In our case such a constraint has an entirely different meaning because we constrain not the actual possible robot position but what the robot believes about its position. One possibility to handle randomly moving obstacles is to redefine (20) and (22) as

$$\mathbf{1}_{\left\{b_{k:k+L}\in\mathcal{B}^{\mathrm{safe}}_{k:k+L}\right\}}\triangleq\prod_{\ell=k}^{k+L}\prod_{\ell=1}^{M}\mathbf{1}_{\left\{-\mathbb{E}[\mathrm{VaR}_\alpha^{b_\ell}[\mathrm{dist}(x_\ell,\mathcal{Y}_\ell^i)]]\geq\delta.\right\}}, \tag{23}$$

$$\mathbf{1}_{\left\{b_{k:k+L}\in\mathcal{B}^{\mathrm{safe}}_{k:k+L}\right\}}\triangleq\prod_{\ell=k}^{k+L}\prod_{\ell=1}^{M}\mathbf{1}_{\left\{-\mathbb{E}[\mathrm{CVaR}_\alpha^{b_\ell}[\mathrm{dist}(x_\ell,\mathcal{Y}_\ell^i)]]\geq\delta.\right\}}, \tag{24}$$

where the additional expectation is with respect to distribution of the safe space $\mathcal{Y}^i$. However this is out of the scope of this paper. For collision avoidance, the robot desires to navigate in the intersection of safe regions regarding all the obstacles. However, as we further show in this paper, in the POMDP setting, it is problematic to constrain possible future robot positions as done in MDP. This fact gives place to the belief-dependent operators presented above.

Another example of a general belief-dependent constraint is Information Gain (IG), defined as follows

$$\phi(b,a,z',b')=\mathrm{IG}(b,a,z',b')=-\mathcal{H}(b')+\mathcal{H}(b), \tag{25}$$

where $\mathcal{H}(\cdot)$ denotes differential entropy. Utilizing this constraint with the form of (11) allows one to reason if the cumulative IG along a planning horizon is significant enough (above threshold δ) with the probability of at least $1-\epsilon$. Such a capability has a number of implications. For instance, in the context of Informative Planning and active SLAM, instead of prompting the agent to maximize its IG, we can require that it does so only if it is able to decrease uncertainty in some tangible amount. The robot can say no. This is a new concept made possible by our general formulation, which therefore can be used to identify, e.g., when to stop exploration [38].

Let us discuss one more important constraint, the probability of reaching a goal (see, e.g., [6]). Throughout the manuscript, for clarity, we assumed that the operator ϕ is identical for all time indices. We now relax that assumption and redefine the inner constraint of the first form as follows[1]

$$c(b_{k:k+L};\phi_{k:k+L},\delta)\triangleq\mathbf{1}_{\{b_{k:k+L},b_{k:k+L}\in\mathcal{B}_{k:k+L},(\sum_{\ell=k}^{k+L-1}\phi_{\ell+1}(b_\ell,b_{\ell+1}))\geq\delta\}}(b_{k:k+L}). \tag{26}$$

Further, let $\phi_{\ell+1}(\cdot)\equiv 0\quad\forall\ell\in k:k+L-2$ and

$$\phi_{k+L}(b_{k+L})=\mathrm{P}(\{x_{k+L}\in\mathcal{X}^{\mathrm{goal}}\}|b_{k+L}), \tag{27}$$

where (27) defines the task of reaching the goal.

4 Approach for Our Probabilistic Belief-Dependent Constraints

Having presented our problem formulation and the examples of possible belief-dependent operators to serve as an inner constraint, we are keen to proceed into the adaptive approach to precisely evaluate the sample approximation of our PC. We used the term "precisely" to emphasize that, in contrast to the only necessary pruning condition suggested in [30] for CC, our adaptive evaluation is necessary and sufficient.

[1]We denote $f\equiv g$ for two operators, if we have $f(x)=g(x)\quad\forall x$.

4.1 Coupled Outer Constraint Evaluation and Belief Tree Construction

In this section, we delve into the evaluation of our novel formulation of the PC (10). We start by presenting a lemma.

Lemma 1 (Representation of our outer constraint)**.**

$$\mathrm{P}\left(c(b_{k:k+L};\phi,\delta)=1|b_k,\pi_{k+1:k+L-1},a_k\right)=\mathbb{E}_{z_{k+1:k+L}}[c(b_{k:k+L};\phi,\delta)|b_k,\pi_{k+1:k+L-1},a_k]. \tag{28}$$

In addition, if the inner constraint conforms to (12)*,namely* $c(b_{k:k+L};\phi,\delta)=\prod_{\ell=k}^{k+L}\mathbf{1}_{A_\ell^\delta}(b_\ell)$*, we have that*

$$\mathbb{E}_{z_{k+1:k+L}}\left[\prod_{\ell=k}^{k+L}\mathbf{1}_{A_\ell^\delta}(b_\ell)|b_k,\pi,a_k\right]=\mathbf{1}_{A_k^\delta}(b_k)\mathbb{E}_{z_{k+1}}[\mathbf{1}_{A_{k+1}^\delta}(b_{k+1})\mathbb{E}_{z_{k+2}}[\mathbf{1}_{A_{k+2}^\delta}(b_{k+2})\dots \tag{29}$$

$$\dots\mathbb{E}_{z_{k+L-1}}[\mathbf{1}_{A_{k+L-1}^\delta}(b_{k+L-1})\mathbb{E}_{z_{k+L}}[\mathbf{1}_{A_{k+L}^\delta}|b_{k+L-1},\pi]|b_{k+L-2},\pi]\dots|b_{k+1},\pi]|b_k,\pi]=$$

$$\mathbf{1}_{A_k^\delta}(b_k)\mathbb{E}_{z_{k+1}}[\mathbf{1}_{A_{k+1}^\delta}(b_{k+1})\mathbb{E}_{z_{k+2}}[\mathrm{P}(c(b_{k+2:k+L};\phi,\delta)=1|b_{k+2},\pi)]=\mathbf{1}_{A_k^\delta}(b_k)\mathbb{E}_{z_{k+1}}[\mathrm{P}(c(b_{k+1:k+L};\phi,\delta)=1|b_{k+1},\pi)]. \tag{30}$$

where the set $A_\ell^\delta\ \forall\ell\in[k:k+L]$ *is defined by* (13)*.*

The reader can find the proof in Appendix A.1. From Lemma 1 we behold how to obtain the best sample approximation of the outer constraint, since the theoretical expectation (28) is out of the reach. In practice, we approximate expectation in (28) with a finite number m of samples of observation episodes, $\{z_{k+1:k+L}^l\}_{l=1}^m$, which we call laces, such that

$$\hat{\mathrm{P}}^{(m)}(c=1|b_k,\pi)=\hat{\mathrm{P}}^{(m)}(c(b_{k:k+L};\phi,\delta)=1|b_k,\pi)\triangleq\tfrac{1}{m}\textstyle\sum_{l=1}^m c(b_{k:k+L}^l;\phi,\delta)=\tfrac{1}{m}\sum_{l=1}^m c^l, \tag{31}$$

where $c^l\triangleq c(b_{k:k+L}^l;\phi,\delta)$ and $c^l\sim\mathrm{P}(c|b_k,\pi)$. The outer constraint (10) becomes

$$\tfrac{1}{m}\textstyle\sum_{l=1}^m c(b_{k:k+L}^l;\phi,\delta)\geq 1-\epsilon, \tag{32}$$

where m is the number of the observation sequences $z_{k+1:k+L}$ expanded from action a_k (9) at the root of the belief tree. From now on, we focus on the sample approximation (32) of the outer (external) constraint (10).

Importantly, our further discussed approach is valid for any sampler utilized to obtain samples of (32). Of course with growing horizon $L\to\infty$ to adequately represent actual distributions more samples will be needed, the m will need to be enlarged. This aspect is an inherent property of the sampler and hence is not our concern for now. If the belief tree is given, we can traverse it from the bottom up and calculate the value of c^l for $l\in 1\dots m$ along the way such that when we reach the root, we have everything to evaluate (32). In general, since the parameter m has to be known, this applies to approaches that decouple belief tree construction from the solution, e.g., SS algorithm [18] and OL setting.

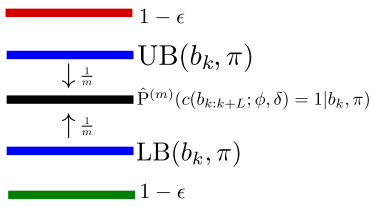


Fig. 4: Illustration of the bounds. If $1-\epsilon$ is below as marked by the green line, we are only able to early accept the policy π using the lower bound $\mathrm{LB}(b_k,\pi)$ (33). It is only possible to discard the policy π with the upper bound $\mathrm{UB}(b_k,\pi)$ (34), in case that $1-\epsilon$ is located above, as marked by the red line.

However, we would like to guide the belief tree construction such that if, e.g., the action does not fulfill the outer constraint we will spend on it as less effort as possible. We shall regard another interesting aspect of (28). Because $c\in\{0,1\}$, by definition $1\geq\frac{1}{m}\sum_{l=1}^m c^l$. This implies that, under the condition $\epsilon=0$, to satisfy (32), we shall require $\sum_{l=1}^m c^l=m$. In other words, in this setting we will not be able to early accept an action (before expanding m future belief laces). Nevertheless, as we will further see, we will be able to do a highly efficient pruning whenever the inner constraint conforms to (12) (multiplicative form). Further, we describe an adaptive constraint evaluation mechanism for a general ϵ and after that focus on the case of $\epsilon=0$.

4.1.1 Accurate Adaptive Constraint Inquiry with $0\leq\epsilon<1$

Having presented the sample approximation of PC based on m samples (32), we are now ready to address a complete belief tree construction. We bound the expression of the sample approximation of the outer constraint (32) from each end using the already expanded part of the belief tree. Suppose the online algorithm at the root for each action expands upon termination m laces appropriate to the drawn samples of observation episodes of length $L-1$, namely $\{z_{k+1:k+L}^l\}_{l=1}^m$. Each sampled lace l corresponds to a particular realization of the return.

Suppose the algorithm already expanded $n\leq m$ laces. The lower bound $\mathrm{LB}(b_k,\pi)$ of (32) is

$$1-\epsilon\overbrace{\leq}^{?}\underbrace{\tfrac{1}{m}\textstyle\sum_{l=1}^n c^l}_{\mathrm{LB}(b_k,\pi)}\leq\underbrace{\tfrac{1}{m}\textstyle\sum_{l=1}^m c^l}_{\hat{\mathrm{P}}^{(m)}(c=1|b_k,\pi)}. \tag{33}$$

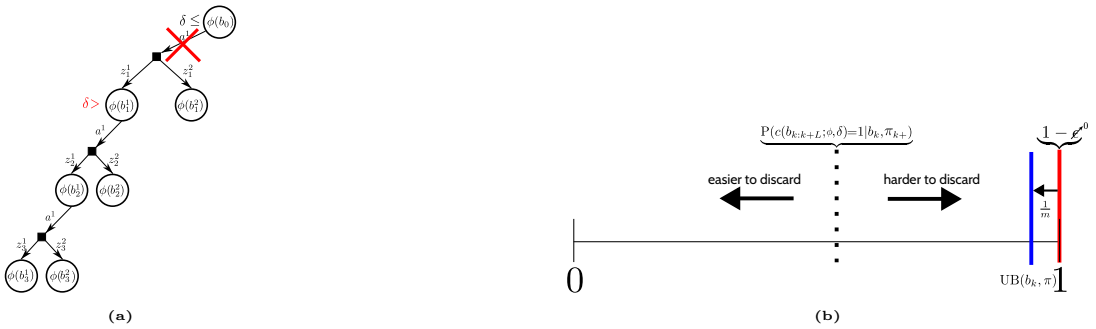


Fig. 5: (a) Fast, adaptive with respect to observation laces, pruning when ϵ=0. If a single descendant belief yields $\phi(b)<\delta$ the whole action branch can be safely discarded (necessary condition (36)). From the other hand if all beliefs in the sub-policy-tree satisfy $\phi(b)\geq\delta$ the PC is fulfilled (sufficient condition (37)). To rephrase that, if the action was not pruned, it is guaranteed to satisfy the PC; **(b)** Visualization of a faster pruning when $\mathrm{P}\left(c(b_{k:k+L};\phi,\delta)=1|b_k,\pi\right)$ is lower.

Whereas the upper bound $\mathrm{UB}(b_k,\pi)$ reads

$$\underbrace{\tfrac{1}{m}\textstyle\sum_{l=1}^{m} c^l}_{\hat{\mathrm{P}}^{(m)}(c=1|b_k,\pi)} \leq \underbrace{\tfrac{m-n}{m}+\tfrac{1}{m}\textstyle\sum_{l=1}^{n} c^l}_{\mathrm{UB}(b_k,\pi)} \overset{?}{<} 1-\epsilon. \tag{34}$$

By the question mark we denote the inequalities that shall be fulfilled online to check either the sample approximation of the outer constraint (32) is met (33) or violated (34). Only one of the inequalities denoted by the question mark will be eventually fulfilled with some n when we progressively expand the laces. We accept a policy fulfilling (32) using the lower bound (33) or invalidate using the upper bound (34) (See Fig. 4). These bounds allow to evaluate (32) **adaptively** before expanding all the m laces of belief sequences $b_{k:k+L}$ and using only n laces instead. We save time that would be spent on the additional $m-n$ laces if one continues to sample observation episodes (laces) up until the budget of m samples is reached.

Such a technique is a applicable for both settings: OL and CL. Note that both bounds advance towards the (31) with the step size $1/m$. Moreover, with each added observations episode lace, only one of the bounds is contracting, the lower (33) bound $\mathrm{LB}(b_k,\pi)$ or the upper (34) bound $\mathrm{UB}(b_k,\pi)$. If the expanded lace results in c^l=1 the lower bound (33) makes a step towards (31). This event happens with probability $\mathrm{P}(c{=}1|b_k,\pi)$. Conversely, if the expanded lace results in c^l=0, the upper bound (34) makes a step towards (31). This event happens with probability $\mathrm{P}(c{=}0|b_k,\pi)$.

One example of an adaptive usage of (33) and (34) is to save time in an OL planning or alternatively spend more time on the action sequences which fulfill the outer constraint (32), namely increase m for a given n up until evaluating (32) is still possible with this n. Envisage a static action sequence to be checked. After each expanded lace c^l of (32) we are probing (33). If fulfilled, we know that the sample approximation of the outer constraint is satisfied, and we can stop dealing with the constraints for this candidate action sequence. Else we are trying (34); if fulfilled, we know that the current action sequence violates the sample approximation of the outer constraint (32). The third possibility is to add one more lace and check again. In such a way, we adaptively expand the lowest possible number of inner constraint laces to be evaluated and **validate** or **invalidate** the action sequence depending on whether the PC (32) is fulfilled on not. The presented adaptivity mechanism is simple, exact and guaranteed to satisfy or discard our PC. To our knowledge no analogs to this exists in the literature, e.g. [30]. Another example is the CL setting, where we deal with policies. Further in this manuscript we focus attentively on the multiplicative form of the inner constraint (12).

4.1.2 Early Termination with Multiplicative Form

In this section we highlight that in case of multiplicative form for each lace $c^l \in \{0,1\}$, it holds that

$$c(b_{k:k+L};\phi,\delta) = \left(\textstyle\prod_{\ell=k}^{k+L} \mathbf{1}_{\{b_\ell:b_\ell\in\mathcal{B}_\ell,\phi(b_\ell)\geq\delta\}}(b_\ell)\right) \leq \left(\textstyle\prod_{\ell=k}^{k+j} \mathbf{1}_{\{b_\ell:b_\ell\in\mathcal{B}_\ell,\phi(b_\ell)\geq\delta\}}(b_\ell)\right) = \bar{c}(b_{k:k+j};\phi,\delta). \tag{35}$$

for $0\leq j\leq L$. Remembering that $c,\bar{c}\in\{0,1\}$, if $\bar{c}^l=0$ so as $c^l=0$.

4.1.3 Efficient Exact Adaptive Pruning with $\epsilon=0$ and Multiplicative Form

The constraint confidence parameter ϵ controls the stiffness of the condition that the distribution of belief-dependent inner constraint shall fulfill. The maximal stiffness is reached when ϵ=0. Leveraging again the multiplicative structure of the inner constraint (12), we have an interesting behavior summarized in the following theorem (See Fig. 5a).

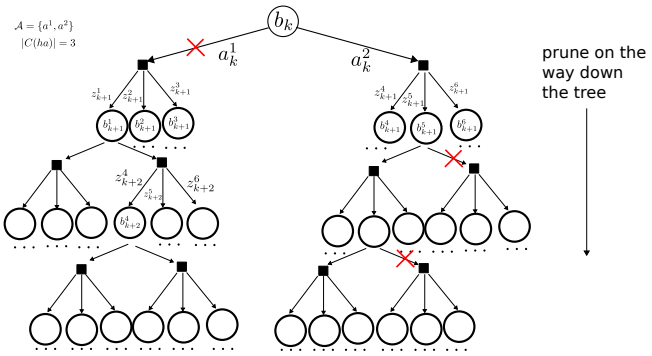


Fig. 6: Illustration of the symmetric belief search tree built by Sparse Sampling algorithm with $\mathcal{A} = \{a^1, a^2\}$ and $m = 3$ sampled observations at each belief node. We prune branches on the way down the tree.

Theorem 1 (Necessary and sufficient condition for feasibility of a Probabilistic Constraint)**.** *Fix* ϵ=0 *and* $\delta\in\mathbb{R}$. *Let the inner constraint comply to* (12)*, namely* $\forall l\ c(b^l_{k:k+L};\phi,\delta)=\prod_{\ell=k}^{k+L}\mathbf{1}_{\{b_\ell:b_\ell\in\mathcal{B}_\ell,\phi(b_\ell)\geq\delta\}}(b^l_\ell)$. *The fact that*

$$\left(\frac{1}{m}\sum_{l=1}^{m}\prod_{\ell=k}^{k+L}\mathbf{1}_{\{b_\ell:b_\ell\in\mathcal{B}_\ell,\phi(b_\ell)\geq\delta\}}(b^l_\ell)\right)=1 \quad \texttt{necessary condition} \tag{36}$$

implies that $\forall l,\ell\quad \phi(b^l_\ell)\geq\delta$. *Moreover, if*

$$\left(\frac{1}{m}\sum_{l=1}^{m}\prod_{\ell=k}^{k+L}\mathbf{1}_{\{b_\ell:b_\ell\in\mathcal{B}_\ell,\phi(b_\ell)\geq\delta\}}(b^l_\ell)\right)<1, \quad \texttt{sufficient condition} \tag{37}$$

so $\exists l,\ell\quad \phi(b^l_\ell)<\delta$.

We provide a proof in Appendix A.2. Theorem 1 says that whenever ϵ=0 and the inner constraint is of the multiplicative flavor (12), then (32) is satisfied if and only if for every belief b in the belief tree it holds $\phi(b)\geq\delta$.

An immediate result of Theorem 1 is the soundness of our pruning technique. On our way down the tree, by arriving to a belief b_ℓ, we prune all the actions in the belief tree resulting in $\phi(b_{\ell+1})<\delta$ for some future observation a single step ahead. In such a way eventually in the belief tree will be solely the actions satisfying the PC (32) with ϵ=0. Importantly, to engage such a pruning we do not need to know actual value of m.

Behold one more interesting aspect. As explained in Section 4.1.1 the upper bound (34) makes a step when the sampled lace l equals to zero (c^l=0). Recall that we discard the policy π and action sequence a_{k+} using the upper bound (34). Whenever ϵ=0, it is sufficient to make a single step to discard such a policy. This step happens with probability $\mathrm{P}\,(c{=}0|b_k,\pi){=}1{-}\mathrm{P}\,(c{=}1|b_k,\pi)$. Therefore if the Probabilistic **theoretical** Constraint (10) with ϵ=0 is violated with a large margin, our method will prune such a policy faster, as visualized in Fig. 5b.

4.2 The Algorithms for PC

In this section, we present algorithms to tackle our novel formulation portrayed by (9), (12) and (32) applying the theory presented in the previous section. Let us reiterate that we focus now on the multiplicative form (12) of the inner constraint. In particular, inspired by SS [18] and adaptivity aspects in [4], we first present an adaptive algorithm originated from SS, for ϵ=0. The SS based methods on top of nonparametric BMDP hardly can be applied with long horizons. To alleviate that and use an arbitrary ϵ in the interval $[0,1)$, we assume static candidate action sequences. We formulated all algorithms for a general belief-dependent operator ϕ. In all our algorithms the objective is calculated over the symmetric belief tree (Fig. 6) and the tree future policy is deterministic.

4.2.1 Probabilistically-constrained Sparse Sampling ($\epsilon = 0$)

In this section we present an algorithm to solve the sample approximation of the following problem

$$\begin{aligned} a^*_k &\in \arg\max_{a_k\in\mathcal{A}} Q^{\pi^*}(b_k,a_k;\rho) \quad \text{subject to} \\ \mathrm{P}\Big(\Big(\prod_{\ell=k}^{k+L}\mathbf{1}_{A^\delta_\ell}(b_\ell)\Big)=1|b_k,a_k,\pi^*_{k+1:k+L-1}\Big)&=1, \end{aligned} \tag{38}$$

where A^δ_ℓ is defined by (13). In this approach, since ϵ=0, using the recursive nature of multiplicative inner constraint proved in Lemma 1 we ensure that our future belief tree policy $\pi^*_{(k+1)}$ is safe with respect to each belief $b(h)$ in the belief tree, namely $\hat{\mathrm{P}}^{(m)}(c{=}1|b(h),a,\pi){=}1$ (Note that PC itself is a belief-dependent operator under a particular execution policy). Indeed, as we will further see, if some action in the way down the tree is not pruned, it has to hold that all the predecessors are fulfilling the PC and the current node. SS based methods employ the Bellman optimality criterion while

traversing the tree from the bottom up. For ϵ=0 we suggest Alg. 1. Leveraging Theorem 1, the Alg. 1 prunes, in line 14, all the actions resulting in even a single future belief to be unsafe. We prune actions violating the outer constraint (32) on the way forward (down the tree) as visualized at Fig 6. Because we actually check the inner constraint (12) on the way forward when the algorithm hits the bottom of the tree, we are left solely with actions fulfilling the PC approximated by (32) with ϵ=0, so we do not need any additional checking on the way up at all. This contrasts the chance-constrained formulation in [30], where the pruning condition is only necessary and not sufficient as in our approach.

Algorithm 1 Prob. Constrained BMDP Sparse Sampling ($\epsilon = 0$) (PCSS)

1: **procedure** PCSS(belief: $b(h)$, history: h, depth: d, threshold: $\delta(h)$) ▷ b as in (5), δ as discussed in Section 3.2.
2: **if** $d = 0$ **then**
3: **return** (Null, 0)
4: **end if**
5: $(a^*, v^*) \leftarrow$ (Null, $-\infty$)
6: **for** $a \in \mathcal{A}$ **do**
7: $v \leftarrow 0.0$, PrunedFlag $\leftarrow$ false ▷ Initialization of Value function and flag for pruning
8: Calculate propagated belief b'^{-} applying action a
9: **for** m_d times **do**
10: Sample $x^o \sim b'^{-}$ followed by $z' \sim \mathbb{P}(z|x^o)$ Observations are created using belief defined by (5) and action a.
11: $b' \leftarrow \psi(b, a, z')$ ▷ Update belief
12: **if** $\phi(b') < \delta(h)$ **then** ▷ Prune action a using Theorem 1
13: PrunedFlag $\leftarrow$ true
14: **break** ▷ Exit from observations loop and go to line 21
15: **end if**
16: $a^{*,\prime}$, $v' \leftarrow$ PCSS(b', haz',$d-1$, $\delta(h)$) ▷ Rule for threshold as in (16). The best next action $a^{*,\prime}$ is redundant.
17: $v+ = (\rho(b, a, b') + \gamma \cdot v')/m_d$ ▷ Calculate Value fun. using regular beliefs as in (5).
18: **end for**
19: **if** PrunedFlag is false and $v > v^*$ **then**
20: $(a^*, v^*) \leftarrow (a, v)$
21: **end if**
22: **end for**
23: **return** (a^*, v^*)
24: **end procedure**

We now endow each belief and corresponding history by superscript denoting a global index at a particular depth of belief tree, see Fig. 6. Similar to SS algorithm, we use Bellman optimality from the leaves up the tree and approximate the (3) by

$$\hat{Q}^{\pi^*_{(\ell+1)+}}(h^{i_\ell}_\ell, a_\ell; \rho) = {}^1\!/_{|C(h^{i_\ell}_\ell a_\ell)|} \sum_{i_{\ell+1} \in C(h^{i_\ell}_\ell a_\ell)} \left(\rho(b^{i_\ell}_\ell, a_\ell, z^{i_{\ell+1}}_{\ell+1}, b^{i_{\ell+1}}_{\ell+1}) + \hat{Q}^{\pi^*_{(\ell+2)+}}(h^{i_{\ell+1}}_{\ell+1}, a^*_{\ell+1}; \rho)\right), \tag{39}$$

where $|C(h^{i_\ell} a_\ell| \equiv m$ and on the way up the tree we select an optimal action using $a^*_{\ell+1} = \underset{a_{\ell+1} \in C(h_{\ell+1})}{\arg\max}\ \hat{Q}^{\pi^*}(b_{\ell+1}, a_{\ell+1}; \rho)$. Since we prune dangerous actions on the way down the tree, it holds that $C(h_{\ell+1}) \subseteq \mathcal{A}$ and $|C(h_{\ell+1})| \leq |\mathcal{A}|$. Therefore,the proof of near optimality from [18], [24], [22] [23] holds with respect to reward. However, we prune at each belief action node using the sample approximation of PC and not theoretical. Therefore our method is only an approximate method. Clearly, it converges to the solution of (9) and (10) as number of sampled observations grows. At the root we have $m = \prod_{i=1}^{L} (m_d)^i$ laces and the approximation of PC for action a_k, when the Alg. 1 returned from the recursion, reads

$$\begin{aligned} &\Big(\frac{\mathbf{1}_{A^\delta_k}(b^1_k)}{|C(h^1_k a_k)|} \sum_{i_{k+1} \in C(h^1_k a^*_k)} \frac{\mathbf{1}_{A^\delta_{k+1}}(b^{i_{k+1}}_{k+1})}{|C(h^{i_{k+1}}_{k+1} a^*_{k+1})|} \sum_{i_{k+2} \in C(h^{i_{k+1}}_{k+1} a^*_{k+1})} \frac{\mathbf{1}_{A^\delta_{k+2}}(b^{i_{k+2}}_{k+2})}{|C(h^{i_{k+2}}_{k+2} a^*_{k+2})|} \cdots \\ &\cdots \frac{\mathbf{1}_{A^\delta_{k+L-1}}(b^{i_{k+L-1}}_{k+L-1})}{|C(h^{i_{k+L-1}}_{k+L-1} a^*_{k+L-1})|} \sum_{i_{k+L} \in C(h^{i_{k+L-1}}_{k+L-1} a^*_{k+L-1})} \mathbf{1}_{A^\delta_{k+L}}(b^{i_{k+L}}_{k+L}) \Big) = 1. \end{aligned} \tag{40}$$

It shall be noted that the presented algorithm is heavy from the computational point of view due to the fact that it expands all actions and predefined number of observations on the way down the tree. It is hard to apply Alg. 1 to large horizons even with our efficient pruning technique. We alleviate this issue in the next section.

Algorithm 2 Arbitrary $0 \leq \epsilon < 1$

1: **Input:** $\mathcal{A}$, $b_k(h_k), \delta(h_k)$ ▷ Set of the action sequences, belief and constant threshold as in explained in Section 3.2
2: $a^*_{k+} \leftarrow$ undef, $\hat{V}^*_{(m)} \leftarrow -\infty$, $S \leftarrow \{\}$ ▷ S is the set of accepted candidate action sequences, $S \subseteq \mathcal{A}$
3: **for** each $a_{k+} \in \mathcal{A}$ **do**
4: **for** $n(a_{k+}) \in 1:m$ **do**
5: Draw observation sequence $z^{n(a_{k+})}_{k+1:k+L} \sim \mathbb{P}(z_{k+1}|b_k, a_k) \prod_{\ell=k+1}^{k+L-1} \mathbb{P}(z_{\ell+1}|b_\ell, a_\ell)$ ▷ Observations are sampled from $\mathbb{P}(z_{\ell+1}|b_k, a_{k:\ell}, z_{k+1:\ell})$ $\ell = k : k+L-1$ using beliefs as in (5).
6: $c^{n(a_{k+})} \leftarrow \mathbf{1}_{A^{\delta_k}}(b_k)$
7: **for** $z^{n(a_{k+})}_\ell \in z^{n(a_{k+})}_{k+1:k+L}$ **do**
8: Calculate $b^{n(a_{k+})}_\ell$, $\phi(b^{n(a_{k+})}_\ell)$, $\rho(b^n_{\ell-1}, a_{\ell-1}, b^n_\ell)$
9: $c^{n(a_{k+})} \leftarrow c^{n(a_{k+})} \cdot \mathbf{1}_{A^\delta_\ell}(b^n_\ell)$ ▷ For definition of A^δ_ℓ see (13).
10: **if** $c^{n(a_{k+})} == 0$ **then**
11: **break** ▷ If inner constraint as in (12) can stop to calculate $\phi(\cdot)$ down the lace once at some belief $\phi(b) < \delta$. See Eq. (35).
12: **end if**
13: **end for**
14: **if** $1 - \epsilon \leq \frac{1}{m} \sum_{l=1}^{n(a_{k+})} c^l$ **then** ▷ Outer constraint is fulfilled
15: $S \leftarrow S \cup \{a_{k+}\}$ ▷ Accept the a_{k+}
16: **break** ▷ check the next action seq.
17: **else if** $\frac{1}{m} \sum_{l=1}^{n(a_{k+})} c^l < 1 - \epsilon - \frac{m - n(a_{k+})}{m}$ **then** ▷ Outer constraint is violated
18: **break** ▷ check the next action seq.
19: **end if**
20: **end for**
21: **end for**
22: **for** each $a_{k+} \in S$ **do** ▷ S contains all feasible a_{k+}
23: Expand missing laces and get $\hat{V}^{(m)}(b_k, a_{k+})$
24: **if** $\hat{V}^*_{(m)} < \hat{V}^{(m)}(b_k, a_{k+})$ **then**
25: $a^*_{k+} \leftarrow a$, $\hat{V}^*_{(m)} \leftarrow \hat{V}^{(m)}(b_k, a_{k+})$
26: **end if**
27: **end for**
28: **Return** a^*_{k+}

4.2.2 Static Action Sequences and Arbitrary ϵ in the Interval $[0, 1)$

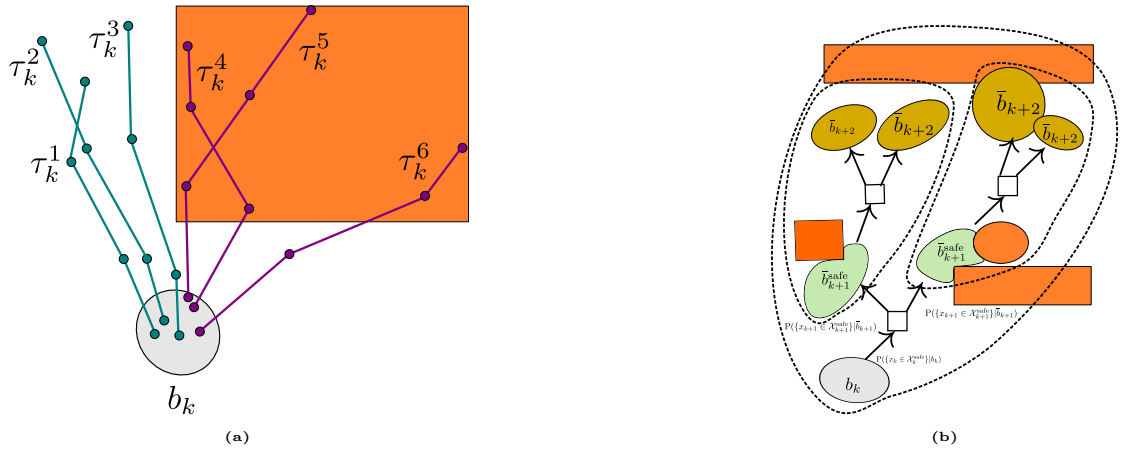
We now turn our attention to an arbitrary ϵ in the interval $[0, 1)$ and solve the sample approximation of (14) and (15) with multiplicative flavor of the inner constraint (12). Our objective is specified as

$$\begin{aligned} a^*_{k+} \in \arg\max_{a_{k+} \in \mathcal{A}_k} \tfrac{1}{m} \textstyle\sum_{l=1}^{m} \sum_{\ell=k}^{k+L-1} \rho(b^l_\ell, a_\ell, b^l_{\ell+1}) \quad \text{subject to} \\ \left(\tfrac{1}{m} \textstyle\sum_{l=1}^{m} \prod_{\ell=k}^{k+L} \mathbf{1}_{A^\delta_\ell}(b^l_\ell)\right) \geq 1 - \epsilon, \end{aligned} \tag{41}$$

where observations episode lace l is sampled from $z^l_{k+1:k+L} \sim \mathbb{P}(z_{k+1}|b_k, a_k) \prod_{\ell=k+1}^{k+L-1} \mathbb{P}(z_{\ell+1}|b_k, a_{k:\ell}, z_{k+1:\ell})$. The set A^δ_ℓ is in accord to (13). We denote the objective portrayed by (14) approximated by empirical mean of m laces of the cumulative rewards as $\hat{V}^{(m)}(b_k, a_{k+})$ and approximate (15) similar to (31). To relax the necessity that ϵ=0, we turn to static candidate action sequences ($\mathcal{A}_k$) and present Alg. 2. Let us pinpoint that we already proposed this algorithm in our parallel paper [38] where we focus on the cumulative form of the inner constraint (11). However, this paper focuses on multiplicative structure (12) and another operator ϕ being (18). In addition, since here we focus on the multiplicative flavor of the inner constraint (12), Alg. 2 ceases to calculate the operator ϕ over the lace if it encountered a belief such that $\phi(b)<\delta$ (line 10 in Alg. 2). This is especially important with long horizons, see Section 4.1.2.

5 Analysis of CC and Upgrades for PC

Although our formulation is universal and belief-dependent, this paper focuses on the agent's safety. Therefore, we shall thoroughly regard the SOTA safety constraint under partial observability, the CC.



(a) (b)

Fig. 7: Visualization of the conventional CC enforced from a belief b_k. By shaded ovals we illustrate the posterior beliefs. The orange rectangles and ovals stand for obstacles. **(a)** In this visualization the purple trajectories are unsafe, whereas the teal trajectories are safe. **(b)** Visualization of sub-policy-tree resolution of CC with horizon L=2 (the blobs encapsulating the beliefs by dashed lines). Only belief b_k here taken from the belief tree is used for reward calculation as in Fig. 1 and the rest of the beliefs are defined differently (Section 5.3).

5.1 Averaging the Probability of Being Safe

We start by showing that in case of candidate policies and not predefined action sequences the CC utilized in [6] averages the probability of to be safe given posterior belief and policy. Observe that due to dependence on the polices, to calculate (7) one **must** marginalize with respect to observations, namely

$$\mathrm{P}(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}|b_k,\pi)=\int_{z_{k+1:\ell}}\mathrm{P}(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}|b_k,\pi_{k:\ell-1},z_{k+1:\ell})\mathbb{P}(z_{k+1:\ell}|b_k,\pi)\mathrm{d}z_{k+1:\ell}=\mathbb{E}_{z_{k+1:\ell}}\left[\mathrm{P}(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}|b_\ell)\middle|b_k,\pi\right]. \quad (42)$$

The robot, with some likelihood, will obtain a **single** posterior belief in the actual inference and not in a planning session. Thus, the condition $\mathbb{E}_{z_{k+1:\ell}}\left[\mathrm{P}(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}|b_\ell)\middle|b_k,\pi\right]\geq\delta$ can be problematic. Let us give a specific example. Suppose the belief b_k is safe (probability to intersect with the dangerous region is zero). Assume that δ=0.7 and we have three equiprobable observations in a myopic setting such that $\mathrm{P}(\{x_{k+1}\in\mathcal{X}_{k+1}^{\text{safe}}\}|\psi(b_k,a_k,z_{k+1}^j))$ equals 0.1, 1.0, 1.0 for j=1, 2, 3 respectively. On average, we have precisely 0.7 such that the (7) is fulfilled. However, one belief is highly unsafe. In contrast, as our formulation (10) is sensitive to the distribution of the future beliefs rendered by future observation sequences, it is aware that only two out of the three observation sequences satisfy the constraint. For example, it will declare (the sampling-based approximation of) (10) is not satisfied if (e.g.) we select ϵ=0 and δ=0.7.

Nevertheless the expectation with respect to future beliefs is also a viable possibility for the constraint. As we explained in Section 2.3 the CC in [30] portrayed by (8) is more general than the one used in [6] and described by (7). From now on whenever we use the notion of CC we mean the CC from [30]. Further, we show two ways to calculate the CC. The first way is through the PDF of the future robot trajectories. The second way utilizes posterior beliefs defined in a different than the usual way (Section 5.3).

5.2 Chance Constraining Future Trajectories

We face that in [30], the CC is imposed, with different threshold, at each non-terminal belief in the belief tree. The non-terminal belief is the belief from which emanates an action, in other words, not a leaf in the belief tree. As we mentioned in Section 1.4, this is necessary to ensure the feasibility of CC in the root of the belief tree. It will be more apparent shortly. At this point let us define the *rewards tree* as belief tree used for the calculation of the rewards and the objective (9) (See Fig. 1 and 5a). The beliefs in the *rewards tree* are defined by (5). As we mentioned, the [30] disregards the disparity of the beliefs in the *rewards tree* and CC tree (to be defined in the next section) but takes into account the disparity in observation likelihood. To shed light on this aspect, we shall analyze the CC imposed at the root of the belief tree b_k common in both belief trees. With this motivation in mind, we focus for the moment on the time index k of the beginning of the planning session and inspect the PDF that the trajectory τ_k will be safe. Note that from the properties of the indicator variable

$$\mathbf{1}_{\left\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\right\}}(\tau_k(\omega))=\mathbf{1}_{\left\{\cap_{\ell=k}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\right\}\right\}}(\tau_k(\omega))=\bigwedge_{\ell=k}^{k+L}\mathbf{1}_{\left\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\right\}}(x_\ell(\omega))=\prod_{\ell=k}^{k+L}\mathbf{1}_{\left\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\right\}}(x_\ell(\omega)) \quad \forall\omega\in\Omega, \quad (43)$$

where $\bigwedge$ is the minimum operator and Ω is the space of the outcomes. Meaning, the safe trajectory is the trajectory comprised of safe states. Only for better and clearer explanation, we assume further in this section that robot safe workspace $\mathcal{X}_\ell^{\text{safe}}$ is given for any ℓ. Relaxing this assumption is straightforward. Another property of the indicator

variable is $\mathrm{P}(\mathbf{1}_{\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}}=1|b_k,\pi,\tau_k)=\mathbf{1}_{\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}}(\tau_k)$, enabling us to write (See Fig. 7a)

$$\mathrm{P}(\mathbf{1}_{\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}}=1|b_k,\pi)=\mathbb{E}_{\tau_k}[\mathbf{1}_{\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}}|b_k,\pi]=\int_{\tau_k}\prod_{\ell=k}^{k+L}\mathbf{1}_{\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}}(x_\ell)\mathbb{P}(\tau_k|b_k,\pi)\mathrm{d}\tau_k. \tag{44}$$

To calculate the value of (44), one must asses $\mathbb{P}(\tau_k|b_k,\pi)$ and since this PDF depends on the policy, it is not possible to directly constraint the robot future trajectories. The $\mathbb{P}(\tau_k|b_k,\pi)$ is by definition averaging with respect to future observations. To show that explicitly, we present the following Lemma.

Lemma 2 (PDF of trajectory)**.** *The PDF of a future trajectory given a belief and the candidate policy* $\mathbb{P}(\tau_k|b_k,\pi_{k:k+L-1})$ *decomposes as*

$$\mathbb{P}_{\mathrm{T}}(x_{k+1}|x_k,a_k)b_k(x_k)\int_{z_{k+1:k+L-1}}\prod_{\ell=k+1}^{k+L-1}\big(\mathbb{P}_{\mathrm{T}}(x_{\ell+1}|x_\ell,\pi(b_\ell(b_{\ell-1},\pi_{\ell-1}(b_{\ell-1}),z_\ell)))\mathbb{P}_{\mathrm{O}}(z_\ell|x_\ell)\big)\mathrm{d}z_{k+1:k+L-1}= \tag{45}$$

$$\mathbb{P}_{\mathrm{T}}(x_{k+1}|x_k,a_k)b_k(x_k)\int_{z_{k+1}}\ldots\Big(\int_{z_{k+L-2}}\mathbb{P}_{\mathrm{T}}(x_{k+L-1}|x_{k+L-2},\pi(b_{k+L-2}(b_{k+L-3},a_{k+L-3},z_{k+L-2})))\cdot$$

$$\Big(\int_{z_{k+L-1}}\mathbb{P}_{\mathrm{T}}(x_{k+L}|x_{k+L-1},\pi(b_{k+L-1}(b_{k+L-2},a_{k+L-2},z_{k+L-1})))\mathbb{P}_{\mathrm{O}}(z_{k+L-1}|x_{k+L-1})\big)\mathrm{d}z_{k+L-1}\Big)\cdot \tag{46}$$

$$\mathbb{P}_{\mathrm{O}}(z_{k+L-2}|x_{k+L-2})\mathrm{d}z_{k+L-2}\big)\ldots\mathrm{d}z_{k+1}.$$

If instead of policy it is given a predefined action sequence $a_{k:k+L-1}$*, we have that*

$$\mathbb{P}(\tau_k|b_k,a_{k:k+L-1})=\mathbb{P}_{\mathrm{T}}(x_{k+1}|x_k,a_k)b_k(x_k)\textstyle\prod_{\ell=k+1}^{k+L-1}\mathbb{P}_{\mathrm{T}}(x_{\ell+1}|x_\ell,a_\ell). \tag{47}$$

We provide the proof in Appendix A.3. As we see, such a formulation averages, in each time step, the motion models corresponding to different actions due to various possible observations. We emphasized this behavior in the final time instance with the magenta color. In particular, when we deal with static action sequences $a_{k:k+L-1}$, the observations cancel out, effectively assuming full observability. Note that this is also the case in the formulation described by (42).

From Lemma 2 we conclude the following. In the case of policies, it is not possible to assume the fully observable setting. To evaluate $\mathbb{E}_{\tau_k}[\mathbf{1}_{\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}}|b_k,\pi]$ one must resort to the averaging with respect to observations and leverage the fact that $\mathbb{E}_{\tau_k}[\mathbf{1}_{\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}}|b_k,\pi]=\mathbb{E}_{z_{(k+1)+}}[\mathbb{E}_{\tau_k}[\mathbf{1}_{\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}}|b_k,\pi,z_{k+1:k+L-1}]|b_k,\pi]$ averaging the PDF's of the trajectories corresponding to each sequence of possible actions dictated by the observations.

5.3 Investigating Chance Constrained BMDP

We now examine the expression $\mathrm{P}(\mathbf{1}_{\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}}=1|b_k,\pi)$ from another angle and reformulate it in the context of posterior beliefs as it cannot be used in belief-dependent solvers in its current form. Such an extension has not been done previously, to the best of our knowledge.

In CC, as we will further see in Lemma 3, the safe event $\{x\in\mathcal{X}^{\text{safe}}\}$ impacts the belief update. Therefore, the posterior beliefs differ in the belief trees used for rewards and CC calculation. In the *rewards tree*, the beliefs are as in (5), whereas in the CC tree, as we will see shortly, we have two types of beliefs. The belief obtained from safe beliefs is

$$\bar{b}_\ell(x_\ell)\triangleq\mathbb{P}\big(x_\ell|b_k,a_{k:\ell-1},z_{k+1:\ell},\textstyle\bigcap_{i=k}^{\ell-1}\{x_i\in\mathcal{X}_i^{\text{safe}}\}\big)=\mathbb{P}\big(x_\ell|\bar{b}_{\ell-1}^{\text{safe}},a_{\ell-1},z_\ell\big), \tag{48}$$

and the safe belief given by

$$\bar{b}_\ell^{\text{safe}}(x_\ell)\triangleq\mathbb{P}\big(x_\ell|b_k,\pi,z_{k+1:k+\ell},\textstyle\bigcap_{i=k}^{\ell}\{x_i\in\mathcal{X}_i^{\text{safe}}\}\big)=\mathbb{P}\big(x_\ell|h_\ell,\bigcap_{i=k}^{\ell}\{x_i\in\mathcal{X}_i^{\text{safe}}\}\big)=\mathbb{P}\big(x_\ell|\bar{b}_{\ell-1}^{\text{safe}},a_{\ell-1},z_\ell,\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big). \tag{49}$$

Please relate the (49) to (5). This aspect is ignored by [30]. Nevertheless, such a disparity is present in discrete and continuous domains altogether. Note that also the definitions (48) and (49) we assumed that robot environment map is given. In most general case the workspace $\mathcal{X}_\ell$ is a part of the state for any ℓ.

Let us present a lemma which will shed light on the relation between the conventional formulation of safety constraint, the CC, and the posterior beliefs. To improve readability let us introduce yet another Bernoulli variable $\iota_\ell(\omega)\triangleq\mathbf{1}_{\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}}(\omega)$. Recall that $\mathbf{1}_{\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}}=\bigwedge_{\ell=k}^{k+L}\iota_\ell$.

Lemma 3 (Average over the posteriors obtained from the safe priors)**.**

$$\mathrm{P}\big(\textstyle\bigcap_{\ell=k}^{k+L}\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}|b_k,\pi\big)=\mathrm{P}\big(\big(\bigwedge_{\ell=k}^{k+L}\iota_\ell\big)=1|b_k,\pi\big)=\mathrm{P}\big(\iota_k=1|b_k\big)\mathbb{E}_{z_{k+1}}\big[\mathrm{P}\big(\iota_{k+1}=1|\bar{b}_{k+1}\big)\mathbb{E}_{z_{k+2}}\big[\mathrm{P}\big(\iota_{k+2}=1|\bar{b}_{k+2}\big)\cdots$$
$$\mathrm{P}\big(\iota_{k+L-1}=1|\bar{b}_{k+L-1}\big)\mathbb{E}_{z_{k+L}}\big[\mathrm{P}\big(\iota_{k+L}=1|\bar{b}_{k+L}\big)\big|a_{k+L-1},\bar{b}_{k+L-1}^{\text{safe}}\big]\cdots\big|a_{\ell+1},\bar{b}_{k+1}^{\text{safe}}\big]\big|a_k,b_k^{\text{safe}}\big]. \tag{50}$$

where $\bar{b}_\ell{=}\psi(\bar{b}^{\text{safe}}_{\ell-1}, a_{\ell-1}, z_\ell)$ which is different than $b_\ell{=}\psi(b_{\ell-1}, a_{\ell-1}, z_\ell))$ used in (19). We provide the proof in Appendix A.4. Here, ψ is a method for Bayesian belief update. Further details can be found in Appendix B. The connection between (49) and (48) is

$$\bar{b}^{\text{safe}}(x) \triangleq \frac{\mathbf{1}_{\{x\in\mathcal{X}^{\text{safe}}\}}(x)\bar{b}(x)}{\int_{\xi\in\mathcal{X}}\mathbf{1}_{\{\xi\in\mathcal{X}^{\text{safe}}\}}(\xi)\bar{b}(\xi)\mathrm{d}\xi}, \tag{51}$$

i.e., we nullify the unsafe portion of the belief and re-normalize.

From Lemma 3, we elicit two facts. The first one is that the CC operates on the level of **sub-policy trees** (See Fig. 7b). The second one is that each sub-policy-tree in (50) differs from a sub-policy-tree used for reward calculation. The observations come from another distribution, and the belief definitions (48), (49) is also different from (5), resulting in disparate from those used for reward calculation beliefs (See Fig. 1). Difference in belief definitions can also be viewed as the difference in belief update originating from making the belief safe (51). However, the root belief b_k is common to both belief trees, the one used for the rewards as in Fig 1 and one used for CC calculation as in Fig. 7b. To the best of our knowledge, such a difference in belief definitions is considered for the first time. Returning to our formulation (29) for the moment and comparing to (50), we highlight that our variant (29) is truly distribution aware as it counts the number of safe posteriors because of the indicator outside the inequality involving the probability value.

Further we will see that because we are dealing with expectations in (50) the disparity in the conditioning of PDF of observations in (50) and (3) is fixable using Importance Sampling (IS). Therefore, we can use the observations sampled for the *rewards tree*. Moreover, the reformulation (50) allows to utilize general belief-dependent operators over the beliefs defined as in (48) if we regard only the right hand side of (50). Still it makes sense only in the context of safety, e.g, (20) and not Information related operators as, for example, (25). This is because the definitions of the beliefs (48) and (49) appearing in (50).

5.4 Upgrade One: Probabilistic Constraint with Safe Trajectories

From the preceding discussion, we devise that we can plug the beliefs defined as in (48) (given the safe arrival to previous time instance event) and (49) into our PC (17) with (19) as such

$$\begin{aligned}&\mathbf{1}_{\{\mathrm{P}(\iota_k=1|b_k)\geq\delta\}}(b_k)\underset{z_{k+1}}{\mathbb{E}}\Big[\mathbf{1}_{\{\mathrm{P}(\iota_{k+1}=1|\bar{b}_{k+1})\geq\delta\}}\underset{z_{k+2}}{\mathbb{E}}\Big[\mathbf{1}_{\{\iota_{k+2}=1|\bar{b}_{k+2})\geq\delta\}}\cdots \\ &\cdots\underset{z_{k+L-1}}{\mathbb{E}}\Big[\mathbf{1}_{\{\mathrm{P}(\iota_{k+L-1}=1|\bar{b}_{k+L-1})\geq\delta\}}\underset{z_{k+L}}{\mathbb{E}}\Big[\mathbf{1}_{\{\mathrm{P}(\iota_{k+L}=1|\bar{b}_{k+L})\geq\delta\}}\Big|\bar{b}^{\text{safe}}_{k+L-1},\pi\Big]\Big|\bar{b}^{\text{safe}}_{k+L-2},\pi\Big]\cdots\Big|\bar{b}^{\text{safe}}_{k+1},\pi\Big]\Big|b^{\text{safe}}_k,\pi\Big]\geq 1-\epsilon.\end{aligned} \tag{52}$$

The formulation pictured by (52) is novel to the best of our knowledge. In the setting of policies we further assume that $\epsilon{=}0$ and solve the

$$\begin{aligned}&a^*_k\in\arg\max_{a_k\in\mathcal{A}}Q^{\pi^*}(b_k,a_k;\rho)\quad\text{subject to}\\ &\mathrm{P}\big(\big(\textstyle\prod_{\ell=k}^{k+L}\mathbf{1}_{\bar{A}^\delta_\ell}(\bar{b}_\ell)\big)=1|b_k,a_k,\pi^*_{k+1:k+L-1}\big)=1,\end{aligned} \tag{53}$$

where $\bar{A}^\delta_\ell\triangleq\{\bar{b}_\ell{:}\bar{b}_\ell\in\bar{\mathcal{B}}_\ell,\phi(\bar{b}_\ell){\geq}\delta\}$ and $\phi(\bar{b}_\ell){=}\mathrm{P}(\iota_\ell{=}1|\bar{b}_\ell){=}\mathrm{P}(\iota_\ell{=}1|h_\ell,\bigcap_{i=k}^{\ell-1}\{x_i{\in}\mathcal{X}^{\text{safe}}_i\})$. The belief-dependent operator here is the probability to be safe given the history and the safe arrival to previous time instance event, namely $\bigcap_{i=k}^{\ell-1}\{x_i{\in}\mathcal{X}^{\text{safe}}_i\}$. The space $\bar{\mathcal{B}}_\ell$ is the space of the beliefs defined in accord to (48) and reachable at time instance ℓ from b_k. In the OL setting instead of constraint as in (41) we will need to approximate (52). We show how to do that in Section 6.1.

5.5 Recasting the CC using Execution Risk

Let us restate the definition [30] of future Execution Risk (ER), namely $\mathrm{er}_k(b_k(h_k),\pi)$ (belief and policy dependent operator). The CC can be recast as $\mathrm{er}_k(b_k(h_k),\pi){\leq}\Delta(h_k)$. Recall that b_k common for the CC and the rewards objectives,

$$\mathrm{er}_k(b_k(h_k),\pi)\triangleq 1-\mathrm{P}\big(\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}^{\text{safe}}_\ell\}|b_k,a_k,\pi_{(k+1)+}\big)\leq\Delta(h_k)\triangleq 1-\delta(h_k), \tag{54}$$

where by $\delta(h_k)$ we denote threshold given in planning session and also the threshold of our approach (12). Since in this paper we deal with deterministic polices we sometimes will write $\mathrm{er}_k(b_k(h_k),a_k,\pi)$ to emphasize that the action a_k has been determined. Moreover, please note that by definition ER (54) is time instance dependent. This is because the trajectory is until time $k{+}L$. The risk at the k-th time step $r_b(b_k)$ is defined by

$$\mathrm{P}\big(\{x_k\notin\mathcal{X}^{\text{safe}}_k\}|b_k\big)=\textstyle\int_{x_k}b_k(x_k)\mathbf{1}_{\{x_k\notin\mathcal{X}^{\text{safe}}_k\}}(x_k)\mathrm{d}x_k=r_b(b_k). \tag{55}$$

Note that $\mathrm{P}\big(\{x_k{\in}\mathcal{X}^{\text{safe}}_k\}|b_k\big)=\int_{x_k}b_k(x_k)\mathbf{1}_{\{x_k\in\mathcal{X}^{\text{safe}}_k\}}(x_k)\mathrm{d}x_k=1-r_b(b_k)$.

Lemma 4 (Recasting). *The CC can be represented recursively and two threshold conditions are equivalent*

$$\mathrm{P}\big(\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}|b_k,\pi\big)=\mathrm{P}\big(\{x_k\in\mathcal{X}_k^{\text{safe}}\}|b_k\big)\underset{z_{k+1}}{\mathbb{E}}\Big[\underbrace{\mathrm{P}\big(\{\tau_{k+1}\in\times_{\ell=k+1}^{k+L}\mathcal{X}_\ell^{\text{safe}}\}|\bar{b}_{k+1},\pi,\big)}_{\geq\delta(h_{k+1})}|b_k,\{x_k\in\mathcal{X}_k^{\text{safe}}\},a_k,\pi\Big]\geq\delta(h_k)\quad(56)$$

$$\mathrm{er}_k(b_k(h_k),a_k,\pi)=\big(r_b(b_k)+(1-r_b(b_k))\underset{z_{k+1}}{\mathbb{E}}\big[\underbrace{\mathrm{er}_{k+1}(\bar{b}_{k+1},\pi)}_{\leq\Delta(h_{k+1})}|\{x_k\in\mathcal{X}_k^{\text{safe}}\},b_k,\pi,a_k\big]\big)\leq\Delta(h_k),\quad(57)$$

where $\Delta(h_\ell)\triangleq1-\delta(h_\ell)$ *and* $\mathrm{er}_\ell(\bar{b}_\ell,\pi)\triangleq1-\mathrm{P}\big(\{\tau_\ell\in\times_{i=\ell}^{k+L}\mathcal{X}_i^{\text{safe}}\}|b_\ell,\pi\big)\ \forall\ell\in[k:k+L]$.

We provide the proof in Appendix A.5. From (57) we infer two important aspects. The first one is that execution risk is recursive and if the CC was imposed merely from the root of the belief tree due to Bellman update each $\mathrm{er}_\ell(\bar{b}_\ell,\pi)$ for $\ell\in[k+1:k+L]$ will be large since it corresponds to an unconstrained optimal action. Therefore, it is highly likely that at the root there will be no feasible action. Further in Section 6.2 we restate how the future thresholds are found by [30] to ensure feasibility of CC at the root h_k of the belief tree by adjustment of $\Delta(h_\ell)$ at each future history h_ℓ. Still it is not for sure that feasible action will be at the root. We reiterate again that the safe event in likelihood of observations in (57) is taken into account in [30], but the difference in belief definitions (49) and (5) is not.

5.6 Implicit Access To Non-terminal Posteriors

The CC is enforced from each nonterminal belief with **adapted** threshold per history $\Delta(h_\ell)$ to ensure feasibility at h_k with $\Delta_k(h_k)$ considered in the planning session in time instance k. We defer the discussion of how to set the threshold to Section 6.2. For the first time in literature, to the best of our knowledge, we consider the difference in belief definitions in the belief tree used for the rewards (belief defined by (5)) and the CC (belief defined by (49)). Hence, we shall decide in which belief tree (pictorially Fig. 1 or Fig. 7b) to threshold each posterior. We threshold each subtree represented by (50) (Fig. 7b). The CC for belief $\bar{b}_\ell$ reads

$$\mathrm{P}\big(\textstyle\bigcap_{i=\ell}^{k+L}\{x_i\in\mathcal{X}_i^{\text{safe}}\}|\bar{b}_\ell,\pi\big)=\mathrm{P}\big(\big(\bigwedge_{i=\ell}^{k+L}\iota_i\big)=1|\bar{b}_\ell,\pi\big)\geq\delta(h_\ell)\ \forall\bar{b}_\ell(h_\ell),\ \ell\in k:k+L.\quad(58)$$

where $0\leq\delta\leq1$. Due to fact that $\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\supseteq\bigcap_{i=\ell}^{k+L}\{x_i\in\mathcal{X}_i^{\text{safe}}\}$ it holds that

$$\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|\bar{b}_\ell\big)\geq\mathrm{P}\big(\textstyle\bigcap_{i=\ell}^{k+L}\{x_i\in\mathcal{X}_i^{\text{safe}}\}|\bar{b}_\ell,\pi\big)\quad(59)$$

the (58) implicitly constraints $\mathrm{P}(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|\bar{b}_\ell\big)$ with $\delta(h_\ell)$. In fact, we never know which actual threshold each posterior belief shall be larger than or equal to. In contrast, in our formulation (19) we require being larger or equal to constant $\delta(h_k)=1-\Delta(h_k)$ solely from the multiplicands of (19).Using the (59) we also can prune the policy π in CC if

$$\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|\bar{b}_\ell(h_\ell)\big)\overset{?}{\leq}\delta(h_\ell).\quad(60)$$

This is equivalent to leveraging the relation $r_b(\bar{b}_\ell)\leq\mathrm{er}_\ell(\bar{b}_\ell,\pi)$ and prune if $\Delta(h_\ell)\leq r_b(\bar{b}_\ell(h_\ell))$. Importantly, this condition means that does not exist policy π such that the condition (58) is met for $\bar{b}_\ell$.

5.7 Upgrade Two: Future Subtrees Resolution in PC

In light of the previous discussion we suggest to constrain also terminal beliefs. As a consequence, we obtain the following reformulation the PC (17) with (19). Set $\epsilon=0$ in (10) and δ instead of being constant per planning session as in (16) is a general function of history. The chance-constrained objective materializes as

$$\begin{aligned}a_k^*\in\arg\max_{a_k\in\mathcal{A}}Q^{\pi^*}(b_k,a_k;\rho)\quad\text{subject to}\\ \mathrm{P}\big(\big(\mathbf{1}_{\bar{A}_{k+L}}(\bar{b}_{k+L},h_{k+L})\textstyle\prod_{\ell=k}^{k+L-1}\mathbf{1}_{\bar{A}_\ell}(\pi^*,\bar{b}_\ell,h_\ell)\big)=1|b_k,a_k,\pi^*_{k+1:k+L-1}\big)=1,\end{aligned}\quad(61)$$

where the $\bar{\mathcal{B}}_\ell$ is the space of reachable from b_k beliefs defined as (48) corresponding to histories and the sets are

$$\bar{A}_\ell\triangleq\{\pi,\bar{b}_\ell,h_\ell{:}\bar{b}_\ell\in\bar{\mathcal{B}}_\ell,\varphi_\ell(\bar{b}_\ell(h_\ell),\pi)\geq\delta(h_\ell)\},\qquad\bar{A}_{k+L}\triangleq\{\bar{b}_{k+L},h_{k+L}{:}\bar{b}_{k+L}\in\bar{\mathcal{B}}_{k+L},\phi(\bar{b}_{k+L}(h_{k+L}))\geq\delta(h_{k+L})\}.\quad(62)$$

The relevant indicator $\mathbf{1}_{\bar{A}_\ell}(\pi,\bar{b}_\ell,h_\ell)$ accepts also history h_ℓ to check if the triple $(\pi,\bar{b}_\ell,h_\ell)\in\bar{A}_\ell$ in the set $\bar{A}_\ell$. The operator $\varphi_\ell(\bar{b}_\ell(h_\ell),\pi)$ operating at the time instance ℓ also accepts the policy π as input and it is defined as a CC enforced onto $\bar{b}_\ell(h_\ell)$ with threshold $\delta(h_\ell)$. The operator $\varphi_\ell(\bar{b}_\ell(h_\ell),\pi)$ expands a future **sub-tree.**

$$\varphi_\ell(\bar{b}_\ell(h_\ell),\pi)=1-\mathrm{er}_\ell(\bar{b}_\ell(h_\ell),\pi)=\mathrm{P}\big(\textstyle\bigcap_{i=\ell}^{k+L}\{x_i\in\mathcal{X}_i^{\text{safe}}\}|\bar{b}_\ell,\pi\big)=\mathrm{P}\big(\bigcap_{i=\ell}^{k+L}\{x_i\in\mathcal{X}_i^{\text{safe}}\}|h_\ell,\bigcap_{i=k}^{\ell-1}\{x_i\in\mathcal{X}_i^{\text{safe}}\},\pi\big)\quad(63)$$

and at the terminal beliefs we have that

$$\phi(\bar{b}_{k+L})=\mathrm{P}\big(\{x_{k+L}\in\mathcal{X}_{k+L}^{\mathrm{safe}}\}|\bar{b}_{k+L}\big)=\mathrm{P}\big(\{x_{k+L}\in\mathcal{X}_{k+L}^{\mathrm{safe}}\}|h_{k+L},\bigcap_{i=k}^{L-1}\{x_i\in\mathcal{X}_i^{\mathrm{safe}}\}\big). \tag{64}$$

Note that in (53) we threshold all the beliefs with the same $\delta(h_k)$ per planning session as described in Section 3.2. This is in contrast to (61) where each $\delta(h)=1-\Delta(h)$ is a result of heuristics described in Section 6.2. Importantly, if we set in (61) $\delta(h_\ell)=0$ for $\ell>k$, we will obtain CC solely at the root with $\delta(h_k)$ specified outside. As we will further see, due to the usage of Bellman optimality in future times, this option does not leave feasible actions at the root b_k of the belief tree and, thereby, is not plausible.

5.8 Conservatism of CC

In this section, we discuss the conservatives of the problem depicted by (61) relative to (53). We need to define what it means to be conservative in the first place.

Definition 1. (Conservatism) Let $a^\dagger$ be solution of (53) with best future tree policy $\pi^*_{(k+1)+}$ and $a^\star$ the solution of (61) with the best future tree policy $\mu^*_{(k+1)+}$. We say that (61) is conservative with respect to (53) if

$$Q^{\pi^*}(b_k,a_k^\dagger;\rho)\geq Q^{\mu^*}(b_k,a_k^\star;\rho). \tag{65}$$

In (53) the δ is constant, namely $\delta(h_k)\equiv\delta(h_\ell)\ \forall\ell\geq k,h_\ell$. It is, also, holds that $\phi(\bar{b}_\ell(h_\ell))\geq\varphi_\ell(\bar{b}_\ell(h_\ell),\pi)$ for any $\ell\in[k{:}k+L]$. We can conclude, using the (59) that, if the $\delta(h_k)\leq\delta(h_\ell)$ in the (61), so the (61) is indeed conservative with respect to (53). Recalling that $\Delta(h)=1-\delta(h)$, the reciprocal condition for ER formulation is $\Delta(h_k)\geq\Delta(h_\ell)$.

5.9 Chance-constrained Adaptive Open Loop Continuous ρ-POMDP

In some problems, as mentioned in [14], sampling from the motion model can be costly in terms of computation time. Motivated by this key insight, we pay attention that the adaptive approach with minor adjustments applies also to the m trajectories approximated CC in the setting of POMDP with static action sequences. To approximate the CC given a candidate action sequence $a_{k:k+L-1}$ we shall draw the laces of the trajectories $\tau_k\sim b_k(x_k)\prod_{\ell=k}^{k+L-1}\mathbb{P}_{\mathrm{T}}(x_{\ell+1}|x_\ell,a_\ell)$. This is a direct result of Lemma 2, specifically, the equation (47). Similar to our PC sample approximation in the OL setting and multiplicative form (41) here we have the following objective and the CC

$$a^*_{k+}\in\arg\max_{a_{k+}\in\mathcal{A}_k}\frac{1}{m}\sum_{l=1}^{m}\sum_{\ell=k}^{k+L-1}\rho(b_\ell^l,a_\ell,b_{\ell+1}^l)\quad\text{subject to} \tag{66}$$

$$\delta\leq\hat{\mathrm{P}}^{(m)}(\mathbf{1}_{\left\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\mathrm{safe}}\right\}}=1|b_k,a_{k+}), \tag{67}$$

where $\hat{\mathrm{P}}^{(m)}(\mathbf{1}_{\left\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\mathrm{safe}}\right\}}=1|b_k,a_{k+})\triangleq\frac{1}{m}\sum_{l=1}^{m}\mathbf{1}_{\left\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\mathrm{safe}}\right\}}(\tau_k^l)$. The adaptive approach for (67) with lower and upper bounds materializes as

$$\delta\overset{?}{\leq}\tfrac{1}{m}\textstyle\sum_{i=1}^{n}\mathbf{1}_{\left\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\mathrm{safe}}\right\}}(\tau_k^l)\leq\tfrac{1}{m}\sum_{l=1}^{m}\mathbf{1}_{\left\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\mathrm{safe}}\right\}}(\tau_k^l) \tag{68}$$

$$\tfrac{1}{m}\textstyle\sum_{i=1}^{m}\mathbf{1}_{\left\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\mathrm{safe}}\right\}}(\tau_k^l)\leq\tfrac{m-n}{m}+\tfrac{1}{m}\sum_{l=1}^{n}\mathbf{1}_{\left\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\mathrm{safe}}\right\}}(\tau_k^l)\overset{?}{<}\delta. \tag{69}$$

Moreover, as in the case of the multiplicative structure of the inner constraint (35), here we use properties of the indicator to stop the safety check over the trajectory τ_k if it was unsafe in some planning future time index. To specify, we define trajectory epoch variable as $e^l(\tau_k)\triangleq\mathbf{1}_{\left\{\tau_k\in\times_{\ell=k}^{k+L}\mathcal{X}_\ell^{\mathrm{safe}}\right\}}(\tau_k^l)=\prod_{\ell=k}^{k+L}\mathbf{1}_{\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\}}(x_\ell^l)$. Similar to the situation with PC we have that $e^l(\tau_k)\leq\prod_{\ell=k}^{k+j}\mathbf{1}_{\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\}}(x_\ell^l)$ for $j\leq L$. In this approach, we again have two belief trees, similar to the situation with policies. However, now for CC, we have an MDP tree, whereas for the rewards we have a belief tree. For the rewards from each trajectory τ_k, we create the lace of the observations $z_{k+1:k+L}$ and corresponding beliefs $b_{k+1:k+L}$ for the calculation of belief-dependent rewards. A similar trick is used in [31] but in a different context. Note that this approach can be used in the setting of an uncertain map (or robot workspace) when the obstacles are represented as landmarks with some volume. To conclude we visualized two approaches PC and CC in the OL setting in Fig 8. We proceed to the summary.

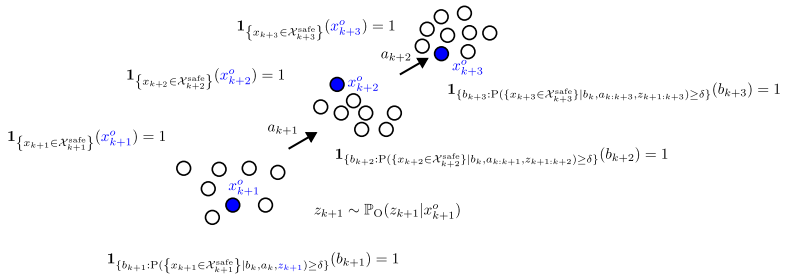


Fig. 8: Visualization of PC versus CC in OL setting with horizon L=3. In PC we constrain the beliefs. Whereas, in CC we constraint trajectories that render the observations. Purely for clarity, the beliefs here are represented by particles.

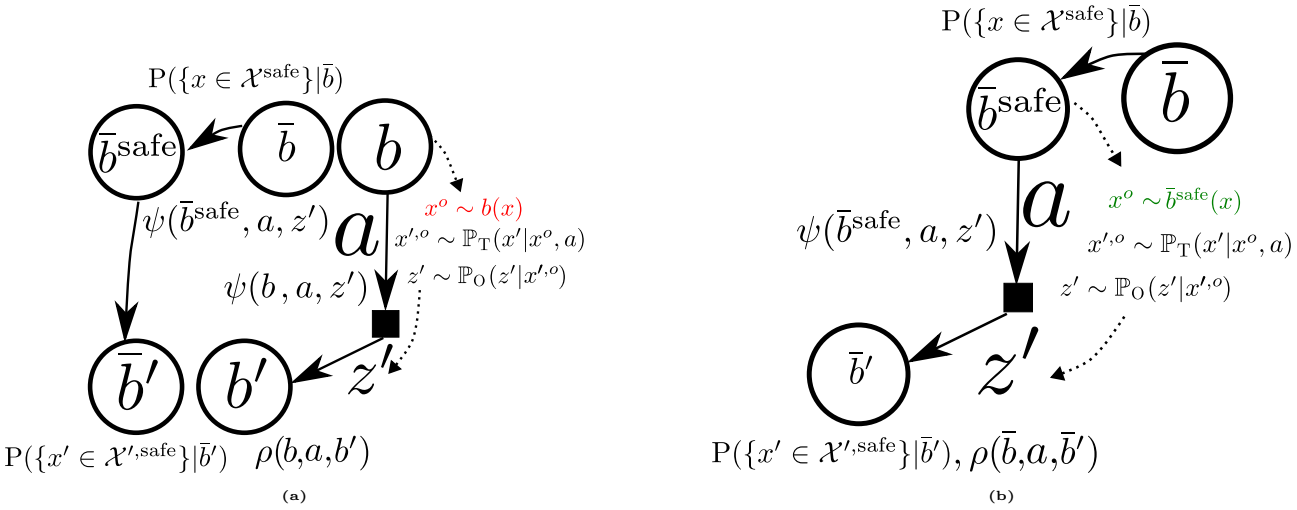


Fig. 9: **(a)** Visualization of the belief tree obtained by Importance Sampling (Section 6.1). For an actual belief at planning time instant k we set $b_k = \bar{b}_k$. Observation is sampled as such $z' \sim \mathbb{P}(z'|b, a)$. **(b)** No Importance Sampling. The same distribution of the observations and belief update is used for belief dependent rewards and CC. Also here we set $b_k = \bar{b}_k$. Observation is sampled as such $z' \sim \mathbb{P}(z'|\bar{b}, \{x \in \mathcal{X}^{\text{safe}}\}, a)$.

5.10 Summary

To summarize, the CC (61) has several key differences versus ours (17), (19).

1. Instead of looking into safe state trajectories in CC, we are dealing with safe posterior beliefs trajectories in PC. Our approach uses the same distribution of the observations and the definition of the beliefs each step ahead as for the reward. At the same time, the CC builds upon the distribution of observations conditioned also on the safe events and different belief definition from the belief used for the rewards. These two distributions and the belief definitions are identical only if the robot workspace is completely safe, e.g., with no obstacles at all, or the belief has finite support lying in the safe space or $\Delta(h) \equiv 0$ and $\delta(h) = 1 - \Delta(h) = 1$ for any history h simulated in planning. This way feasible belief shall be already safe.
2. In CL setting, our PC with ϵ=0 and the multiplicative form of inner constraint allows efficient exact pruning while CC does not.
3. In addition, in a nonparametric setting not always the belief can be made safe. This problem is a direct result of a particle representation. Hence, in such scenarios, one shall use our PC.

6 Approach to Chance-constrained Continuous ρ-POMDP

The investigation of CC merged with a general belief-dependent reward in continuous domains has led us to need an algorithmic extension. As mentioned, there are two prominent online approaches for solving a continuous POMDP with belief-dependent rewards in a nonparametric domain: SS [18], and PFT-DPW [34]. In continuous domains and belief-dependent rewards, it is unclear how to apply a heuristics guided forward search described by [30]. Instead of using the heuristics, we utilize the Bellman principle to resolve that issue. We aim to solve the sample approximation of (61).

In [30], the discrepancy in observation distribution $\mathbb{P}(z_{\ell+1}|b_\ell, a_\ell)$ for rewards and $\mathbb{P}(z_{\ell+1}|\bar{b}_\ell, \{x_\ell \in \mathcal{X}_\ell^{\text{safe}}\}, a_\ell)$ for CC addressed by considering a discrete and finite observation space and exhaustively expanding all the observations and

calculating appropriate likelihoods. Such an approach is not possible in a continuous setting. Additionally, [30] in $\mathbb{P}(z_{\ell+1}|\bar{b}_\ell, \{x_\ell \in \mathcal{X}_\ell^{\text{safe}}\}, a_\ell)$ instead of using $\bar{b}_\ell$ sequentially updated from safe belief (48) use b defined in accord to (5). To tackle all these issues, we resort to Importance Sampling (IS) such that only a single set of observations is maintained. Let us emphasize that such a problem has not been addressed so far. Moreover, the disparity of the belief definitions was not addressed at all. Note that this issue does not exist in our probabilistic approach (Alg. 1). Next, we contribute an **Importance Sampling** based approach for chance-constrained **continuous** POMDP.

6.1 IS Approach for Chance-constrained Continuous POMDP

As we have seen in Lemma 3 and equation (3) the distributions of the observations of the CC and the action value function are different (61), as well as the belief update.

Let us observe a single step ahead in arbitrary future time index $\ell \in [k{+}1{:}k{+}L{-}1]$. For $\ell{=}k$ we just draw the observations because $b_k \equiv \bar{b}_k^{\text{safe}}$. Since we draw observations sequentially, the extension to an arbitrary horizon is straightforward. In case of the objective function, the desired PDF is $\mathbb{P}(z_{\ell+1}|b_\ell, a_\ell)$, whereas for the CC we are dealing with $\mathbb{P}(z_{\ell+1}|\bar{b}_\ell, \{x_\ell \in \mathcal{X}_\ell^{\text{safe}}\}, a_\ell)$. Only in the time index k we have that b_k is shared in the conditioning event of both distribution densities of observations (Fig. 7b). In general $\bar{b}_\ell(x_\ell)$ is defined by (48) and $b_\ell(x_\ell)$ in accordance to (5). Note that equivalent way to write these distribution densities is $\mathbb{P}(z_{\ell+1}|b_k, \pi, z_{k+1:k+\ell}, \bigcap_{i=k+1}^{\ell}\{x_i \in \mathcal{X}_i^{\text{safe}}\})$ and $\mathbb{P}(z_{\ell+1}|b_k, \pi, z_{k+1:\ell})$. With a growing horizon, these PDFs can significantly depart from each other. As a result of this discrepancy, there are two different distributions of future observations (since conditioned on different events). With CC, we should have, thus, sampled from each and effectively constructed two belief trees rooted at b_k. Putting aside that it would be an enormous computational burden, the question of how to apply a consistent policy in both trees requires clarification. We reiterate that our PC formulation (see Section 3) does not exhibit this discrepancy.

To avoid construction of two belief trees, we suggest to construct a single belief tree where observations are sampled from $\mathbb{P}(z_{\ell+1}|b_k, \pi, z_{k+1:\ell})$ and properly re-weighted via IS for the evaluation of the CC. Specifically, suppose we sampled m_d samples $\{z_{\ell+1}^j\}_{j=1}^{m_d} \sim \mathbb{P}(z_{\ell+1}|b_k, \pi, z_{k+1:\ell})$. From now on, we can think about

$$\hat{\mathbb{P}}_{(m_d)}(z_{\ell+1}|b_k, \pi, z_{k+1:\ell}) = \frac{1}{m^d}\sum_{j=1}^{m_d}\delta(z_{\ell+1} - z_{\ell+1}^j), \tag{70}$$

as the PDF of the discrete probability (Fig. 9a). Importantly, in this case $\mathbb{P}(z_{\ell+1}|b_k, \pi, z_{k+1:\ell}, \bigcap_{i=k}^{\ell}\{x_i \in \mathcal{X}_i^{\text{safe}}\})$ is **absolutely continuous** with respect to $\mathbb{P}(z_{\ell+1}|b_k, \pi, z_{k+1:\ell})$. Let us prove that.

Lemma 5 (Absolute Continuity)**.** $\mathbb{P}(z_{\ell+1}|b_k, \pi, z_{k+1:\ell}, \bigcap_{i=k}^{\ell}\{x_i \in \mathcal{X}_i^{\text{safe}}\}) \ll \mathbb{P}(z_{\ell+1}|b_k, \pi, z_{k+1:\ell})$.

We provide the proof in Appendix A.6. Since the absolute continuity holds, we can safely use IS. Leveraging IS, we obtain the desired PDF utilizing the same samples through the following manipulation

$$\hat{\mathbb{P}}_{(m_d)}\big(z_{\ell+1}|b_k, \pi, z_{k+1:\ell}, \bigcap_{i=k}^{\ell}\{x_i \in \mathcal{X}_i^{\text{safe}}\}\big) = \frac{1}{\sum_{j=1}^{m_d} w_{\ell+1}^{z,j}}\sum_{j=1}^{m_d} w_{\ell+1}^{z,j}\delta(z_{\ell+1} - z_{\ell+1}^j), \tag{71}$$

where the j-th weight is given by

$$w_{\ell+1}^{z,j} = \frac{1}{m_d}\frac{\mathbb{P}(z_{\ell+1}=z_{\ell+1}^j|b_k, \pi, z_{k+1:\ell}, \bigcap_{i=k}^{\ell}\{x_i \in \mathcal{X}_i^{\text{safe}}\})}{\mathbb{P}(z_{\ell+1}=z_{\ell+1}^j|b_k, \pi, z_{k+1:\ell})}. \tag{72}$$

In Appendix C we specify expressions for the nominator and denominator.

Let us clarify again that with the proposed IS-based approach, the sampled observations are used for both the objective and the CC (Fig. 9a). However, for the CC we re-weight the samples using Importance weight to obtain the correct expected value according to (50). To sample sequentially the observations we use the beliefs from the rewards tree (See Fig. 9a) defined by (5). We now present a sample approximation of ER (54), (57). Suppose we approximate the expectation from Lemma 4 by samples from $\mathbb{P}(z_{\ell+1}|b_k, \pi, z_{k+1:\ell})$. We obtain, using (71)

$$\text{er}_\ell(\bar{b}_\ell, a_\ell, \pi) = r_b(\bar{b}_\ell) + (1 - r_b(\bar{b}_\ell))\int_{z_{\ell+1}}\frac{1}{\sum_{j=1}^{m_d} w_{\ell+1}^{z,j}}\sum_{j=1}^{m_d} w_{\ell+1}^{z,j}\delta(z_{\ell+1} - z_{\ell+1}^j)\text{er}_{\ell+1}(\bar{b}_{\ell+1}, \pi)\text{d}z_{\ell+1} = \tag{73}$$

$$r_b(\bar{b}_\ell) + (1 - r_b(\bar{b}_\ell))\frac{1}{\sum_{j=1}^{m} w_{\ell+1}^{z,j}}\sum_{j=1}^{m} w_{\ell+1}^{z,j}\text{er}_{\ell+1}(\bar{b}_{\ell+1}(h_\ell a_\ell z_{\ell+1}^j), \pi) \tag{74}$$

Remark: We also can utilize this approach to approximate (52) in the OL setting. We however suggest something else in Section 7.

6.2 Necessary Condition proposed by [30] for Feasibility of Chance Constraint

In this section we extend the necessary condition for feasibility of CC proposed by [30] to continuous spaces through IS. This is one of the building blocks of our approach to solve continuous belief-dependent chance-constrained POMDP, see Alg. 3 and Alg. 8. We endow our IS approach with a pruning mechanism. Recall that the belief $b(h)$ is indexed by history h in the belief tree.

The paper [30] utilizes the necessary condition for the feasibility of an action. For completeness let us present the following Lemma, which is merely rewriting with our notations the pruning condition from [30], extended to the continuous spaces with IS and considering difference in the belief definitions (5) with (48) and (49).

Lemma 6 (Necessary Condition for Feasibility of CC). *Fix* $0\leq\Delta(h_\ell)\leq 1$ *and* $a_\ell\in\mathcal{A}$. *Suppose that*

$$\mathrm{er}_\ell(\bar{b}_\ell(h_\ell), a_\ell, \pi_{(\ell+1)+}) \leq \Delta(h_\ell) \quad \texttt{necessary condition.} \tag{75}$$

The following holds for every child $\forall i \in 1 : m_d$ *of* $h_\ell a_\ell$ *for any* $\ell = k : k + L - 1$

$$\mathrm{er}_{\ell+1}(\bar{b}_{\ell+1}(h_\ell a_\ell z^i_{\ell+1}), \pi) \leq \frac{1}{w^{z,i}_{\ell+1}}\big(\frac{\Delta(h_\ell)-r_b(\bar{b}_\ell(h_\ell))}{1-r_b(\bar{b}_\ell(h_\ell))} - \sum_{\substack{j=1\\ j\neq i}}^{m} w^{z,j}_{\ell+1}\mathrm{er}_{\ell+1}(\bar{b}_{\ell+1}(h_\ell a_\ell z^j_{\ell+1}), \pi)\big), \tag{76}$$

We provide the proof in Appendix E. Using the fact that $0\leq r_b(\bar{b}_{\ell+1})\leq \mathrm{er}_{\ell+1}(\bar{b}_{\ell+1}, \pi)$ we have the necessary pruning condition.

$$r_b(\bar{b}^i_{\ell+1}) \leq \frac{1}{w^{z,i}_{\ell+1}}\big(\frac{\Delta(h_\ell)-r_b(\bar{b}_\ell(h_\ell))}{(1-r_b(b_\ell(h_\ell)))} - \sum_{\substack{j=1\\ j\neq i}}^{m} w^{z,j}_{\ell+1} r_b(\bar{b}^j_{\ell+1})\big), \tag{77}$$

and three intuitive options to set $\Delta(h_\ell a_\ell z^i_{\ell+1}))$. They are specified as

1.
$$\Delta(h_\ell a_\ell z^i_{\ell+1})) = \frac{1}{w^{z,i}_{\ell+1}}\big(\frac{\Delta(h_\ell)-r_b(\bar{b}_\ell(h_\ell))}{1-r_b(\bar{b}_\ell(h_\ell))} - \sum_{\substack{j=1\\ j\neq i}}^{m} w^{z,j}_{\ell+1} r_b(\bar{b}_{\ell+1}(h_\ell a_\ell z^j_{\ell+1}))\big); \tag{78}$$

2.
$$\Delta(h_\ell a_\ell z^i_{\ell+1})) = \frac{1}{w^{z,i}_{\ell+1}}\big(\frac{\Delta(h_\ell)-r_b(\bar{b}_\ell(h_\ell))}{1-r_b(\bar{b}_\ell(h_\ell))}\big); \tag{79}$$

3.
$$\Delta(h_\ell a_\ell z^i_{\ell+1})) = \Delta(h_\ell); \tag{80}$$

4.
$$\Delta(h_\ell) \equiv 1 \quad \forall \ell > k. \tag{81}$$

We classify the condition expressed by Eq. (78) as a necessary condition for (75) to hold. The Eq. (79) we regard as a fast necessary condition for (75) to hold. This way we hope that on the way up the tree will be feasible actions, but this is not ensured. Another possibility is to set $\Delta'(haz')=\Delta(h)$ as in (80). We consider the condition (80) as sufficient since we know from (57) that if every z' (for fixed ha) it holds that $\mathrm{er}'(haz')=\Delta(h)$ and $r_b(b(h))=0$, the a is feasible. The condition (81) enforces CC only at the root with $\Delta(h_k)$ being specified outside. Eq.(77) is used in our Alg. 3 and further in Alg. 8, specifically in line 19 and 18 respectively. Moreover, we can conclude that with (80) the problem (61) is conservative with respect to (53). This is because in this setting it holds $\Delta(h_\ell)\leq\Delta(h_k)$ (See Section 5.8). With (78) or (79) the situation shall be simulated. In this setting, we cannot conclude conservativnes of (61) with respect to (53). The (80) is equivalent to (16)

We conclude this section by observing that it is possible that $\mathrm{er}_\ell(\bar{b}_\ell(h_\ell), \pi) > \Delta(h_\ell)$ but (77) still holds (merely necessary condition). This is in striking contrast to our PC pruning, as we proved in Theorem 1.

6.3 The Algorithms for CC

In this section, we describe in detail the algorithms for the solution of chance-constrained continuous ρ-POMDP in the setting of policies (Closed Loop) and static candidate action sequences (Open Loop). Note, in the setting of policies, since we are using only necessary condition for pruning described in section 6.2 and Alg 5 on the way down the tree (line 19 in Alg. 3 and line 18 Alg. 8), we still need to verify the constraint on the way up (line 28 in Alg. 3 and line 26 Alg. 8). In both algorithms for CL, we also enforce the condition to be safe on terminal beliefs and utilize early termination using the relation described by (60). Additionally, both algorithms can be used with CC enforced solely from the root of the belief tree b_k by using (81).

6.3.1 Chance Constrained Sparse Sampling (CL)

Our solver for the chance-constrained continuous belief-dependent POMDP (Section 6.1) is formulated as Alg. 3. This algorithm is designed to solve the sample approximation of the objective portrayed by (61). In this algorithm we utilize the IS as described 6.1. In line 11 we sample observation from the belief defined by (5). In line 15 we prune action a using the condition (60). We calculate the IS approximated ER (74) at the line 27 and verify the CC at the line 28 because pruning using (77) here is only necessary condition and not sufficient. This algorithm works in accord with the scheme illustrated by Fig. 9a. Reward is calculated over the beliefs defined by (5). Further we show more efficient variant of chance-constrained approach (Alg. 8).

Algorithm 3 Chance-constrained BMDP Sparse Importance Sampling (CCSS-IS)

procedure CCSS-IS(belief: $b(h)$, belief: $\bar{b}$ depth: d, threshold: $\Delta(h)$) ▷ b as in (5) whereas $\bar{b}$ as in (48)
  if $d = 0$ **then**
    return (Null, 0, $\mathrm{P}(\{x \notin \mathcal{X}^{\text{safe}}\}|\bar{b})$) ▷ $\mathrm{P}(\{x \in \mathcal{X}^{\text{safe}}\}|\bar{b}) = 1 - r_b(\bar{b}), \mathrm{P}(\{x \notin \mathcal{X}^{\text{safe}}\}|\bar{b}) = r_b(\bar{b})$
  end if
  $(a^*, v^*, \mathrm{er}_{k+L}(\bar{b}, \pi^*)) \leftarrow$ (Null, $-\infty, 1$) ▷ $\mathrm{er}_\ell(\bar{b}, \pi^*) \leq 1$
  $\bar{b}^{\text{safe}} \leftarrow$ Make $\bar{b}$ safe ▷ obtain safe belief as in equation (49) using (51) from belief defined by (48)
  for $a \in \mathcal{A}$ **do**
    $v \leftarrow 0.0$ ▷ Initialization of Value function
    Calculate propagated belief b'^{-} from b and $\bar{b}'^{-}$ from $\bar{b}^{\text{safe}}$ applying action a ▷ Need to propagate both beliefs for weights calculation in line 12.
    for $j \in 1 : m_d$ **do**
      Sample $x^{o,j} \sim b'^{-}$ followed by $z'^{,j} \sim \mathbb{P}(z|x^{o,j})$ Observations are created using belief defined by (5) and action a.
      Calculate $w^{z,\prime,j}$ See Appendix C.
      $\bar{b}'^{,j} \leftarrow \psi(\bar{b}^{\text{safe}}, a, z'^{,j})$ ▷ Calculate posterior as in (48)
      $\Delta(haz'^{,j}) \leftarrow \Delta\text{PRIME}(\Delta(h), r_b(\bar{b}), w^{z,\prime,j})$, $\delta(haz'^{,j}) \leftarrow 1 - \Delta(haz'^{,j})$ ▷ Call Alg. 4
      if $\mathrm{P}(\{x' \in \mathcal{X}^{\text{safe},\prime}\}|\bar{b}'^{,j}) < \delta(haz'^{,j})$ **then** ▷ Equivalent to $\mathrm{P}(\{x' \notin \mathcal{X}^{\text{safe},\prime}\}|\bar{b}'^{,j}) > \Delta(haz'^{,j})$
        next action ▷ Prune action a using (60). We still do not know if $\mathrm{er}(\bar{b}'^{,j}|\pi) \leq \Delta(haz'^{,j})$. The condition (60) is only necessary but for any policy π.
      end if
    end for
    if PRUNEACTIONCHANCE($\{\bar{b}'^{,j}\}_{j=1}^{m_d}$, $\{w^{z,\prime,j}\}_{j=1}^{m_d}$, $\bar{b}$, $\Delta(h)$) **then**
      next action ▷ See Alg. 5. We still do not know if $\mathrm{er}(\bar{b}|a, \pi^*) \leq \Delta(h)$. The condition (77) is only necessary
    end if
    for $j \in 1 : m_d$ **do** ▷ At this point we have all the observations for not pruned action a, namely $\{z'^{,1}, \ldots, z'^{,m_d}\}$.
      $b'^{,j} \leftarrow \psi(b, a, z'^{,j})$ ▷ The belief defined by (5) is updated only for not pruned actions.
      $a'^{,*}$, v', $\mathrm{er}_{k+L-d+1}(\bar{b}'^{,j}, \pi^*) \leftarrow$ CCSS-IS($b'^{,j}$, $\bar{b}'^{,j}$, $d-1$, $\Delta(haz'^{,j})$) ▷ π^* is applied from time of $\bar{b}'$, $a'^{,*}$ is ignored
      $v+ = (\rho(b, a, b'^{,j}) + \gamma \cdot v')/m_d$ ▷ Reward is calculated over the beliefs defined by (5)
    end for
    $\mathrm{er}_{k+L-d}(\bar{b}, a, \pi^*) \leftarrow \mathrm{P}(\{x \notin X^{\text{safe}}\}|\bar{b}) + \mathrm{P}(\{x \in \mathcal{X}^{\text{safe}}\}|\bar{b}) \cdot \frac{1}{\sum_{j=1}^{m_d} w^{z,\prime,j}} \sum_{j=1}^{m_d} w^{z,\prime,j} \mathrm{er}_{k+L-d+1}(\bar{b}'^{,j}, \pi^*)$ ▷ Approximation of (57) using (74).
    if $\mathrm{er}_{k+L-d}(\bar{b}, a, \pi^*) \leq \Delta(h)$ and $v > v^*$ **then** ▷ CC check
      $(a^*, v^*, \mathrm{er}_{k+L-d}(\bar{b}, \pi^*)) \leftarrow (a, v, \mathrm{er}_{k+L-d}(\bar{b}, a, \pi^*))$
    end if
  end for
  return (a^*, v^*, $\mathrm{er}_{k+L-d}(\bar{b}, \pi^*)$)
end procedure

Algorithm 4 Update Δ to assure Feasibility of Chance Constraint at the root

procedure $\Delta\text{PRIME}(\Delta(h), r_b(\bar{b})$, $w^{z,\prime})$
  return the result of (79) or (80)
end procedure

6.3.2 Chance-constrained Open Loop Continuous ρ-POMDP

We now describe, in detail, Alg. 6 outlining the adaptive approach from Section 5.9 to chance-constrained ρ-POMDP. Here the CC is enforced only from the root of the belief tree. The Alg. 6 has two phases. In the first phase, it samples a minimal number of trajectories for each candidate action sequence required to **adaptively** evaluate m samples based approximation of the CC (67), as described in Section 5.9. In the second phase for each survived candidate action sequence

Algorithm 5 Necessary condition for Feasibility of Chance Constraint

procedure PRUNEACTIONCHANCE($\{\bar{b}'^{,j}\}_{j=1}^{m_d}$, $\{w^{z,\prime,j}\}_{j=1}^{m_d}$, $\bar{b}$, Δ)
  for each $\bar{b}'^{,i}$ **do**
    if (77) is not met **then**
      return true ▷ prune
    end if
  end for
  return false
end procedure

a_{k+} we use existing $n(a_{k+})$ trajectories and sample $m-n(a_{k+})$ missing trajectories. We then create observations and calculate beliefs, corresponding rewards and the objective (66).

7 Taking Safety Events to the Objective

In this section, we suggest a modification of the objective. To eliminate the discrepancy in objective and the CC (50) and PC with safe trajectories (52) we take safe events $\{x\in\mathcal{X}^{\text{safe}}\}$ of future states to the objective.

7.1 Objective Modification

The IS approach introduced in Section 6.1 converges to the theoretical solution when $m_d\to\infty$. In the planning phase, the IS needed to resolve the discrepancy due to the separation of the CC satisfaction (falling trajectories are not pushed forward in time) and the future return maximization (regular belief/observations PDF in the belief tree). However, the mechanics of IS introduces a computational burden. Let us ask another question to ameliorate the situation from the computational point of view. Can we relinquish the requirement of IS?

Specifically, say, we are using $\mathbb{P}(z_{\ell+1}|b_k,\pi,z_{k+1:\ell},\bigcap_{i=k}^{\ell}\{x_i\in\mathcal{X}_i^{\text{safe}}\})$ $\ell=k{:}k+L-1$ and corresponding beliefs $\bar{b}_{\ell+1}$ for the calculation of the belief-dependent reward. In other words, we change the conventional objective as such. Instead of using (4), we use the distribution of the observations and the belief update from the CC such that the objective takes the form of

$$U^L(b_k,\pi)\triangleq\int_{z_{k+1:k+L}}\mathbb{P}(z_{k+1}|b_k,a_k)\prod_{j=k+1}^{k+L-1}\mathbb{P}\big(z_{j+1}|b_k,\pi,z_{k+1:j},\textstyle\bigcap_{i=k}^{j}\{x_i\in\mathcal{X}_i^{\text{safe}}\}\big)\sum_{\ell=k}^{k+L-1}\rho(\bar{b}_\ell,\pi_\ell,\bar{b}_{\ell+1})\mathrm{d}z_{k+1:k+L}= \quad (82)$$

$$\sum_{\ell=k}^{k+L-1}\int_{z_{k+1:\ell+1}}\big(\mathbb{P}(z_{k+1}|b_k,a_k)\prod_{j=k+1}^{\ell}\mathbb{P}\big(z_{j+1}|b_k,\pi,z_{k+1:j},\textstyle\bigcap_{i=k}^{j}\{x_i\in\mathcal{X}_i^{\text{safe}}\}\big)\big)\rho(\bar{b}_\ell,\pi_\ell,\bar{b}_{\ell+1})\mathrm{d}z_{k+1:\ell+1}. \quad (83)$$

The above modification can be interpreted as follows. Although we calculate the belief-dependent rewards on the entire belief, following the belief-dependent reward calculation only the safe state particles of the posterior belief are pushed forward in time with action and observation. This behavior is identical to that we obtained in the CC (50) and PC with safe trajectories portrayed by (52). We face matched distribution and definition of future beliefs in (50), (52) and in the objective (82). Note that we can not write that in (83) we have the sum of the expectations because in general

$$\mathbb{P}(z_{k+1:\ell}|b_k,\pi,\bigcap_{i=k}^{\ell}\{x_i\in\mathcal{X}_i^{\text{safe}}\})=\prod_{j=k+1}^{\ell}\mathbb{P}\big(z_{j+1}|b_k,\pi,z_{k+1:j},\bigcap_{i=k}^{\ell}\{x_i\in\mathcal{X}_i^{\text{safe}}\}\big)\neq\prod_{j=k+1}^{\ell}\mathbb{P}\big(z_{j+1}|b_k,\pi,z_{k+1:j},\bigcap_{i=k}^{j}\{x_i\in\mathcal{X}_i^{\text{safe}}\}\big).$$

We marked the difference by the red color. The benefit of such a modification is significantly faster decision-making. Further, we empirically demonstrate the substantial acceleration with good performance quality.

What will be the impact of this modification on decision-making? This question has not been addressed to the best of our knowledge. We leave for future work the analysis of the above tempering with the objective. Our vision is that on the small extent of the quality of the optimal solution, utilization of the objective (82) for the reward will accelerate the performance by avoiding the need for IS in CC and the need to maintain a pair of beliefs in CC (50) and PC with safe trajectories (52). Let us reiterate that if $\delta(h)\equiv1$ for any history h simulated in the planning session such a modification is lossless for **feasible** policies. This is because in this case $b\equiv\bar{b}\equiv\bar{b}^{\text{safe}}$.

7.2 Algorithms for Modified Objective

We now present algorithms utilizing such a modified objective. Our belief dependent operator ϕ is as in (18) but applied on $\bar{b}_\ell$ defined by (48) instead of b_ℓ defined by (5), namely $\phi(\bar{b}_\ell)=\mathrm{P}(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}|\bar{b}_\ell)$.

Algorithm 6 Chance-constrained open-loop ρ-POMDP

Input: $\mathcal{A}$, b_k, h_k ▷ Set of the candidate action sequences
$a^*_{k+} \leftarrow$ undef, $\hat{V}^*_{(m)} \leftarrow -\infty$, $S \leftarrow \{\}$
for each $a_{k+} \in \mathcal{A}_k$ **do**
  for $n(a_{k+}) \in 1:m$ **do**
    Draw $x_k^{n(a_{k+})} \sim b_k(x_k)$ ▷ The beginning of the trajectory in accord to (47)
    $e^{n(a_{k+})} \leftarrow \mathbf{1}_{\{x_k \in \mathcal{X}_k^{\text{safe}}\}}(x_k^{n(a_{k+})})$
    for $\ell \in k+1 : k+L$ **do**
      Draw $x_\ell^{n(a_{k+})} \sim \mathbb{P}_T(x_\ell | x_{\ell-1}^{n(a_{k+})}, a_{\ell-1})$
      $e^{n(a_{k+})} \leftarrow e^{n(a_{k+})} \cdot \mathbf{1}_{\{x_\ell \in \mathcal{X}_\ell^{\text{safe}}\}}(x_\ell^n)$
      if $e^{n(a_{k+})} == 0$ **then**
        break
      end if
    end for
    if $\delta \leq \frac{1}{m}\sum_{l=1}^{n(a_{k+})} e^l$ **then**
      $S \leftarrow S \cup \{a_{k+}\}$ ▷ Accept the a_{k+}
      break ▷ check the next action seq.
    else if $\frac{1}{m}\sum_{l=1}^{n(a_{k+})} e^l < \delta - \frac{m-n(a_{k+})}{m}$ **then**
      break ▷ check the next action seq.
    end if
  end for
end for
for each $a_{k+} \in S$ **do** ▷ S contains all feasible a_{k+}
  expand all m laces using already drawn $\{\tau_k^l\}_{i=1}^{n(a_{k+})}$ and get $\hat{V}^{(m)}(b_k, a_{k+})$ ▷ Only here we sample observation laces
  if $\hat{V}^*_{(m)} < \hat{V}^{(m)}(b_k, a_{k+})$ **then**
    $a^*_{k+} \leftarrow a$, $\hat{V}^*_{(m)} \leftarrow \hat{\mathcal{U}}^{(m)}(b_k, a_{k+})$
  end if
end for
Return a^*_{k+}

7.2.1 Probabilistically-constrained Sparse Sampling with Safe Trajectories ($\epsilon = 0$)

In Alg. 7 we only constrain each posterior belief similar to Alg. 1. No additional checks are needed since we use our Theorem 1 for sufficient condition for feasibility. However, as can be seen in lines 11 and 18 of Alg. 7, pictorially, we utilize the scheme from Fig. 9b. In Alg. 7 in time instance $\ell{+}1$ we sample observation from $\mathbb{P}(z_{\ell+1}|b_k, \pi, z_{k+1:\ell}, \bigcap_{i=k}^{\ell}\{x_i \in \mathcal{X}_i^{\text{safe}}\})$ for $\ell{=}k{:}k{+}L{-}1$ using belief defined by (49) in time ℓ and action a_ℓ.

7.2.2 Matched Chance-constrained Sparse Sampling

Our efficient variant from Section 7.1 is summarized in Alg. 8. In this algorithm, we do not use IS. Therefore we maintain only the belief defined by (49) as in Fig. 9b. The reward is calculated over the beliefs defined by (48).

7.2.3 Arbitrary $\epsilon \in [0, 1)$ for PC Open Loop Safe Trajectories Approach

Approach from Section 5.4 enables us to employ Alg. 2 with observation laces sampled sequentially from

$$\mathbb{P}(z_{\ell+1}|b_k, a_{k:\ell}, z_{k+1:\ell}, \textstyle\bigcap_{i=k}^{\ell}\{x_i \in \mathcal{X}_i^{\text{safe}}\}) = \mathbb{P}(z_{\ell+1}|\bar{b}_\ell^{\text{safe}}, a_\ell) \quad \ell = k : k+L-1$$

using beliefs as in (49) in time ℓ and action a_ℓ instead of observations sampled from

$$\mathbb{P}(z_{\ell+1}|b_k, a_{k:\ell}, z_{k+1:\ell}) = \mathbb{P}(z_{\ell+1}|b_\ell, a_\ell) \quad \ell = k : k+L-1$$

using beliefs as in (5) and Alg. 2. Using this approach we use the PC with safe trajectories (52) instead of requiring that (29) larger or equal than $1{-}\epsilon$. Note that also in modified version of Alg. 2 we can use safety related general belief dependent operators described in Section 3.3 (Similar to reformulated CC (50)). We leave this aspect for future research.

Algorithm 7 Matched Probabilistic BMDP Sparse Sampling with Safe Trajectories (PCSSST)

1: **procedure** PCSSST(belief: $\bar{b}$, history: h, depth: d, threshold $\delta(h)$)
2: **if** $d = 0$ **then**
3: **return** (Null, 0)
4: **end if**
5: $(a^*, v^*) \leftarrow$ (Null, $-\infty$)
6: $\bar{b}^{\text{safe}} \leftarrow$ Make $\bar{b}$ safe ▷ as in equation (51)
7: **for** $a \in \mathcal{A}$ **do**
8: $v \leftarrow 0.0$, PrunedFlag $\leftarrow$ false ▷ Value function
9: Calculate propagated belief $\bar{b}'^{-}$ from $\bar{b}^{\text{safe}}$ applying action a
10: **for** m_d times **do**
11: Sample $x^o \sim \bar{b}'^{-}$ followed by $z' \sim \mathbb{P}(z|x^o)$ ▷ Matched belief, only the safe trajectories are kept
12: $\bar{b}' \leftarrow \psi(\bar{b}^{\text{safe}}, a, z')$
13: **if** $\mathrm{P}(\{x' \in \mathcal{X}^{\text{safe},\prime}\}|\bar{b}') < \delta(h)$ **then** ▷ It is possible that other operators can be used here
14: PrunedFlag $\leftarrow$ true
15: **break** ▷ Exit from observations loop and go to line 21
16: **end if**
17: $a'^{,*}$, v', $\leftarrow$ PCSSST(b',haz', $d-1$, $\delta(h)$)
18: $v+ = (\rho(\bar{b}, a, \bar{b}'^{,j}) + \gamma \cdot v')/m_d$ ▷ Reward calculated over the matched with chance constr. beliefs
19: **end for**
20: **if** PrunedFlag is false and $v > v^*$ **then**
21: $(a^*, v^*) \leftarrow (a, v)$
22: **end if**
23: **end for**
24: **return** (a^*, v^*)
25: **end procedure**

8 Simulations and Results

In this section, we study our proposed algorithms. Since the paper [6] presents a parametric method for Gaussian beliefs, it is relevant for us solely from the constraint formulation perspective. Due to the fact that our comparison will be with CC, in the setting of policies we employ CC with the future thresholds as in (80) to make (61) overconservative with respect to (53). Note that we perform a separate study of (80) versus (79) and report results in Table 4.

We demonstrate theoretical findings on two problems in continuous domains in terms of states and observations, navigation to the static goal and target tracking. Both problems are under the umbrella of Belief Space Planning with a given map. Our examination of the proposed approach has two parts.

The first part is the setting of policies. There we first verify that with CC enforced merely from the root of the belief tree in Alg. 3 and Alg. 8 by selecting the $\Delta(h)$ in accord to (81), these algorithms return no feasible solution. We then perform an ablation study of chance-constrained approaches Alg. 3, Alg. 8 with and without IS correspondingly versus our PC approach Alg. 1 and upgraded variant with safe trajectories Alg. 7. Our simulation is in an MPC framework, that is re-planning after each step. We simulate the number of trials. Each trial consists of a few alternating planning and execution sessions. We compare the cumulative over trials and number of planning sessions, running times of the planning sessions, and cumulative rewards along the simulated execution of the selected online policy, which is, in fact, the algorithm itself. Most importantly, we measure the quality of various safety formulations by the number of collisions that happened when the robot executed the selected optimal actions. Our action space is the space of motion primitives of unit vectors $\mathcal{A} = \{\rightarrow, \nearrow, \uparrow, \nwarrow, \leftarrow, \swarrow, \downarrow, \searrow, \text{Null}\}$. For simplicity, in the setting of policies in both problems, our belief-dependent reward is

$$\rho(b, a, b') = \tfrac{1}{m_x} \textstyle\sum_{i=1}^{m_x} r(x'^{,i}) \quad x'^{,i} \sim b', \tag{84}$$

where m_x is the number of the belief particles. However, as we further prove in Appendix D, we still can account for uncertainty even with the reward being the first moment of a state-dependent reward.

The second part is the static action sequences. Here we shall compare the chance-constrained formulation, Alg. 6, with Alg. 2 and the variant with safe trajectories as explained in Section 7.2.3. Interestingly, the reciprocal parameters in the PC and CC would be as follows. In Alg. 2 we shall select δ=1 and $1-\epsilon$ equal to δ in Alg. 6. This is because safe trajectory in Alg. 6 reciprocal to probability to be safe given posterior belief in Alg. 2 and probability of trajectories to be safe thresholded by δ reciprocal to the probability of a sequence of future beliefs to be safe thresholded by $1-\epsilon$.

Algorithm 8 Matched Chance-constrained BMDP Sparse Sampling (FastCCSS)

procedure FastCCSS(belief: $\bar{b}$ depth: d, threshold $\Delta(h)$) ▷ $\bar{b}$ as in (48)
  if $d = 0$ **then**
    return (Null, 0, $\mathrm{P}(\{x \notin \mathcal{X}^{\text{safe}}\}|\bar{b})$) ▷ $\mathrm{P}(\{x \in \mathcal{X}^{\text{safe}}\}|\bar{b}) = 1 - r_b(\bar{b}), \mathrm{P}(\{x \notin \mathcal{X}^{\text{safe}}\}|\bar{b}) = r_b(\bar{b})$
  end if
  $(a^*, v^*, \mathrm{er}_{k+L-d}(\bar{b}, \pi^*)) \leftarrow$ (Null, $-\infty, 1$) ▷ $\mathrm{er}_\ell(\bar{b}, \pi^*) \leq 1$
  $\bar{b}^{\text{safe}} \leftarrow$ Make $\bar{b}$ safe ▷ obtain safe belief as in equation (49) using (51) from belief defined by (48)
  for $a \in \mathcal{A}$ **do**
    $v \leftarrow 0.0$ ▷ Value function initialization
    Calculate propagated belief $\bar{b}'^{-}$ from $\bar{b}^{\text{safe}}$ applying a
    for $j \in 1 : m_d$ **do**
      Sample $x^{o,j} \sim \bar{b}'^{-}$ followed by $z'^{,j} \sim \mathbb{P}(z|x^{o,j})$ Observations are created using belief defined by (49) and action a.
      $\bar{b}'^{,j} \leftarrow \psi(\bar{b}^{\text{safe}}, a, z'^{,j})$
      $\Delta(haz'^{,j}) \leftarrow \Delta\text{PRIME}(\Delta(h), r_b(\bar{b}), 1/m_d), \delta(haz'^{,j}) \leftarrow 1 - \Delta(haz'^{,j})$ ▷ Call Alg. 4
      if $\mathrm{P}(\{x' \in \mathcal{X}^{\text{safe},\prime}\}|\bar{b}'^{,j}) < \delta(haz')$ **then**
        next action ▷ Prune action a using (60)
      end if
    end for
    if PRUNEACTIONCHANCE($\{\bar{b}'^{,j}\}_{j=1}^{m_d}, \{\frac{1}{m_d}\}_{j=1}^{m_d}, \bar{b}, \Delta(h)$) **then**
      next action ▷ See Alg. 5
    end if
    for $j \in 1 : m_d$ **do**
      $a'^{,*}, v', \mathrm{er}_{k+L-d+1}(\bar{b}'^{,j}, \pi^*) \leftarrow$ FastCCSS($\bar{b}'^{,j}, d-1, \Delta(haz'^{,j})$)
      $v+ = (\rho(\bar{b}, a, \bar{b}'^{,j}) + \gamma \cdot v')/m_d$ ▷ Reward is calculated over the beliefs defined by (49) and (48).
    end for
    $\mathrm{er}_{k+L-d}(\bar{b}, a, \pi^*) \leftarrow \mathrm{P}(\{x \notin \mathcal{X}^{\text{safe}}\}|\bar{b}) + \mathrm{P}(\{x \in \mathcal{X}^{\text{safe}}\}|\bar{b}) \cdot \frac{1}{m_d} \sum_{j=1}^{m_d} \mathrm{er}_{k+L-d+1}(\bar{b}'^{,j}, \pi^*)$
    if $\mathrm{er}_{k+L-d}(\bar{b}, a, \pi^*) > \Delta(h)$ and $v > v^*$ **then** ▷ CC check
      $(a^*, v^*, \mathrm{er}_{k+L-d}(\bar{b}, \pi^*)) \leftarrow (a, v, \mathrm{er}_{k+L-d}(\bar{b}, a, \pi^*))$
    end if
  end for
  return (a^*, v^*, $\mathrm{er}_{k+L-d}(\bar{b}, \pi^*)$)
end procedure

We simulate this setting on the first problem under consideration, navigation to the static goal. The space of candidate action sequences $\mathcal{A}_k$ is several diverse paths to the goal on top of the Probabilistic Road Map (PRM) [17], starting from the vertex closest to the expected value of prior belief. In the setting of static action sequences, we consider a general belief-dependent reward.

8.1 Studied Problems

In this section we describe the problems under consideration. In both problems we have a single obstacle.

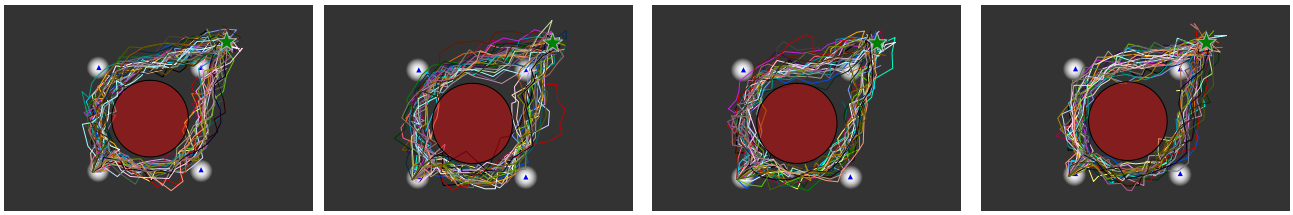
8.1.1 Navigation to static Goal

Following the previous discussion we are ready to delve into the details of our first problem. We adopt the well known problem of navigation to the goal with collision avoidance. The belief in this problem is maintained over the robot's position which is a 2-dimensional vector. In the setting of policies, our state dependent reward is $r(x') = -\|x' - x^g\|_2^2$. By x^g we denote the location of the goal. Hence, our theoretical reward is $\rho(b, a, b') = \mathbb{E}_{x' \sim b'}[r(x')]$. Note that such $\rho(b, a, b')$ accounts for belief uncertainty as we show in Appendix D. In the setting of static action sequences since all paths lead to the goal we select such a reward only for beliefs in the final time index $k+L$. For intermediate time instances we set the reward to be $\rho(b, a, b') = -\mathrm{Trace}(\Sigma(b'))$, where $\Sigma(\cdot)$ is a covariance matrix of the corresponding belief. Our obstacle has a circular shape with a center at x^o and radius r^o. We approximate the probability of not having the collision by

$$\mathbb{P}\left(\{x_\ell \in \mathcal{X}_\ell^{\text{safe}}\} \,|\Box\right) = 1 - \frac{1}{m_x} \sum_{i=1}^{m_x} \mathbf{1}_{\{\|x_\ell - x^o\|_2 \leq r^o\}}(x_\ell^i), \tag{85}$$

(a) (b)

Fig. 10: Visualization of a single trial from 50 of **navigation to the static goal problem** at our **first map**. The hyperparameters are $m_1 = 10, m_2 = 10, L = 2, m_x = 100$. **(a)** Constraints are deactivated ($\delta = 0$). The robot solves solely the objective (9), actual belief b_k is not made safe. We show 21 execution of optimal action found by Alg. 1. **(b)** Here we show one trial, namely 21 times robot executes the best action found by Alg. 1 with $\delta = 0.7$.



(a) (b) (c) (d)

Fig. 11: Visualization of the 50 trials actual (not planning)trajectories the robot solving **navigation to the static goal problem** at our **first map**.The hyperparameters are $m_1 = 10, m_2 = 10, L = 2, m_x = 100$, $\delta = 0.8$: **(a)** Alg. 1. **(b)** Alg. 3. **(c)** Alg. 8 **(d)** Alg. 7.

where $x_\ell^i \sim \square$ and $\square$ can be b_ℓ defined by (5) or $\bar{b}_\ell$ as in (48).

For the static action sequences we represent the map as a binary grid where one represents an obstacle and zero a free space. We verify the probability to be safe by checking the cell of each particle, summing the Boolean values of the cells and dividing by the overall number of the cells.

Motion and observation models, and the initial belief are $\mathbb{P}_{\mathrm{T}}(\cdot|x,a){=}\mathcal{N}(x{+}a,\Sigma_{\mathrm{T}})$, $\mathbb{P}_{\mathrm{O}}(\cdot|x;\{x^{b,i}\}_{i=1}){=}\mathcal{N}(x,\Sigma_{\mathrm{O}})$, $b_0{=}\mathcal{N}(x_0,\Sigma_0)$ respectively. The robot obtains an observation from the closest beacon. The covariance matrices are diagonal $\Sigma_T = I \cdot \sigma_w^2$ and for policies

$$\Sigma_O(x;\{x^{b,i}\}_{i=1}){=}\begin{cases}\sigma_w^2 I \min\limits_i d_i, & \text{if } \min\limits_i d_i \geq r_{\min}\\ \sigma_v^2 I, & \text{else}\end{cases} \tag{86}$$

where $d_i{=}\|x{-}x^{b,i}\|_2$, $x^{b,i}$ is the 2D location of the beacon i. We set the parameters to be $r_{\min}{=}0.01$, $\sigma_w^2{=}0.1$ and $\sigma_v^2{=}0.01$, $\gamma{=}0.99$.

For the static action sequences we set $\sigma_w^2{=}0.01$ and slightly change the covariance of the observation model and have just two areas at the binary grid map. One area has low observation noise and another high. The low observation noise is $\Sigma_O{=}0.0001I$ and the high observation noise is $\Sigma_O = 0.1I$.

The initial belief admits a Gaussian distribution $\mathcal{N}(x_k;\mu,\Sigma)$. For policies we selected the covariance $\Sigma = \sigma I = 0.1 \cdot I$ and mean $\mu = (0.0, 0.0)^T$. The initial ground truth state of the robot was set to $x_k^{\mathrm{gt}} = (-0.5, -0.2)^T$. For the static action sequences, we selected $\mu = x_k^{\mathrm{gt}} = (5.0, 0.0)^T$ and $\Sigma = \sigma I = 0.01 \cdot I$.

8.1.2 Target Tracking

Now we describe the second problem. We simulate this problem only in the setting of policies. In this problem we have a moving target in addition to the agent. In this problem the belief is maintained over both positions, the agent and the target. The state dependent reward in this problem is $r(x'){=}-\|x'^{,\mathrm{agent}}{-}x'^{,\mathrm{target}}\|_2^2$. It accounts for the uncertainty of both the target and the agent in a similar manner as in the previous problem. Moreover, now we have a squared obstacle. We check collision now according to

$$\mathbb{P}(\{x_\ell^{\mathrm{agent}} \in \mathcal{X}_\ell^{\mathrm{safe}}\}\,|\square) = 1 - \tfrac{1}{m_x}\textstyle\sum_{i=1}^{m_x} \mathbf{1}_{\{\|x_\ell^{\mathrm{agent}} - x^o\|_\infty \leq r^o\}}(x_\ell^{\mathrm{agent},i}), \tag{87}$$

Table 1: 50 Trials of 21 planning sessions and executions of optimal action of four Algorithms 1, 3, 8, 7.Same seed in four algorithms. This problem is the **navigation to static goal** described in Section 8.1.1 in **our first map** $L = 2$. Here we study the number of collisions and the reward value.

Parameters				num collisions				mean cum. rew. ± std				mean cum. rew. no coll ± std			
m_x	m_1	m_2	δ	PCSS 1	CCSS-IS 3	MatchedCCSS 8,	PCSSST 7	PCSS 1	CCSS-IS 3	MatchedCCSS 8,	PCSSST 7	PCSS 1	CCSS-IS 3	MatchedCCSS 8,	PCSSST 7
100	10	10	0.9	3/50	3/50	5/50	2/50	-155.93 ± 18.92	-160.05 ± 21.77	-158.20 ± 16.47	-158.49 ± 19.73	-155.87 ± 18.48	-160.26 ± 22.40	-158.18 ± 16.60	-158.16 ± 20.0
100	10	10	0.8	5/50	6/50	4/50	4/50	-155.00 ± 14.88	-156.66 ± 17.01	-154.60 ± 17.60	-151.72 ± 15.12	-154.26 ± 15.034	-155.56 ± 17.13	-154.87 ± 16.91	-150.53 ± 14.20
100	10	10	0.7	6/50	11/50	13/50	11/50	-147.33 ± 16.03	-151.83 ± 20.40	-145.11 ± 13.20	-150.87 ± 19.22	-147.48 ± 16.20	-153.74 ± 21.41	-143.85 ± 11.56	-152.70 ± 19.80
100	10	10	0.6	19/50	20/50	13/50	14/50	-149.84 ± 16.29	-149.78 ± 18.02	-147.47 ± 18.29	-144.78 ± 14.30	-151.94 ± 14.80	-149.32 ± 17.04	-150.71 ± 19.22	-143.43 ± 15.37
100	10	10	0.0	50/50	50/50	50/50	50/50	-106.14 ± 8.65	-107.94 ± 9.86	-109.88 ± 10.68	-106.14 ± 8.65				

Table 2: 50 Trials of 21 planning sessions and executions of optimal action of four Algorithms 1, 3, 8, 7. Same seed in four algorithms. This problem is the **navigation to static goal** described in Section 8.1.1 in our **first map** $L = 2$. In this table we study speedup (91).

Parameters				cum. plan. time [sec]				speedup Alg. 1 rel to 3	speedup Alg. 8 rel to 3	speedup Alg. 7 rel to 3
m_x	m_1	m_2	δ	PCSS 1	CCSS-IS 3	MatchedCCSS 8	PCSSST 7			
100	10	10	0.9	1162.41	2938.63	1095.23	1149.76	**0.60**	**0.63**	**0.61**
100	10	10	0.8	1167.33	2976.67	1117.29	1179.46	**0.61**	**0.62**	**0.60**
100	10	10	0.7	1191.02	2964.89	1157.73	1182.83	**0.60**	**0.61**	**0.60**
100	10	10	0.6	1186.62	3043.97	1162.80	1203.82	**0.61**	**0.62**	**0.60**
100	10	10	0.0	1527.71	4226.43	1523.82	1547.63	**0.64**	**0.64**	**0.63**

where $x_\ell^{\text{agent},i} \sim \square$ and $\square$ can be b_ℓ defined by (5) or $\bar{b}_\ell$ as in (48). Here, the $\|\xi\|_\infty = \max_i |\xi_i|$, where ξ_i is the coordinate i of vector ξ_i. The motion model of the target is identical to the motion model of the agent and follows

$$\mathbb{P}_{\text{T}}(\cdot|x, a) = \mathcal{N}(x^{\text{agent}} + a^{\text{agent}}, \Sigma_T) \cdot \mathcal{N}(x^{\text{target}} + a^{\text{target}}, \Sigma_T), \tag{88}$$

where by x we denote the concatenated $\{x^{\text{agent}}, x^{\text{target}}\}$. For the target action we use a circular buffer with $\{\leftarrow, \uparrow\}$ action sequence. We maintain a belief over the agent and the target. For simplicity, similar to [15], we assume that in inference as well as in planning session we know the target action sequence. The observation model is also the multiplication of the observation model from the previous section with the additional observation model due to a moving target. Therefore, the overall observation model is

$$\mathbb{P}_{\text{O}}(\cdot|x; \{x^{b,i}\}_{i=1}) = \mathcal{N}(x^{\text{agent}}, \Sigma_{\text{O}}(x^{\text{agent}}; \{x^{b,i}\}_{i=1})) \cdot \mathcal{N}(x^{\text{agent}} - x^{\text{target}}, \Sigma_{\text{O}}(x^{\text{agent}}, x^{\text{target}})), \tag{89}$$

where $\Sigma_{\text{O}}(x^{\text{agent}}; \{x^{b,i}\}_{i=1})$ conforms to (86) and

$$\Sigma_{\text{O}}(x^{\text{agent}}, x^{\text{target}}) = \begin{cases} \sigma_w^2 I \|x^{\text{agent}} - x^{\text{target}}\|_2, \text{ if } \|x^{\text{agent}} - x^{\text{target}}\|_2 \geq r_{\min} \\ \sigma_v^2 I, \text{else} \end{cases} \tag{90}$$

Importantly, the target does not collide with obstacles, it can **fly above**. In this problem we selected the parameters to be $r_{\min}$=0.01, σ_w^2=0.1 and σ_v^2=0.01, γ=0.99. The initial belief admits Gaussian distribution $\mathcal{N}(x_k; \mu, \Sigma)$ with covariance $\Sigma = \sigma I = 0.01 \cdot I$ and mean $\mu = (\underbrace{0, 0}_{\text{agent}}, \underbrace{10, 0}_{\text{target}})^T$. Initial ground truth state of the robot and the target was set to $x_k^{\text{gt}} = (\underbrace{-0.5, -0.2}_{\text{agent}}, \underbrace{10, 0}_{\text{target}})^T$.

8.2 Measures of Acceleration

For each pair of algorithms we calculate the speedup according to

$$\frac{t^{\text{baseline}} - t^{\text{algorithm}}}{t^{\text{baseline}}}. \tag{91}$$

Eq. (91) measures saved time relative to the baseline running time. In a similar manner the relative fraction of number of expanded and not pruned actions N is

$$\frac{N^{\text{baseline}} - N^{\text{algorithm}}}{N^{\text{baseline}}}. \tag{92}$$

Note also that it is possible that the algorithms declare that no feasible solution exists or the actual belief cannot be made safe, if all samples fall inside the obstacle.

8.3 Policies

In this section, we study our both problems under consideration in the context of policies. In both problems, each studied configuration of parameters, and in each trial, we also ran the Alg. 3 and Alg. 8 enforcing CC solely from the root by using (81). Each such run returned that no feasible solution exists. In addition, in both problems when δ=0, the agent crashed in all 50 trials.

Table 3: 50 Trials of 21 planning sessions and executions of optimal action of four Algorithms 1, 3, 8, 7. Same seed in four algorithms. This problem is the **navigation to static goal** described in Section 8.1.1 in our **first map** $L = 2$. In this table we study the saved actions fraction (92).

Parameters				Total expanded actions				actions frac. Alg. 3 rel to 1	actions frac. Alg. 8 rel to 7
m_x	m_1	m_2	δ	PCSS 1	CCSS-IS 3	MatchedCCSS 8	PCSSST 7		
100	10	10	0.9	609913	596070	580005	606365	0.023	0.043
100	10	10	0.8	616117	597623	594274	622640	0.03	0.046
100	10	10	0.7	632289	602769	618113	625493	0.05	0.011
100	10	10	0.6	631216	617473	622583	639718	0.022	0.026

Table 4: 50 Trials of 21 planning sessions and executions of optimal action of Algorithm 3. This problem is the **navigation to static goal** described in Section 8.1.1 in **our first map** $L = 2$. Here we study the number of collisions and the reward value as function of future thresholds in CC.

Parameters					num collisions	mean cum. rew. $\pm$ std	mean cum. rew. no coll $\pm$ std	Cum. plan time [sec]	Cum. expanded actions
m_x	m_1	m_2	$\Delta(h_k)$	$\Delta(h_\ell)$					
100	10	10	0.1	(79)	4/50	-159.02 ± 16.02	-157.91 $\pm$ 15.97	4447.47	583606
				(80)	3/50	-160.05 ± 21.77	-160.26 $\pm$ 22.40	4506.64	596070
100	10	10	0.2	(79)	7/50	-151.43 ± 20.26	-152.33 ± 21.55	4568.71	606670
				(80)	6/50	-156.66 ± 17.01	-155.56 ± 17.13	4495.29	597623

8.3.1 Navigation to static Goal

We present results for the first problem (see Section 8.1.1) in Tables 1, 2 and 3. We visualize a single trial of 21 alternating planning and execution sessions in Fig. 10. In Fig. 11 we show actual robot trajectories from 50 trials. As we see from Table 1 barring some noise due to sample approximations the number of collisions consistently increases with decreasing δ. Moreover, as explained in Section 5.6, the CC in Alg. 3 pictured by (61) is supposed to be more conservative than the PC in Alg. 1. Indeed, we see in Table 1, the trend is that the cumulative reward of Alg. 3 is slightly smaller than the one of Alg. 1. Due to the fact that this relation is not preserved in Alg. 8 versus Alg. 7 we conclude that roughly the reason is that in CC only the safe trajectories are pushed forward in time with action and observation, resulting in beliefs as in (48), (49) and not as in (5). From Table 2 we elicit that avoiding IS in Alg. 3 yields speed up of approximately 60%. In addition we behold in Table 3 that when we use pruning suggested by [30] and explained in Section 6.2 it prunes more actions than our pruning described in Section 4.1.3. In Table 4 we report results of running Alg. 3 with two heuristics for future Δ.

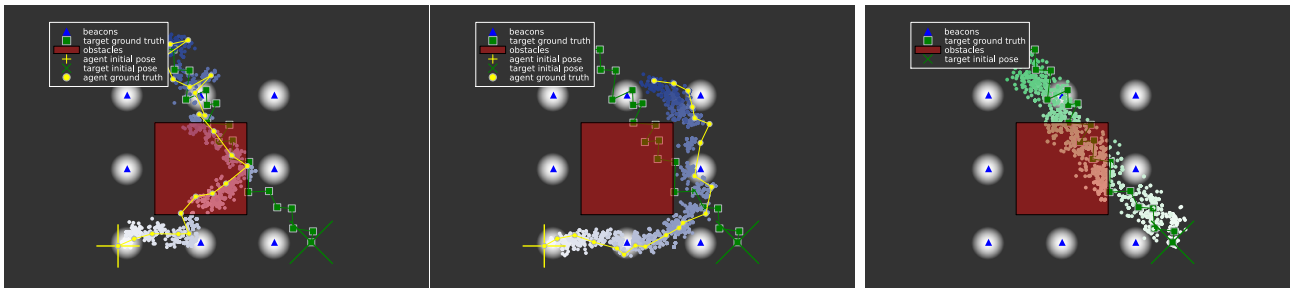
We proceed now to the same experiments with our second problem.

8.3.2 Target Tracking

In this section, we study the Target Tracking problem defined in Section 8.1.2. Similarly to the previous section, we show an example of the problem in Fig. 12. In Fig. 13 we show actual and not planning 50 robot trajectories. These 50 trajectories the robot has obtained by performing 50 trials of alternating 21 planning and execution sessions. We present the number of collisions and cumulative rewards of 50 trajectories of the robot in Table 5. Interestingly, here, we mark that the expected cumulative reward without collisions is not always smaller than the mean cumulative reward, including collided trajectories. We explain that by the fact that in this problem, the reward also depends on the target particles, so the behavior is more affected by sample noise. From Table 6 we again heed a stable speedup of approximately 60% relative to Alg. 3. Surprisingly, we see in Table 7 that not always Alg. 3 expands fewer actions than Alg. 1. This can be due to sample noise, since even with the same seed in both algorithms, the belief in line 22 in what Alg. 3 is updated after all the observations were sampled, as opposed to Alg. 1. This is also corroborated by the fact that, as in the previous problem, Alg. 8 constantly expands less actions than Alg. 7.

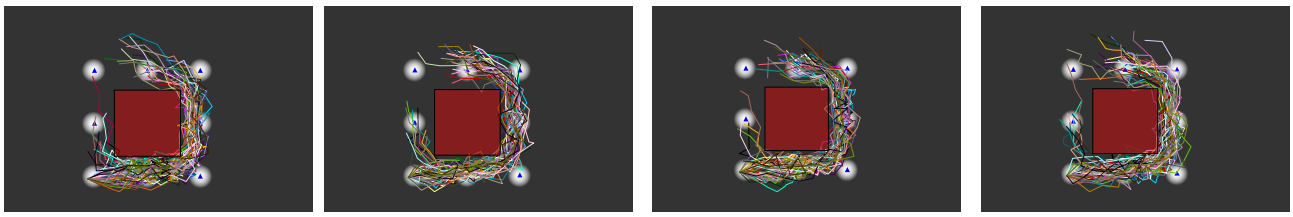
8.4 Static Action Sequences

From the previous section we gain that the number of collisions as a function of δ is roughly the same for all algorithms. This can be because, in the setting of policies, we do replanning after each session, also known as Model Predictive Control (MPC). In static action sequences setting, we do not do replanning. The robot just follows the identified optimal candidate action sequence. In this setting, to obtain $\mathcal{A}_k$, we simulate the problem of distance to goal on top of diverse paths found by the Probabilistic Road Map (PRM). See Fig. 14. With such an approach, we can increase the horizon significantly compared to the policies in the previous section. We obtained a horizon of 10 or 11, depending on the path. We show 50 Robot trajectories in Fig. 15. The green area in Fig. 15 is the low measurement noise area as explained in Section 8.1.1. The gray square is the obstacle. In a modified version of Alg. 2, we make belief safe if possible; if not, we



(a) (b) (c)

Fig. 12: Visualization of a single trial from 50 of **target tracking problem** at our **second map**. The hyperparameters are $m_1 = 10, m_2 = 10, L = 2, m_x = 100$ **(a)** Constraints are deactivated. The robot solves solely the objective (9), actual belief b_k is not made safe. Here we show only the particles of the agent. The agent executed 21 alternating planning and execution sessions; **(b)** Here we show 21 executions of the best action found by Alg. 1, agent particles. **(c)** 21 executions of the best action found by Alg. 1, target particles.



(a) (b) (c) (d)

Fig. 13: Visualization of the 50 trials actual (not planning) trajectories the robot solving **target tracking problem** at our **second map**.The hyperparameters are $m_1 = 10, m_2 = 10, L = 2, m_x = 100, \delta = 0.8$: **(a)** Alg. 1. **(b)** Alg. 3. **(c)** Alg. 8 **(d)** Alg. 7.

leave the belief as is. We set in Alg. 6 δ=0.6 and compared with two variants of Alg. 2 with δ=1 and ϵ=0.4. Table 8 shows a similar number of collisions by three planning algorithms. However, Alg. 6 appears to be faster.

9 Conclusions

We proposed a novel formulation of belief-dependent Probabilistically Constrained continuous POMDP. Our formulation allows us, adaptively with respect to observation laces, to accept or reject the candidate policy/action sequence satisfying or violating the Probabilistic Constraint. We also uplifted chance-constrained POMDP to continuous domains in terms of states and observations and general belief-dependent rewards. Our simulations corroborate the superiority of our efficient algorithms in terms of celerity. In all simulations with policies we obtained a typical speedup of 60% of PC versus CC. In the settings of static action sequences our uplifted CC approach appears to be faster than PC. We intend to continue investigating the proposed formulation towards larger horizons using anytime online approaches.

Appendix A Proofs

A.1 Proof of Lemma 1 (Representation of Our Outer Constraint).

To verify the inner constraint $c(b_{k:k+L}; \phi, \delta)$ we need to know lace of the beliefs, operator ϕ and δ. To rephrase that

$$\mathrm{P}\left(c(b_{k:k+L}; \phi, \delta) = 1 | b_k, \pi, a_k, b_{k+1:k+L}\right) = c(b_{k:k+L}; \phi, \delta). \tag{93}$$

In addition let us state the fact that $\mathbb{P}(b_{k+1:k+L}|b_k, \pi, a_k, z_{k+1:k+L})$ is Dirac's delta function. All in all, we can write

$$\begin{aligned}
&\mathrm{P}\left(c(b_{k:k+L}; \phi, \delta) = 1 | b_k, \pi, a_k\right) = \textstyle\int_{b_{k+1:k+L}} \mathrm{P}\left(c(b_{k:k+L}; \phi, \delta) = 1 | b_k, \pi, a_k, b_{k+1:k+L}\right) \cdot \\
&\Big(\int_{z_{k+1:k+L}} \mathbb{P}(b_{k+1:k+L}, z_{k+1:k+L} | b_k, \pi, a_k) \mathrm{d}z_{k+1:k+L}\Big) \mathrm{d}b_{k+1:k+L} = \int_{\substack{b_{k+1:k+L} \\ z_{k+1:k+L}}} \mathrm{P}\left(c(b_{k:k+L}; \phi, \delta) = 1 | b_k, \pi, a_k, b_{k+1:k+L}\right) \cdot \\
&\mathbb{P}(b_{k+1:k+L} | b_k, \pi, a_k, z_{k+1:k+L}) \mathrm{d}b_{k+1:k+L} \mathbb{P}(z_{k+1:k+L} | b_k, \pi, a_k) \mathrm{d}z_{k+1:k+L} = \mathbb{E}_{z_{k+1:k+L}}[c(b_{k:k+L}; \phi, \delta) | b_k, \pi, a_k].
\end{aligned} \tag{94}$$

Table 5: 50 Trials of 21 planning sessions and executions of optimal action of four Algorithms 1, 3, 8, 7.Same seed in four algorithms. This problem is the **target tracking** described in Section 8.1.2 in **our second map** L=2. Here we study the number of collisions and the reward value.

Parameters				num collisions				mean cum. rew. ± std				mean cum. rew. no coll ± std			
m_x	m_1	m_2	δ	PCSS 1	CCSS-IS 3	MatchedCCSS 8,	PCSSST 7	PCSS 1	CCSS-IS 3	MatchedCCSS 8,	PCSSST 7	PCSS 1	CCSS-IS 3	MatchedCCSS 8	PCSSST 7
100	10	10	0.9	7/50	8/50	11/50	10/50	-112.10 ± 19.80	-105.76 ± 21.89	-103.68 ± 19.32	-103.49 ± 20.31	-111.44 ± 19.77	-107.07 ± 21.75	-108.29 ± 18.59	-101.11 ± 18.12
100	10	10	0.8	17/50	12/50	11/50	20/50	-108.07 ± 19.91	-101.74 ± 18.74	-101.87 ± 16.95	-99.83 ± 18.55	-109.47 ± 20.66	-102.74 ± 19.71	-104.26 ± 17.08	-105.63 ± 15.25
100	10	10	0.7	22/50	23/50	24/50	24/50	-106.61 ± 22.52	-98.69 ± 19.86	-97.63 ± 20.43	-102.93 ± 18.25	-107.16 ± 26.05	-100.33 ± 21.27	-103.92 ± 19.31	-102.72 ± 18.40
100	10	10	0.0	50/50	50/50	50/50	50/50	-54.15 ± 7.81	-53.39 ± 7.81	-109.88 ± 10.68	-52.88 ± 7.15				

Table 6: 50 Trials of 21 planning sessions and executions of optimal action of four Algorithms 1, 3, 8, 7. Same seed in four algorithms. This problem is the **target tracking** described in Section 8.1.2 in our **second map** $L = 2$. In this table we study speedup (91).

Parameters				cum. plan. time [sec]				speedup Alg. 1 rel to 3	speedup Alg. 8 rel to 3	speedup Alg. 7 rel to 3
m_x	m_1	m_2	δ	PCSS 1	CCSS-IS 3	MatchedCCSS 8	PCSSST 7			
100	10	10	0.9	1635.96	4244.44	1630.13	1689.77	**0.61**	**0.62**	**0.60**
100	10	10	0.8	1662.99	4476.03	1668.59	1741.12	**0.63**	**0.63**	**0.61**
100	10	10	0.7	1691.39	4472.66	1666.97	1731.69	**0.62**	**0.63**	**0.61**
100	10	10	0.0	2530.47	7221.41	2443.94	2516.20	**0.65**	**0.66**	**0.65**

If in addition in case of (12), we have that

$$\mathbb{E}_{z_{k+1:k+L}}[c(b_{k:k+L};\phi,\delta)|b_k,\pi,a_k] = \int_{z_{k+1:k+L}} \mathbb{P}(z_{k+1:k+L}|b_k,\pi_{(k+1)+},a_k)\big(\prod_{\ell=k}^{k+L}\mathbf{1}_{\{\phi(b_\ell)\geq\delta\}}(b_\ell)\big)\mathrm{d}z_{k+1:k+L} = \tag{95}$$

$$\int_{z_{k+1:k+L}} \mathbb{P}(z_{k+1:k+L-1}|b_k,\pi_{(k+1)+},a_k)\mathbb{P}(z_{k+L}|b_k,\pi_{(k+1)+},a_k,z_{k+1:k+L-1})\big(\prod_{\ell=k}^{k+L}\mathbf{1}_{\{\phi(b_\ell)\geq\delta\}}(b_\ell)\big)\mathrm{d}z_{k+1:k+L} = \tag{96}$$

$$\mathbf{1}_{\{\phi(b_k)\geq\delta\}}(b_k)\mathbb{E}_{z_{k+1}}\big[\mathbf{1}_{\{\phi(b_{k+1})\geq\delta\}}(b_{k+1})\mathbb{E}_{z_{k+2}}\big[\mathbf{1}_{\{\phi(b_{k+2})\geq\delta\}}(b_{k+2})\dots \tag{97}$$

$$\dots\mathbb{E}_{z_{k+L-1}}\big[\mathbf{1}_{\{\phi(b_{k+L-1})\geq\delta\}}(b_{k+L-1})\mathbb{E}_{z_{k+L}}\big[\mathbf{1}_{\{\phi(b_{k+L})\geq\delta\}}\big|b_{k+L-1},\pi\big]\big|b_{k+L-2},\pi\big]\dots\big|b_{k+1},\pi\big]\big|b_k,\pi\big].$$

■

A.2 Proof of Theorem 1 (Necessary and sufficient condition for feasibility of PC)

Before we start let us state that by definition, using $c(b^l_{k:k+L};\phi,\delta) = \prod_{\ell=k}^{k+L}\mathbf{1}_{\{\phi(b_\ell)\geq\delta\}}(b^l_\ell)$ holds

$$1 \geq \tfrac{1}{m}\sum_{l=1}^m c(b^l_{k:k+L};\phi,\delta) = \tfrac{1}{m}\sum_{l=1}^m\big(\prod_{\ell=k}^{k+L}\mathbf{1}_{\{\phi(b_\ell)\geq\delta\}}(b^l_\ell)\big) \tag{98}$$

Suppose that $1 \geq \frac{1}{m}\sum_{l=1}^m c(b^l_{k:k+L};\phi,\delta) \geq 1$, so

$$\sum_{l=1}^m\big(\prod_{\ell=k}^{k+L}\mathbf{1}_{\{\phi(b_\ell)\geq\delta\}}(b^i_\ell)\big) = m \tag{99}$$

Suppose in contradiction that $\exists l,\ell$ such that $\mathbf{1}_{\{\phi(b_\ell)\geq\delta\}}(b^l_\ell) = 0$. We have that

$$\sum_{i=1}^m\big(\prod_{\ell=k}^{k+L}\mathbf{1}_{\{\phi(b_\ell)\geq\delta\}}(b^l_\ell)\big) < m \tag{100}$$

This proves the first statement. For the second statement we prove the reciprocal implication. Assume that $\forall l,\ell$ holds $\mathbf{1}_{\{\phi(b_\ell)\geq\delta\}}(b^l_\ell) = 1$, we arrived at the fulfilling equation (99). ■

A.3 Proof of Lemma 2 (PDF of the trajectory)

$$\mathbb{P}(x_{k:k+L}|b_k,\pi_{k:k+L-1}) = \int_{z_{k+1:k+L-1}} \mathbb{P}(x_{k:k+L},z_{k+1:k+L-1}|b_k,\pi)\mathrm{d}z_{k+1:k+L-1} = \tag{101}$$

$$\int_{z_{k+1:k+L-1}} \mathbb{P}(x_{k+L}|z_{k:k+L-1},x_{k:k+L-1},b_k,\pi)\mathbb{P}(z_{k:k+L-1},x_{k:k+L-1}|b_k,\pi)\mathrm{d}z_{k+1:k+L-1} = \tag{102}$$

$$\int_{z_{k+1:k+L-1}} \mathbb{P}_{\mathrm{T}}(x_{k+L}|x_{k+L-1},a_{k+L-1})\mathbb{P}(z_{k+L-1}|x_{k:k+L-1},z_{k+1:k+L-2},b_k,\pi)\mathbb{P}(x_{k:k+L-1},z_{\ell+1:k+L-2}|b_k,\pi)\mathrm{d}z_{k+1:k+L-1} = \tag{103}$$

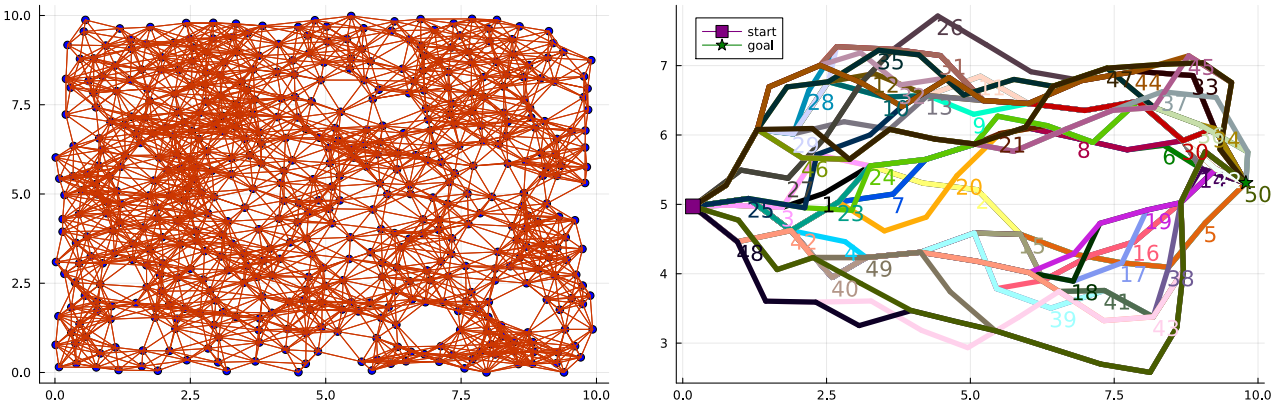
$$\int_{z_{k+1:k+L-1}} \mathbb{P}_{\mathrm{T}}(x_{k+L}|x_{k+L-1},a_{k+L-1})\mathbb{P}_{\mathrm{O}}(z_{k+L-1}|x_{k+L-1})\mathbb{P}(x_{k:k+L-1},z_{k+1:k+L-2}|b_k,\pi)\mathrm{d}z_{k+1:k+L-1}. \tag{104}$$

We behold the recurrence relation. Overall we have that

$$\mathbb{P}(\tau_k|b_k,\pi_{k:k+L-1}) = \mathbb{P}_{\mathrm{T}}(x_{k+1}|x_k,a_k)b_k(x_k)\int_{z_{k+1:k+L-1}}\prod_{k=\ell+1}^{k+L-1}\big(\mathbb{P}_T(x_{\ell+1}|x_\ell,\pi(b_\ell(b_{\ell-1},a_{\ell-1},z_\ell)))\mathbb{P}_{\mathrm{O}}(z_\ell|x_\ell)\big)\mathrm{d}z_{k+1:k+L-1}. \tag{105}$$

Table 7: 50 Trials of 21 planning sessions and executions of optimal action of Algorithms 1, 3, 8, 7. Same seed in all four algorithms. This problem is the **target tracking** described in Section 8.1.2 in our **second map** L=2. In this table we study the saved actions fraction (92).

Parameters				Total expanded actions				actions frac. Alg. 3 rel to 1	actions frac. Alg. 8 rel to 7
m_x	m_1	m_2	δ	PCSS 1	CCSS-IS 3	MatchedCCSS 8	PCSSST 7		
100	10	10	0.9	512035	509027	514234	529120	0.0059	0.028
100	10	10	0.8	519206	540897	530558	546304	−0.042	0.029
100	10	10	0.7	530455	540110	526456	548712	−0.018	0.041



(a) PRM **(b)** Obtained diverse paths

Fig. 14: Separate, algorithmically selected paths to the goal **(b)** on top of PRM **(a)**. We show the path number on the vertex, which is removed for finding the subsequent diverse path. The last's path number is shown at its final vertex (the goal). Paths start from the vertex closest to the mean value of the belief b_k.

In case we are given a static action sequence

$$\mathbb{P}(\tau_k|b_k, a_{k:k+L-1}) = \mathbb{P}_{\mathrm{T}}(x_{k+1}|x_k, a_k)b_k(x_k)\prod_{\ell=k+1}^{k+L-1}\mathbb{P}_{\mathrm{T}}(x_{\ell+1}|x_\ell, a_\ell)\int_{z_{k+1:k+L-1}}\prod_{\ell=k+1}^{k+L-1}\mathbb{P}_{\mathrm{O}}(z_\ell|x_\ell)\mathrm{d}z_{k+1:k+L-1} \tag{106}$$

This completes the proof. ■

A.4 Proof of Lemma 3 (Average over the Safe Posteriors)

$$\underbrace{\mathrm{P}\big(\bigcap_{\ell=k}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\right\}|b_k,\pi\big)}_{(a)} = \mathrm{P}\big(\left\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\right\}|b_k\big)\underbrace{\mathrm{P}\big(\bigcap_{\ell=k+1}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\right\}\big|\left\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\right\},b_k,\pi\big)}_{(b)} \tag{107}$$

Let us focus on the expression we marked by (b). The $\mathrm{P}\big(\bigcap_{\ell=k+1}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\right\}\big|\left\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\right\}b_k,\pi\big)$ equals to

$$\int_{\bar{b}_{k+1}}\mathrm{P}\big(\bigcap_{\ell=k+1}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\right\}|\bar{b}_{k+1},\left\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\right\},b_k,\pi\big)\mathbb{P}\big(\bar{b}_{k+1}|\left\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\right\}b_k,\pi\big)\mathrm{d}\bar{b}_{k+1} = \tag{108}$$

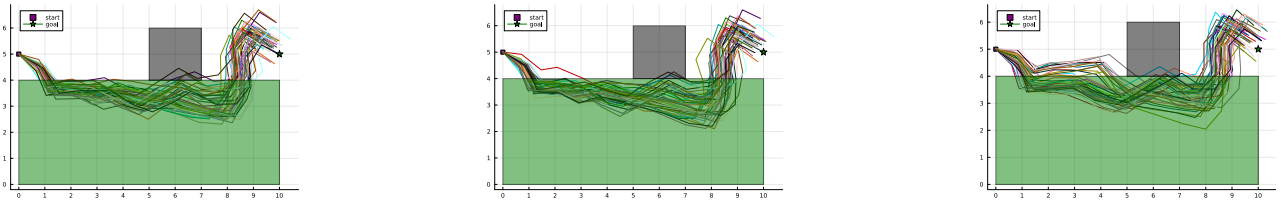
$$\int_{\bar{b}_{k+1}}\mathbb{P}\big(\bar{b}_{k+1}|\left\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\right\},b_k,\pi\big)\mathrm{P}\big(\bigcap_{\ell=k+1}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\right\}|\bar{b}_{k+1},\pi\big)\mathrm{d}\bar{b}_{k+1} \tag{109}$$

Merging the two expressions we obtain that $\mathrm{P}\big(\bigcap_{\ell=k}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\right\}|b_k,\pi\big)$ equals to

$$\mathrm{P}\big(\left\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\right\}|b_k\big)\int_{\bar{b}_{k+1}}\mathbb{P}\big(\bar{b}_{k+1}\big|\left\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\right\},b_k,\pi\big)\underbrace{\mathrm{P}\big(\bigcap_{\ell=k+1}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\right\}|\bar{b}_{k+1},\pi\big)}_{(c)}\mathrm{d}\bar{b}_{k+1} \tag{110}$$

Table 8: Number of collisions in openloop setting. Same seed. Modified Alg. 2 is as explained in Section 7.2.3.

	Alg. 2	Alg. 2 mod.	Alg. 6
num cols	4	9	5
plan time [sec]	48.23 ± 0.61	55.09 ± 0.61	32.27 ± 0.99



(a) **(b)** **(c)**

Fig. 15: Visualization of the 50 trials actual (not planning) trajectories the robot solving **navigation to static goal** at our **third map**. The gray area is the obstacle and the green area is the low measurement noise area; **(a)** Alg. 2. **(b)** Alg. 2 with modifications from section 7.2.3. **(c)** Alg. 6.

We observe that expression (a) is very similar to (c), namely

$$\begin{aligned}&\mathrm{P}\big(\textstyle\bigcap_{\ell=k+1}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\right\}|\bar{b}_{k+1},\pi\big)=\mathrm{P}\big(\left\{x_{k+1}\in\mathcal{X}_{k+1}^{\text{safe}}\right\}|\bar{b}_{k+1}\big)\cdot\\&\textstyle\int_{\bar{b}_{k+2}}\mathbb{P}(\bar{b}_{k+2}|\left\{x_{k+1}\in\mathcal{X}_{k+1}^{\text{safe}}\right\},\bar{b}_{k+1},\pi)\underbrace{\mathrm{P}\big(\bigcap_{\ell=k+2}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\right\}|\bar{b}_{k+2},\pi\big)}_{(d)}\mathrm{d}\bar{b}_{k+2}\end{aligned}\tag{111}$$

Merging the two we got

$$\begin{aligned}&\mathrm{P}\big(\textstyle\bigcap_{\ell=k}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\right\}|b_k,\pi\big)=\mathrm{P}\big(\left\{x_k\in\mathcal{X}_k^{\text{safe}}\right\}|b_k\big)\textstyle\int_{\bar{b}_{k+1}}\mathbb{P}(\bar{b}_{k+1}|\left\{x_k\in\mathcal{X}_k^{\text{safe}}\right\},b_k,\pi)\mathrm{P}\big(\left\{x_{k+1}\in\mathcal{X}_{k+1}^{\text{safe}}\right\}|\bar{b}_{k+1}\big)\cdot\\&\textstyle\int_{\bar{b}_{k+2}}\mathbb{P}(\bar{b}_{k+2}|\left\{x_{k+1}\in\mathcal{X}_{k+1}^{\text{safe}}\right\},\bar{b}_{k+1},\pi)\mathrm{P}\big(\bigcap_{\ell=k+2}^{k+L}\left\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\right\}|\bar{b}_{k+2},\pi\big)\mathrm{d}\bar{b}_{k+2}\mathrm{d}\bar{b}_{k+1}.\end{aligned}\tag{112}$$

We observe the recurrence relation.

Now we show that marginalization can be done with respect to the observations. Moreover we will see that the beliefs are different from the belief tree build for the rewards. Let us assume that ℓ is the last index ($\ell=k+L$)

$$\begin{aligned}&\textstyle\int_{\bar{b}_{k+L}}\mathrm{P}\big(\{x_{k+L}\in\mathcal{X}_{k+L}^{\text{safe}}\}|\bar{b}_{k+L}\big)\mathbb{P}\big(\bar{b}_{k+L}|\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\},\bar{b}_{k+L-1},\pi\big)\mathrm{d}\bar{b}_{k+L}=\\&\textstyle\int_{\bar{b}_{k+L}}\mathrm{P}\big(\{x_{k+L}\in\mathcal{X}_{k+L}^{\text{safe}}\}|\bar{b}_{k+L}\big)\int_{z_{k+L}}\mathbb{P}\big(\bar{b}_{k+L}|\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\},\bar{b}_{k+L-1},\pi,z_{k+L}\big)\end{aligned}\tag{113}$$

$$\mathbb{P}(z_{k+L}|\bar{b}_{k+L-1},\pi,\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\})\mathrm{d}z_{k+L}\mathrm{d}\bar{b}_{k+L}=$$

$$\textstyle\int_{\bar{b}_{k+L}}\int_{z_{k+L}}\mathrm{P}\big(\{x_{k+L}\in\mathcal{X}_{k+L}^{\text{safe}}\}|\bar{b}_{k+L}\big)\delta\big(\bar{b}_{k+L}-\psi(\bar{b}_{k+L-1},\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\},a_{k+L-1},z_{k+L})\big)\tag{114}$$

$$\begin{aligned}&\mathbb{P}\big(z_{k+L}|a_{k+L-1},\bar{b}_{k+L-1},\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\}\big)\mathrm{d}z_{k+L}\mathrm{d}\bar{b}_{k+L}=\\&\textstyle\int_{z_{k+L}}\int_{\bar{b}_{k+L}}\mathrm{P}(\{x_{k+L}\in\mathcal{X}_{k+L}^{\text{safe}}\}|\bar{b}_{k+L})\delta\big(\bar{b}_{k+L}-\psi(\bar{b}_{k+L-1},\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\},a_{k+L-1},z_{k+L})\big)\mathrm{d}\bar{b}_{k+L}\end{aligned}\tag{115}$$

$$\begin{aligned}&\mathbb{P}\big(z_{k+L}|a_{k+L-1},\bar{b}_{k+L-1},\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\}\big)\mathrm{d}z_{k+L}=\\&\textstyle\int_{z_{k+L}}\mathrm{P}\big(\{x_{k+L}\in\mathcal{X}_{k+L}^{\text{safe}}\}|\psi(\bar{b}_{k+L-1},\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\},a_{k+L-1},z_{k+L})\big)\end{aligned}\tag{116}$$

$$\begin{aligned}&\mathbb{P}\big(z_{k+L}|a_{k+L-1},\bar{b}_{k+L-1},\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\}\big)\mathrm{d}z_{k+L}=\\&\textstyle\mathbb{E}_{z_{k+L}}\big[\mathrm{P}\big(\{x_{k+L}\in\mathcal{X}_{k+L}^{\text{safe}}\}|\psi(\bar{b}_{k+L-1},\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\},a_{k+L-1},z_{k+L})\big)\big|a_{k+L-1},\bar{b}_{k+L-1},\{x_{k+L-1}\in\mathcal{X}_{k+L-1}^{\text{safe}}\}\big]\end{aligned}\tag{117}$$

We plug this result into expression for $k+L-1$ and do the same trick to $\bar{b}_{k+L-1}$ ■

A.5 Proof of Lemma 4 (Recast)

Doing the same trick again as in Lemma 3 only from the beginning of time indexes instead of the end, we have that

$$\begin{aligned}&\mathrm{P}\big(\textstyle\bigcap_{\ell=k}^{k+L}\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}|b_k,\pi\big)=\mathrm{P}\big(\{x_k\in\mathcal{X}_k^{\text{safe}}\}|b_k\big)\cdot\\&\textstyle\int_{\bar{b}_{k+1}}\int_{z_{k+1}}\mathbb{P}\big(\bar{b}_{k+1}|\{x_k\in\mathcal{X}_k^{\text{safe}}\},b_k,\pi,z_{k+1}\big)\mathbb{P}\big(z_{k+1}|\{x_k\in\mathcal{X}_k^{\text{safe}}\},b_k,\pi\big)\mathrm{d}z_{k+1}\mathrm{P}\big(\textstyle\bigcap_{\ell=k+1}^{k+L}\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}|\bar{b}_{k+1},\pi\big)\mathrm{d}\bar{b}_{k+1}\end{aligned}\tag{118}$$

leading to the desired result. Following the notations in [30] we have that

$$\mathrm{er}_k(b_k,\pi) = 1-(1-r_b(b_k))\cdot\int_{\bar{b}_{k+1}}\mathbb{P}\big(\bar{b}_{k+1}|\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\},b_k,\pi\big)\mathrm{P}\big(\bigcap_{\ell=k+1}^{k+L}\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\}|\bar{b}_{k+1},\pi\big)\mathrm{d}\bar{b}_{k+1} = \tag{119}$$

$$1-(1-r_b(b_k))\cdot\int_{z_{k+1}}\mathrm{P}\big(z_{k+1}|\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\},b_k,\pi\big)(1-\mathrm{er}_{k+1}(\bar{b}_{k+1},\pi))\mathrm{d}z_{k+1} = \tag{120}$$

$$1-(1-r_b(b_k))\cdot\big(1-\int_{z_{k+1}}\mathrm{P}\big(z_{k+1}|\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\},b_k,\pi\big)\mathrm{er}_{k+1}(\bar{b}_{k+1},\pi)\mathrm{d}z_{k+1}\big) = \tag{121}$$

$$r_b(b_k)+(1-r_b(b_k))\int_{z_{k+1}}\mathrm{P}\big(z_{k+1}|\{x_k\in\mathcal{X}_k^{\mathrm{safe}}\},b_k,\pi\big)\mathrm{er}_{k+1}(\bar{b}_{k+1},\pi)\mathrm{d}z_{k+1}. \tag{122}$$

Hence, we have the equivalence as asserted. ■

A.6 Proof of Lemma 5 (Absolute continuity of observation likelihoods)

Assume in contradiction that absolute continuity does not hold. That is, there exists observation $z_{\ell+1}=\zeta$ such that $\mathbb{P}(z_{\ell+1}=\zeta|b_k,\pi,z_{k+1:\ell})=0$ and $\mathbb{P}(z_{\ell+1}=\zeta|b_k,\pi,z_{k+1:\ell},\bigcap_{i=k}^{\ell}\{x_i\in\mathcal{X}_i^{\mathrm{safe}}\})>0$. Applying Bayes rule we have that

$$\mathbb{P}(z_{\ell+1}=\zeta|b_k,\pi,z_{k+1:\ell},\bigcap_{i=k}^{\ell}\{x_i\in\mathcal{X}_i^{\mathrm{safe}}\}) = \frac{\mathrm{P}(\bigcap_{i=k}^{\ell}\{x_i\in\mathcal{X}_i^{\mathrm{safe}}\}|b_k,\pi,z_{k+1:\ell},z_{\ell+1}=\zeta)\mathbb{P}(z_{\ell+1}=\zeta|b_k,\pi,z_{k+1:\ell})}{\mathbb{P}(\bigcap_{i=k}^{\ell}\{x_i\in\mathcal{X}_i^{\mathrm{safe}}\}|b_k,\pi,z_{k+1:\ell})}=0. \tag{123}$$

■

Appendix B Posterior Conditioned on the Safe Prior (Section 5.3)

The safe event influence Belief-MDP motion model in the following way

$$\mathbb{P}\big(\bar{b}'|\bar{b},a,\{x\in\mathcal{X}^{\mathrm{safe}}\}\big)=\int_{z'\in\mathcal{Z}}\mathbb{P}\big(\bar{b}'|\bar{b},a,z',\{x\in\mathcal{X}^{\mathrm{safe}}\}\big)\mathbb{P}\big(z'|a,\bar{b},\{x\in\mathcal{X}^{\mathrm{safe}}\}\big)\mathrm{d}z' =$$
$$\int_{z'\in\mathcal{Z}}\delta(\bar{b}'-\psi(\bar{b}^{\mathrm{safe}},a,z'))\mathbb{P}(z'|a,\bar{b},\{x\in\mathcal{X}^{\mathrm{safe}}\})\mathrm{d}z' \tag{124}$$

We first calculate the propagated belief conditioned on the safe prior.

$$\mathbb{P}\big(x'|\bar{b},a,\big\{x\in\mathcal{X}^{\mathrm{safe}}\big\}\big) = \frac{\int_{x\in\mathcal{X}}\mathbf{1}_{\{x\in\mathcal{X}^{\mathrm{safe}}\}}(x)\mathbb{P}_{\mathrm{T}}(x'|x,a)\bar{b}(x)\mathrm{d}x}{\int_{\xi\in\mathcal{X}}\mathbf{1}_{\{\xi\in\mathcal{X}^{\mathrm{safe}}\}}(\xi)\bar{b}(\xi)\mathrm{d}\xi} \tag{125}$$

b and event safe, meaning that belief supposed to be zero at non safe places. Finally,

$$\mathbb{P}\big(z'|a,\bar{b},\{x\in\mathcal{X}^{\mathrm{safe}}\}\big)=\int_{x'\in\mathcal{X}'}\mathbb{P}_{\mathrm{O}}(z'|x')\mathbb{P}\big(x'|\bar{b},a,\{x\in\mathcal{X}^{\mathrm{safe}}\}\big)\mathrm{d}x'. \tag{126}$$

We can also look at the above from slightly different angle. We define $\bar{b}^{\mathrm{safe}}$ as in (51) such that

$$\mathbb{P}\big(x'|\bar{b},a,\{x\in\mathcal{X}^{\mathrm{safe}}\}\big)=\int_{x\in\mathcal{X}}\mathbb{P}_{\mathrm{T}}(x'|x,a)\bar{b}^{\mathrm{safe}}(x)\mathrm{d}x. \tag{127}$$

We can use the safe belief defined above in the belief update as follows

$$\mathbb{P}(x'|\bar{b},a,z',\{x\in\mathcal{X}^{\mathrm{safe}}\}) = \frac{\mathbb{P}(z'|\bar{b},a,x',\{x\in\mathcal{X}^{\mathrm{safe}}\})\mathbb{P}(x'|\bar{b},a,\{x\in\mathcal{X}^{\mathrm{safe}}\})}{\mathbb{P}(z'|\bar{b},a,\{x\in\mathcal{X}^{\mathrm{safe}}\})} = \frac{\mathbb{P}_{\mathrm{O}}(z'|x')\mathbb{P}(x'|\bar{b},a,\{x\in\mathcal{X}^{\mathrm{safe}}\})}{\int_{\xi'}\mathbb{P}_{\mathrm{O}}(z'|\xi')\mathbb{P}(\xi'|\bar{b},a,\{x\in\mathcal{X}^{\mathrm{safe}}\})\mathrm{d}\xi'} \tag{128}$$

Now the ψ is conventional belief update operator receiving as input $\psi(\bar{b}^{\mathrm{safe}},a,z')$.

Appendix C Derivation of the Importance Weights (Section 6.1)

To calculate the likelihoods of the observations we shall do the following. Suppose that the belief is represented by samples. For simplicity we show calculation for belief b_k. However this is not a limitation. Conditioning on different beliefs in two observation likelihoods is supported without any change. The only necessity is two different beliefs represented by weighted particles.

$$b_k(x_k)\approx\textstyle\sum_{i=1}^{N}w_k^i\delta(x_k-x_k^i), \tag{129}$$

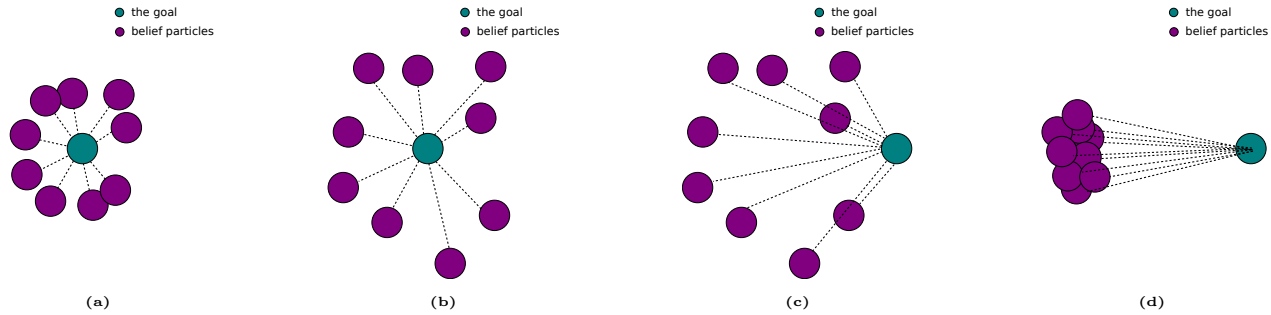


Fig. 16: Geometrical visualization of the natural belief uncertainty measure imprinted in the mean distance to the goal. **(a)** Less spread results in lowering all the distances, thereby the mean. **(b)** The reciprocal situation. **(c)** Another situation, here, to decrease the mean distance to the goal, one has to reduce the spread and the distance between the expected value of the belief and the goal. **(d)** The spread is decreased, but the distance between the expected value of the belief and the goal is large (Appendix D).

Let us introduce another notation $\delta(x_k - x_k^i) = \delta^{x_k^i}(x_k)$ so

$$\mathbb{P}\big(x_{k+1}|b_k, a_k, \{x_k \in \mathcal{X}_k^{\text{safe}}\}\big) = \frac{\int_{x_k \in \mathcal{X}} \mathbf{1}_{\{x_k \in \mathcal{X}_k^{\text{safe}}\}}(x_k)\mathbb{P}_{\text{T}}(x_{k+1}|x_k, a_k)b_k(x_k)\mathrm{d}x_k}{\int_{\xi_k \in \mathcal{X}} \mathbf{1}_{\{\xi_k \in \mathcal{X}_k^{\text{safe}}\}}(\xi_k)b_k(\xi_k)\mathrm{d}\xi_k} \approx \tag{130}$$

$$\frac{\int_{x_k \in \mathcal{X}} \mathbf{1}_{\{x_k \in \mathcal{X}_k^{\text{safe}}\}}(x_k)\mathbb{P}_{\text{T}}(x_{k+1}|x_k, a_k)\big(\sum_{i=1}^N w_k^i \delta^{x_k^i}(x_k)\big)\mathrm{d}x_k}{\int_{\xi_k \in \mathcal{X}} \mathbf{1}_{\{\xi_k \in \mathcal{X}_k^{\text{safe}}\}}(\xi_k)\big(\sum_{i=1}^N w_k^i \delta^{x_k^i}(x_k)\big)\mathrm{d}\xi_k} = \tag{131}$$

$$\frac{\sum_{i=1}^N w_k^i \mathbf{1}_{\{x_k \in \mathcal{X}_k^{\text{safe}}\}}(x_k^i)\mathbb{P}_{\text{T}}(x_{k+1}|x_k^i, a_k)}{\sum_{i=1}^N w_k^i \mathbf{1}_{\{x_k \in \mathcal{X}_k^{\text{safe}}\}}(x_k^i)} \approx \frac{\sum_{i=1}^N w_k^i \mathbf{1}_{\{x_k \in \mathcal{X}_k^{\text{safe}}\}}(x_k^i)\delta^{x_{k+1}^i}(x_{k+1})}{\sum_{i=1}^N w_k^i \mathbf{1}_{\{x_k \in \mathcal{X}_k^{\text{safe}}\}}(x_k^i)}. \tag{132}$$

We got that

$$\mathbb{P}(z_{k+1} = z_{k+1}^j|b_k, \{x_k \in \mathcal{X}_i^{\text{safe}}\}, a_k) \approx \frac{\sum_{i=1}^N w_k^i \mathbf{1}_{\{x_k \in \mathcal{X}_k^{\text{safe}}\}}(x_k^i)\mathbb{P}_{\text{O}}(z_{k+1}^j|x_{k+1}^i)}{\sum_{i=1}^N w_k^i \mathbf{1}_{\{x_k \in \mathcal{X}_k^{\text{safe}}\}}(x_k^i)}. \tag{133}$$

In case of the denominator we arrive to the same expression, only without the indicator.

$$\mathbb{P}(z_{k+1} = z_{k+1}^j|b_k, a_k) \approx \frac{\sum_{i=1}^N w_k^i \mathbb{P}_{\text{O}}(z_{k+1}^j|x_{k+1}^i)}{\sum_{i=1}^N w_k^i}. \tag{134}$$

In reality, however, it is possible that after we discard all the samples of the belief which are not safe we are left with a very small set of samples or an empty set. To alleviate this issue we resample the safe particles to a constant number of samples N.

Appendix D Mean Distance to Goal Accountability for Uncertainty

In this section, we discuss in depth why the mean distance to goal intrinsically accounts for belief uncertainty. We show a geometrical visualization in Fig. 16. Since the distance is no negative, the less spread of belief implies lower distances and vise versa. Further, let us show that algebraically.

Theorem 2. *Let y be an arbitrary distributed random vector with μ_y and Σ_y being the expected value and covariance matrix of y, respectively; and let Λ be arbitrary matrix. The following relation is correct*

$$\mathbb{E}[y^T \Lambda y] = \text{tr}\,[\Lambda \Sigma_y] + \mu_y^T \Lambda \mu_y, \tag{135}$$

where by tr *we denote the trace operator.*

Proof. Since the quadratic form is a scalar quantity, $y^T \Lambda y = \text{tr}(y^T \Lambda y)$. Next, by the cyclic property of the trace operator,

$$\mathbb{E}[\text{tr}(y^T \Lambda y)] = \mathbb{E}[\text{tr}(\Lambda y y^T)]. \tag{136}$$

Since the trace operator is a linear combination of the components of the matrix, it therefore follows from the linearity of the expectation operator that

$$\mathbb{E}[\text{tr}(\Lambda y y^T)] = \text{tr}(\Lambda \mathbb{E}(y y^T)). \tag{137}$$

A standard property of variances then tells us that this is

$$\mathrm{tr}(\Lambda(\Sigma_x + \mu_x \mu_x^T)). \tag{138}$$

Applying the cyclic property of the trace operator again, we get

$$\mathrm{tr}(\Lambda\Sigma_y) + \mathrm{tr}(\Lambda\mu_y\mu_y^T) = \mathrm{tr}(\Lambda\Sigma_y) + \mathrm{tr}(\mu_y^T\Lambda\mu_y) = \mathrm{tr}(\Lambda\Sigma_y) + \mu_y^T\Lambda\mu_y. \tag{139}$$

■

Now, we set $y = x - x^g$, where $x \sim b$ with the mean μ_x and covariance matrix Σ_x; x^g is the deterministic goal location. Recall that covariance matrix is invariant to the deterministic translational shifts of a random vector, so $\Sigma_{x-x^g} = \Sigma_x$. Moreover, by setting $\Lambda = I$ we obtain

$$\mathbb{E}[y^T\Lambda y] = \mathbb{E}[(x-x^g)^T I (x-x^g)] = \mathbb{E}[\|x-x^g\|_2^2] = \mathrm{tr}[\Sigma_x] + \|\mu_x - x^g\|_2^2. \tag{140}$$

We arrived at the desired result. As we observe in Fig. 16, the trace of the covariance matrix controls the spread of the belief in the first summand; the second summand is the distance between the expected value of the belief.

Appendix E Necessary Condition for Feasibility of CC (Lemma 6)

In this section, we develop necessary condition for feasibility of CC from [30]. Through IS we extend the condition presented in [30] to continuous spaces in terms of states and the observations.

Suppose that $0 \le \Delta \le 1$ and $\mathrm{er}_\ell(\bar{b}_\ell, \pi) \le \Delta$. From now on for clarity suppose the weights are already normalized. We use (74).

$$r_b(\bar{b}_\ell) + (1 - r_b(\bar{b}_\ell)) \sum_{j=1}^m w_{\ell+1}^{z,j} \mathrm{er}_{\ell+1}(\bar{b}_{\ell+1}(h_\ell a_\ell z_{\ell+1}^j), \pi) \le \Delta, \tag{141}$$

We choose some child $j = i$ and arrive at

$$\mathrm{er}_{\ell+1}(\bar{b}_{\ell+1}(h_\ell a_\ell z_{\ell+1}^i), \pi) \le \tfrac{1}{w_{\ell+1}^{z,i}} \big(\tfrac{\Delta - r_b(\bar{b}_\ell)}{(1-r_b(\bar{b}_\ell))} - \textstyle\sum_{\substack{j=1 \\ j\neq i}}^m w_{\ell+1}^{z,j} \mathrm{er}_{\ell+1}(\bar{b}_{\ell+1}(h_\ell a_\ell z_{\ell+1}^j), \pi) \big), \tag{142}$$

If $r_b(\bar{b}_\ell)$=1, so $\mathrm{P}(\{x_\ell \in \mathcal{X}_\ell^{\mathrm{safe}}\} | \bar{b}_\ell)$=0. Namely, at each x_ℓ or $\bar{b}_\ell(x_\ell) = 0$ or $x_\ell \notin \mathcal{X}_k^{\mathrm{safe}}$. In this case the probability density

$$\mathbb{P}(z_{\ell+1} = z_{\ell+1}^j | b_k, \pi, z_{k+1:\ell}, \textstyle\bigcap_{i=k}^\ell \{x_i \in \mathcal{X}_i^{\mathrm{safe}}\}) \tag{143}$$

is undefined due to conditioning on the empty set. We will need to discard such policy before pruning due to the chance constraint violation. Now we show the inverse relation. If $r_b(\bar{b}_\ell)$<1, that is $\mathrm{P}(\{x_\ell \in \mathcal{X}_\ell^{\mathrm{safe}}\} | \bar{b}_\ell)$>0. This implies that $\bar{b}_\ell^{\mathrm{safe}}$ has support larger that empty set. Suppose that motion and observation models has infinite support. This imply that

$$\mathbb{P}(z_{\ell+1}{=}z_{\ell+1}^j | b_k, \pi, z_{k+1:\ell}, \textstyle\bigcap_{i=k}^\ell \{x_i{\in}\mathcal{X}_i^{\mathrm{safe}}\}){=} \int\limits_{x_{\ell+1}} \mathbb{P}_{\mathrm{O}}(z_{\ell+1}{=}z_{\ell+1}^j | x_{\ell+1}) \int\limits_{x_\ell} \mathbb{P}_{\mathrm{T}}(x_{\ell+1}|x_\ell, a_\ell) \bar{b}_\ell^{\mathrm{safe}}(x_\ell) \mathrm{d}x_\ell \mathrm{d}x_{\ell+1} > 0. \tag{144}$$

These arguments are valid if we approximate the belief by weighted samples since we approximate the probability $\mathrm{P}(\{x_\ell \in \mathcal{X}_\ell^{\mathrm{safe}}\} | \bar{b}_\ell)$ using same samples. If the models do not posses infinite support leading to $w_{\ell+1}^{z,i}$=0 we just skip the child i and not prune. Anyway this is only the necessary condition and we will need to verify the CC for not pruned policies. To do pruning using technique from [30] we note that $r_b(\bar{b}_{\ell+1}) \le \mathrm{er}_{\ell+1}(\bar{b}_{\ell+1}, \pi)$ and got that

$$r_b(\bar{b}_{\ell+1}^i) \le \tfrac{1}{w_{\ell+1}^{z,i}} \big(\tfrac{\Delta - r_b(\bar{b}_\ell)}{(1-r_b(\bar{b}_\ell))} - \textstyle\sum_{\substack{j=1 \\ j\neq i}}^m w_{\ell+1}^{z,j} r_b(\bar{b}_{\ell+1}^j) \big). \tag{145}$$

If $\mathrm{er}_\ell(\bar{b}_\ell, \pi) \le \Delta$ the above shall hold for every child of $\bar{b}_\ell$. ■

References

[1] Michal Ajdarów, Šimon Brlej, and Petr Novotnỳ. Shielding in resource-constrained goal pomdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14674–14682, 2023. 3

[2] Mauricio Araya, Olivier Buffet, Vincent Thomas, and Françcois Charpillet. A pomdp extension with belief-dependent rewards. In *Advances in Neural Information Processing Systems (NIPS)*, pages 64–72, 2010. 2

[3] Brian Axelrod, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Provably safe robot navigation with obstacle uncertainty. *The International Journal of Robotics Research*, 37(13-14):1760–1774, 2018. 3

[4] M. Barenboim and V. Indelman. Adaptive information belief space planning. In *the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI)*, July 2022. 12

[5] Y. Boers, H. Driessen, A. Bagchi, and P. Mandal. Particle filter based entropy. In *2010 13th International Conference on Information Fusion*, pages 1–8, 2010. 6

[6] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 723–730, 2011. 2, 3, 6, 7, 9, 15, 26

[7] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian D Reid, and John J Leonard. Simultaneous localization and mapping: Present, future, and the robust-perception age. *IEEE Trans. Robotics*, 32(6):1309 – 1332, 2016. 4, 8

[8] Louis Dressel and Mykel J. Kochenderfer. Efficient decision-theoretic target localization. In Laura Barbulescu, Jeremy Frank, Mausam, and Stephen F. Smith, editors, *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS 2017, Pittsburgh, Pennsylvania, USA, June 18-23, 2017*, pages 70–78. AAAI Press, 2017. 2

[9] Mathieu Fehr, Olivier Buffet, Vincent Thomas, and Jilles Dibangoye. rho-pomdps have lipschitz-continuous epsilon-optimal value functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6933–6943. Curran Associates, Inc., 2018. 2

[10] Kristoffer M Frey, Ted J Steiner, and Jonathan P How. Collision probabilities for continuous-time systems without sampling [with appendices]. *arXiv preprint arXiv:2006.01109*, 2020. 3

[11] Astghik Hakobyan, Gyeong Chan Kim, and Insoon Yang. Risk-aware motion planning and control using cvar-constrained optimization. *IEEE Robotics and Automation Letters*, 4(4):3924–3931, 2019. 9

[12] Weiqiao Han, Ashkan Jasour, and Brian Williams. Non-gaussian risk bounded trajectory optimization for stochastic nonlinear systems in uncertain environments. *arXiv preprint arXiv:2203.03038*, 2022. 3

[13] Qi Heng Ho, Tyler Becker, Ben Kraske, Zakariya Laouar, Martin Feather, Federico Rossi, Morteza Lahijanian, and Zachary N Sunberg. Recursively-constrained partially observable markov decision processes. *arXiv preprint arXiv:2310.09688*, 2023. 4

[14] Marcus Hoerger, Hanna Kurniawati, and Alberto Elfes. Multilevel monte carlo for solving pomdps on-line. In *Intl. J. of Robotics Research*, volume 42, pages 196–213. Sage Publications Sage UK: London, England, 2023. 19

[15] David Hsu, Wee Sun Lee, and Nan Rong. A point-based pomdp planner for target tracking. In *2008 IEEE International Conference on Robotics and Automation*, pages 2644–2650. IEEE, 2008. 29

[16] V. Indelman, L. Carlone, and F. Dellaert. Planning in the continuous domain: a generalized belief space approach for autonomous navigation in unknown environments. *Intl. J. of Robotics Research*, 34(7):849–882, 2015. 2

[17] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, 1996. 27

[18] Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine learning*, 49(2):193–208, 2002. 2, 10, 12, 13, 20

[19] Majid Khonji and Duoaa Khalifa. Heuristic search in dual space for constrained fixed-horizon pomdps with durative actions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14927–14936, 2023. 3

[20] M. Kochenderfer, T. Wheeler, and K. Wray. *Algorithms for Decision Making*. MIT Press, 2022. 2, 4

[21] Jongmin Lee, Geon-Hyeong Kim, Pascal Poupart, and Kee-Eung Kim. Monte-carlo tree search for constrained pomdps. *Advances in Neural Information Processing Systems*, 31, 2018. 3

[22] Michael H Lim, Tyler J Becker, Mykel J Kochenderfer, Claire J Tomlin, and Zachary N Sunberg. Generalized optimality guarantees for solving continuous observation pomdps through particle belief mdp approximation. *arXiv preprint arXiv:2210.05015*, 2022. 13

[23] Michael H Lim, Tyler J Becker, Mykel J Kochenderfer, Claire J Tomlin, and Zachary N Sunberg. Optimality guarantees for particle belief approximation of pomdps. *Journal of Artificial Intelligence Research*, 77:1591–1636, 2023. 13

[24] Michael H. Lim, Claire Tomlin, and Zachary N. Sunberg. Sparse tree search optimality guarantees in pomdps with continuous observation spaces. In *Intl. Joint Conf. on AI (IJCAI)*, pages 4135–4142, 7 2020. 13

[25] Giulio Mazzi, Alberto Castellini, and Alessandro Farinelli. Risk-aware shielding of partially observable monte carlo planning policies. *Artificial Intelligence*, 324:103987, 2023. 3

[26] Robert J Moss, Arec Jamgochian, Johannes Fischer, Anthony Corso, and Mykel J Kochenderfer. Constrainedzero: Chance-constrained pomdp planning using learned probabilistic failure surrogates and adaptive safety constraints. *arXiv preprint arXiv:2405.00644*, 2024. 3

[27] C. Papadimitriou and J. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987. 1

[28] Julio A Placed and José A Castellanos. Enough is enough: Towards autonomous uncertainty-driven stopping criteria. *arXiv preprint arXiv:2204.10631*, 2022. 2, 4

[29] Samantha Samuelson and Insoon Yang. Safety-aware optimal control of stochastic systems using conditional value-at-risk. In *2018 Annual American Control Conference (ACC)*, pages 6285–6290. IEEE, 2018. 9

[30] Pedro Santana, Sylvie Thiébaux, and Brian Williams. Rao*: An algorithm for chance-constrained pomdp's. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. 2, 3, 4, 5, 7, 9, 11, 13, 15, 16, 17, 18, 20, 21, 22, 30, 35, 37

[31] M. Shienman and V. Indelman. Nonmyopic distilled data association belief space planning under budget constraints. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2022. 19

[32] David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2164–2172, 2010. 2

[33] Zachary Sunberg and Mykel Kochenderfer. Online algorithms for pomdps with continuous state, action, and observation spaces. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28, 2018. 2

[34] Zachary Sunberg and Mykel J. Kochenderfer. POMCPOW: an online algorithm for pomdps with continuous state, action, and observation spaces. *CoRR*, abs/1709.06196, 2017. 20

[35] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT press, Cambridge, MA, 2005. 3

[36] Aditya Undurti and Jonathan P How. An online algorithm for constrained pomdps. In *2010 IEEE International Conference on Robotics and Automation*, pages 3966–3973. IEEE, 2010. 3

[37] Nan Ye, Adhiraj Somani, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. *JAIR*, 58:231–266, 2017. 2

[38] Andrey Zhitnikov and Vadim Indelman. Simplified continuous high dimensional belief space planning with adaptive probabilistic belief-dependent constraints. *IEEE Trans. Robotics*, 2024. 2, 4, 9, 14

[39] Andrey Zhitnikov, Ori Sztyglic, and Vadim Indelman. No compromise in solution quality: Speeding up belief-dependent continuous pomdps via adaptive multilevel simplification. *arXiv preprint arXiv:2310.10274*, 2023. 2

Anytime Probabilistically Constrained Provably Convergent Online Belief Space Planning

Andrey Zhitnikov[1] and Vadim Indelman[2,3]
[1]Technion Autonomous Systems Program (TASP)
[2]Department of Aerospace Engineering
[3]Department of Data and Decision Science
Technion - Israel Institute of Technology, Haifa 32000, Israel
andreyz@campus.technion.ac.il, vadim.indelman@technion.ac.il

***Abstract*—Taking into account future risk is essential for an autonomously operating robot to find online not only the best but also a safe action to execute. In this paper, we build upon the recently introduced formulation of probabilistic belief-dependent constraints. We present an anytime approach employing the Monte Carlo Tree Search (MCTS) method in continuous domains. Unlike previous approaches, our method assures safety anytime with respect to the currently expanded search tree without relying on the convergence of the search. We prove convergence in probability with an exponential rate of a version of our algorithms and study proposed techniques via extensive simulations. Even with a tiny number of tree queries, the best action found by our approach is much safer than the baseline. Moreover, our approach constantly finds better than the baseline action in terms of objective. This is because we revise the values and statistics maintained in the search tree and remove from them the contribution of the pruned actions.**

***Index Terms*—MCTS, BSP, Belief-dependent constraints, Anytime Constraint Satisfaction**

I. Introduction and Related Work

CASTING decision-making under uncertainty as a Partially Observable Markov Decision Process (POMDP) is considered State-Of-The-Art (SOTA). Under partial observability the decision-making agent does not have complete information about the state of the problem, so it can only make its decisions based on its "belief" about the state. In a continuous domains in terms of POMDP state, the belief, in a particular time index, is the Probability Density Function (PDF) of the state given all concurrent information in terms of performed actions and received observations in an alternating manner, plus the prior belief. A POMDP is known to be undecidable [1] in finite time.

Introducing various constraint formulations into POMDP is essential for, e.g., ensuring safety [2], [3] and efficient Autonomous Exploration [4]. Yet, the existing online approaches in anytime setting have problems and therefore fall short of providing reliable and safe optimal autonomy. This crucial gap we aim to fill in this paper.

Similar to almost any online POMDP solver today such as MCTS, our method constructs a belief tree and uses the tree to represent the POMDP policy. We prune dangerous actions from the belief tree and revise the values and statistics that an MCTS tree maintains. Anytime, our search tree contains only the safe actions in accord to our definition of safe action, which will appear shortly. Our work lies in continuous domain in terms of actions and the observations. In such a setting, there are approaches to tackle averaged cumulative constraint using anytime MCTS methods [5], [6]. We now linger on the explanation of what the averaged constraint is.

Under partial observability, namely in the POMDP setting, there are naturally two stages to consider in order to introduce a constraint. The first stage arises from the belief itself. Usually, at this stage, the state-dependent payoff operator is averaged with respect to the corresponding belief to obtain a belief-dependent one. It is then summed up to achieve a cumulative payoff. We use the term payoff to differentiate between reward operator and emphasize that a belief-dependent payoff constraint operator shall be as large as possible as opposed to the cost operator. The second stage arises from the distribution of possible future observations episodes. At this stage, commonly, the cumulative payoff is again averaged but with respect to future observations episodes and then thresholded, thereby forming an averaged cumulative constraint. Such a formulation is sufficient for ensuring safety in limited cases as we will further see in Section VI-A. This is because it permits deviations of the individual values within the summation.

Let us now describe the MCTS methods mentioned above to tackle averaged cumulative constraint. The seminal paper in this direction is [7]. It leans on the rearrangement of the constrained objective using the occupancy measure described in [8]. Such a reformulation is appealing since it transforms the problem into linear programming bringing convexity to the table and enjoying from strong duality. The authors of [5] extend the approach from [7] to continuous spaces. Still, both papers [7] and [5] assure constraint satisfiability only at the limit of the convergence of the iterative procedure, namely in infinite time. Since these are iterative methods, to assure anytime constraint satisfiability we need to project the obtained occupancy measure at each iteration to the space defined by the constraint. If dual methods are involved [9] such a projection does not make much sense, e.g., the projection might lead to a step direction vector on the boundary of all the constraints, making it zero vector. Employing the primal methods in continuous spaces also appears to be problematic since the summations in [7] are transformed into integrals. The paper [6] provides some sort of anytime satisfiability

This work was supported by the Israel Science Foundation (ISF).

by introducing high-level action primitives (options). Still, [6] suffers from limitations, e.g. it requires crafting low-level policies, meaning knowing how the robot shall behave a priori. In addition, the options shall be locally feasible. Additionally, for efficiency reasons, the duality based approaches perform a single tree query of the MCTS, instead of running MCTS until convergence in the maximization of the Lagrangian dual objective function phase (See section 8.5.2 in [9]) of dual ascend.

In all three papers [7], [5], [6] the averaged cumulative constraint is enforced solely from the root of the belief tree. This is suboptimal since within a planning session it is not taken into account that the constraint will be enforced at the future planning sessions. In other words, the contemplation of a robot about the future differs from its actual future behavior. This aspect has been fixed by [10]. As we will further see in Section IV, our approach naturally handles this problem. Moreover, [10] assures fulfillment (admission) of the recursive averaged cumulative constraint anytime with respect to search tree constructed partially with the reward bounds and partially with rewards themselves. Yet, the algorithm presented in [10] requires that the value function is bounded on the way down the tree to assure the exploration. This is commonly achieved by assuming that the state-dependent reward is trivially bounded from above and below. This does not hold for general belief-dependent reward functions. Moreover, the exploration outlined in that paper is valid for discrete spaces only. All in all, the extension of that work to continuous spaces and belief-dependent rewards requires clarification.

a) Support for general belief dependent rewards and payoff/cost operators and MCTS convergence: We now clarify whether or not the mentioned above solvers support belief-dependent cost/payoff operators and rewards. It was suggested in [3],[4] that general belief-dependent payoff/cost operators are extremely important. As mentioned in [3] Value-at-Risk (VaR) and Conditional VaR (CVaR) over the distance to the safe space allow for control of the depth the robot can plunge into the obstacle. To rephrase that, these operators measure how bad the disaster (collision) will be. See Appendix D, for details. The Information Gain discussed in [4] is relevant for exploration. The paper [4] discussed the general belief-dependent averaged constraint of the form (38) in a high dimensional setting and in the context of Information Gain. The iterative schemes in [7], [5] lean on the convergence of MCTS. It has been shown in [11] that even in discrete spaces and with bounded rewards it can take a very long time for MCTS to converge. In the case of unbounded reward or the cost-augmented objective of [7], [5], the MCTS may converge slowly. If such an augmented reward has a large variance, it will be needed a huge amount of tree queries for action-value estimate (to be defined shortly) at each belief node of the belief tree to converge. The large variance can be the result of an unrestrained variability of the rewards or a large Lagrange multiplier.

There are several constraint formulations for POMDP. Below we discuss the most prominent techniques one by one.

b) Shielding POMDPs: There is a growing body of literature on shielding POMDPs. The shield is a technique to disable the actions that can be executed by the agent and violate the shield definition. There are several shield definitions. Online methods [12], [13] in this category utilize Partially Observable Monte-Carlo Planning (POMCP) algorithm [14]. These works have the same problems we are solving in this paper: one way or another, the actions violating the shield definition participate in the planning procedure, yielding a suboptimal result. The work [13] enforces the shied outside the POMCP planning. As we further show, not considering safety in the future times, namely within the planning session, can lead to a suboptimal planning result.

c) Chance Constrained (CC) Online Planning: A recent work [15] tackles online planning with chance constraints in an anytime setting. This paper suggests using a Neural Network (NN) to approximate CC enforced, with an adaptive threshold, from each belief considered in the planning session. This work trains NN offline. Therefore the error stemming from the discrepancy of simulated and real data is unknown. Moreover, it is not clear how complex the NN shall be to achieve zero loss in training to ensure no error in CC approximation, so even if no discrepancy discussed before exists, the NN inference may be slow. In this method, dangerous actions do not participate in the planning session.

d) Safe control Under Partial Observability: There are a variety of robust control approaches natively tailored for continuous state/action/observation spaces [16],[17]. However, these methods are usually limited to very specific rewards/objectives and tasks, such as reaching a goal state or to be as close as possible to a nominal trajectory. Moreover, in both papers the system dynamics are control-affine. Without this assumption, it is not clear how to enforce the constraint through a derivative of the barrier function.

A. Contributions

Below we list down our contributions in the same order as they appear in the manuscript.

- By constraining directly the problem space and not the dual space we present an anytime MCTS based algorithm for safe online decision making with safety governed by a Probabilistic Constraint (PC). Our approach enjoys anytime safety guarantees with respect to the belief-tree expanded so far and works in continuous state, action and observation spaces. When stopped anytime, the action returned can be considered as the best safe action under the safe future policy (tree policy) expanded so far. Our search tree **solely** consists of safe actions. We prove convergence in probability with an exponential rate of our approach.
- Another contribution on our end is constraining the beliefs with incorporated outcome uncertainty stemming from an action performed by the robot and without incorporating the received observation. This is alongside the constraint over the posterior belief with included last observation. To the best of our knowledge, no previous works do that.
- We also spot a problem happening in duality based approaches arising from averaging unsafe actions in MCTS

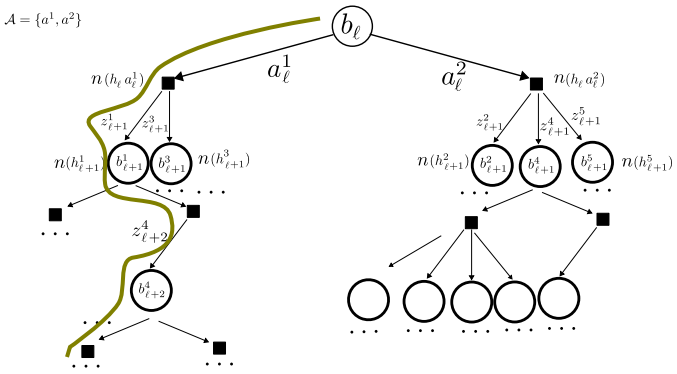


Fig. 1: Here we plot the asymmetric search tree approximating stochastic future policy. For simplicity the action space here is $\mathcal{A}=\{a^1,a^2\}$. We behold that many actions emanating from each belief node and each action has weight defined by relevant visitation count as in (8). Thus, the MCTS approximates stochastic future policy. Note that here the observations and beliefs has global index (superscript) while actions have local index according to the action number in the space $\mathcal{A}$.

phase. Therefore, an additional contribution of ours is an analysis of this phenomenon.

- We simulate our finding on several continuous POMDP problems.

B. Notation

We use the $\square$ as a placeholder for various quantities. The values in $\square$ can be replaced by one of the respective options. We also extensively use the indicator function notation, which is $\mathbf{1}_A(\square)$. This function equals to one if and only if $\square\in A$. By lowercase letters we denote the random variables of their realizations depending on context. By the bold font we denote vectors of operators in time of different lengths. We denote estimated values by $\hat{\square}$.

C. Paper Roadmap

This paper proceeds with the following structure. Section II presents relevant background. Section III then formulates the problem. Section IV presents our approach. Section VI discusses our baseline. Section VII gives experimental validation of the proposed methodology. Finally, Section VIII concludes the paper.

II. Background

This section gives the background required for presenting our approach. Specifically, we discuss belief-dependent POMDP, its reformulation to Belief-MDP (BMDP), and the MCTS.

A. Belief-dependent POMDP

The POMDP is a tuple $\langle\mathcal{X},\mathcal{A},\mathcal{Z},\mathrm{T},\mathrm{O},\rho,\gamma,b_0\rangle$ where $\mathcal{X},\mathcal{A},\mathcal{Z}$ represent continuous state, action, and observation spaces with $x\in\mathcal{X}$, $a\in\mathcal{A}$, $z\in\mathcal{Z}$ the individual state, action, and observation, respectively. $\mathrm{T}(x',a,x)\triangleq\mathbb{P}_\mathrm{T}(x'|x,a)$ is a stochastic transition model from the past state x to the subsequent x' through action a, $\mathrm{O}(z,x)\triangleq\mathbb{P}_\mathrm{O}(z|x)$ is the stochastic observation model. $\rho{:}\mathcal{B}\times\mathcal{A}\times\mathcal{Z}\times\mathcal{B}\mapsto\mathbb{R}$ is a belief-dependent reward incurred as a result of taking an action a from the belief b, receiving and observation z' and updating the belief to b'. By $\mathcal{B}$ we denote the space of all possible beliefs. $\gamma\in(0,1]$ is the discount factor, b_0 is the prior belief. Purely for clarity of the exposition we further assume that the reward depends solely on a pair of consecutive-in-time beliefs and an action in between. In addition we suppose $\gamma{=}1$. To remove unnecessary clutter we assume that planning starts from b_0. Extension to the arbitrary planning time is straightforward.

Let h_ℓ be a history. The history is the set that comprises the prior belief b_0, the actions $a_{0:\ell-1}$ and the observations $z_{1:\ell}$ that would be obtained by the agent up to time instance ℓ such that $h_\ell\triangleq\{b_0,a_{0:\ell-1},z_{1:\ell}\}$. We emphasize by the green color that b_0 is given, but the actions $a_{0:\ell-1}$ and observations $z_{1:\ell}$ can vary. In addition due to the assumption that the planning session starts from the prior belief b_0 we can have only the future history simulated in planning in this work. For completeness we define $h_0\triangleq\{b_0\}$ The posterior belief b_ℓ is given by

$$b_\ell(x_\ell)\triangleq\mathbb{P}(x_\ell|b_0,a_{0:\ell-1},z_{1:\ell})=\mathbb{P}(x_\ell|h_\ell)=\mathbb{P}(x_\ell|b_\ell). \tag{1}$$

The belief is a function of history such that we sometimes write $b(h)$ instead of $b(x)$ and use the corresponding h notation to point to the belief $b(h)$. The actions within the history are coming from the execution policy. A deterministic policy π is a sequence of functions $\pi{=}\pi_{0:\ell-1}$ for $\ell\in[1\dots L{-}1]$, where the momentary function $\pi_i{:}\mathcal{B}\mapsto\mathcal{A}$ $\forall i$. In each time index, the policy maps belief to action. For better readability sometimes we will omit the time index for policy or denote $\pi_{0:\ell-1}$ as π_{0+} and $\pi_{1:\ell-1}$ as π_{1+} . The policy can also be stochastic. In this case, it is a distribution of taking an action a_ℓ from a belief $\pi_\ell(a_\ell,b_\ell){=}\pi_\ell(a_\ell,h_\ell){=}\mathbb{P}^\pi_\ell(a_\ell|b_\ell(h_\ell)){=}\mathbb{P}^\pi_\ell(a_\ell|h_\ell)$[1]. Here the action space $\mathcal{A}$ is the space of outcomes and the mapping is $\pi_i{:}\mathcal{B}\times\mathcal{A}\mapsto\mathbb{R}$. We have that $\pi_{0:L-1}{=}\{\mathbb{P}^\pi_i\}_{i=0}^{L-1}$. Yet, in h_ℓ we have a specific realization of actions of such a policy in previous time instances. When the agent performs an action a and receives an observation z', it shall update its belief from b to b'. Let us denote the update operator by ψ such that $b'{=}\psi(b,a,z')$. In our context, it will be a Particle Filter (PF) since we focus on the setting of nonparametric beliefs. However, this is not an inherent limitation of our approach. Any belief update method would be suitable. We define a propagated belief b'^{-} as the belief b after the robot performed an action a and before it received and observation, namely

$$b^-_\ell(x_\ell)\triangleq\mathbb{P}(x_\ell|h_{\ell-1},a_{\ell-1}){=}\mathbb{P}(x_\ell|h^-_\ell){=}\mathbb{P}(x_\ell|b^-_\ell). \tag{2}$$

We define $h^-_\ell\triangleq h_\ell\backslash\{z_\ell\}{=}\{b_0,a_{0:\ell-1},z_{1:\ell-1}\}$. The unconstrained, online decision making objective is the action-value function specified as

$$Q^\pi(b_0,a_0;\boldsymbol{\rho}_1)\triangleq\mathbb{E}^{\mathrm{T},\mathrm{O}}_{z_1}\big[\rho_1(b_0,a_0,b_1){+}V^\pi(b_1;\boldsymbol{\rho}_2)\big|b_0,a_0\big]. \tag{3}$$

Here the we added the subscript to the reward $\rho_{\square+1}(b_\square,b_{\square+1})$ to emphasize that it is a random variable and it is allowed not to specify dependency on consecutive-in-time beliefs and the

[1]Here, the capability of history being switched with the belief has to be inspected for a particular belief update. In MCTS, as we will shortly see, the stochastic policy is history-dependent and can vary even if the belief is the same at different history nodes. In this paper, the belief update is a particle filter. Therefore, the probability of obtaining the same belief at different histories is zero.

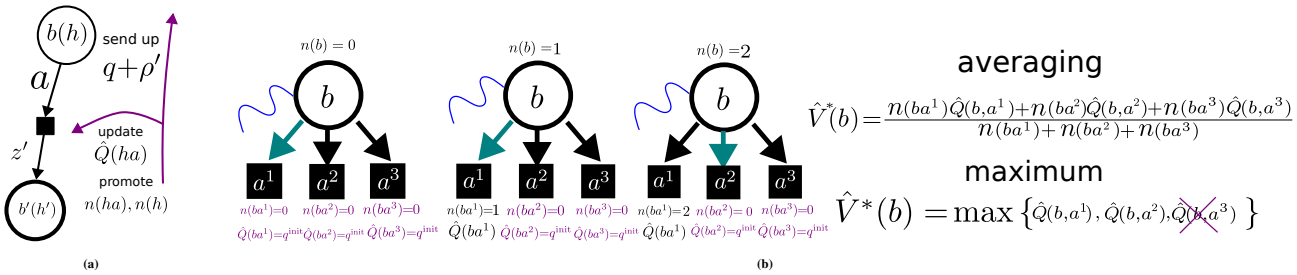


Fig. 2: **(a)** Visualization of the MCTS operations when ascending up the search tree. We update $\hat{Q}(ha)$, visitation counts $n(ha)$ and $n(h)$, send up the lace q of the cumulative reward; **(b)** Illustration of the MCTS operation when descending down the tree. First, upon reaching a leaf node, the current action space is unfolded to belief-action nodes. MCTS selects each action infinitely often. At the way up the belief tree the classical MCTS takes the average of the actions tried so far (after relevant updates on the way up) to update the estimator of (3). In this illustration, a^3 still did not tried and therefore do not participate. On the way up $n(ha^3)$ stays zero.

action in between. The $V^{\pi}(b_{\square};\boldsymbol{\rho}_{\square+1})$ is the value function under the stochastic policy π and $\boldsymbol{\rho}_{\ell}$ is a vector of belief-dependent operators of appropriate length. The value function materializes as

$$V^{\pi}(b_0;\boldsymbol{\rho}_1)\triangleq\mathbb{E}^{\mathrm{T,O}}\big[\sum_{\ell=0}^{L-1}\rho_{\ell+1}(b_\ell,a_\ell,b_{\ell+1})\big|b_0,\pi\big]. \tag{4}$$

Let us present the following lemma to better understand the structure of (4) under a stochastic policy.

Lemma 1 (Representation of the Value Function): The value function under a stochastic execution policy complies to the following form

$$\begin{aligned}&\mathbb{E}^{\mathrm{T,O}}\big[\sum_{\ell=0}^{L-1}\rho_{\ell+1}(b_\ell,a_\ell,b_{\ell+1})\big|b_0,\pi\big]=\\&\sum_{\ell=0}^{L-1}\mathbb{E}^{\mathrm{T,O}}\big[\rho_{\ell+1}(b_\ell,a_\ell,b_{\ell+1})\big|b_0,\pi\big]=\\&\sum_{\ell=0}^{L-1}\mathop{\mathbb{E}}_{a_0}\Big[\mathop{\mathbb{E}}_{b_1}\Big[\mathop{\mathbb{E}}_{a_1}\Big[\mathop{\mathbb{E}}_{b_2}\Big[\dots\\&\mathop{\mathbb{E}}_{a_\ell}\big[\mathbb{E}\big[\rho_{\ell+1}|b_\ell,a_\ell\big]\big|b_\ell,\pi_\ell\big]\dots\Big|b_1,a_1\Big]\Big|b_1,\pi_1\Big]\Big|b_0,a_0\Big]\Big|b_0,\pi_0\Big].\end{aligned} \tag{5}$$

We laid out the detailed proof in Appendix A. In online decision making, the future belief tree policy π_{1+} is approximated as part of the decision process. We denote the best future policy as $\pi^*_{(k+1)+}$. The best deterministic policy for the present time is given by $\pi_0(b_0)=\arg\max_{a_0\in\mathcal{A}}Q^{\pi^*_{1+}}(b_0,a_0;\boldsymbol{\rho}_1)$. The best stochastic policy is the solution of $\max_{\pi_\ell}\mathbb{E}_{a_\ell\sim\mathbb{P}^{\pi}_{\ell}(a_\ell|b_\ell)}[Q^{\pi^*_{(\ell+1)+}}(b_\ell,a_\ell;\boldsymbol{\rho}_{\ell+1})]$. The interlink between (4) and (3) is $V^{\pi}(b_\ell;\boldsymbol{\rho}_{\ell+1})=Q^{\pi_{(\ell+1)+}}(b_\ell,\pi_\ell(b_\ell);\boldsymbol{\rho}_{\ell+1})$ in case of deterministic policies and $V^{\pi}(b_\ell;\boldsymbol{\rho}_{\ell+1})=\mathbb{E}_{a_\ell\sim\mathbb{P}^{\pi}_{\ell}(a_\ell|b_\ell)}[Q^{\pi}(b_\ell,a_\ell;\boldsymbol{\rho}_{\ell+1})]$ in case of the stochastic policies.

B. Belief State MDP

To employ solvers crafted for fully observable Markov Decision Processes (MDP) we can cast POMDP as a Belief-MDP (BMDP). The BMDP is a following tuple $\langle\mathcal{B},\mathcal{A},\mathrm{T}_b,\rho,\gamma,b_0\rangle$, where $\mathcal{B}$ is the space of all possible beliefs defined by (1). The belief state transition model follows

$$\mathrm{T}_b(b,a,b')\triangleq\mathbb{P}_{\mathrm{T}_b}(b'|b,a)=\int_{z'\in\mathcal{Z}}\underbrace{\mathbb{P}(b'|b,a,z')}_{\delta(b'-\psi(b,a,z'))}\mathbb{P}(z'|b,a)\mathrm{d}z'. \tag{6}$$

The next section describes SOTA approach to solve unconstrained continuous POMDP online, namely MCTS. There we deal with estimators of the (3) and (4). We denote estimated values by $\hat{\square}$.

Further, we shorten the notation and mark $\hat{V}^{\pi^*}(b;\boldsymbol{\rho}')$ by $\hat{V}^*(h)$ and $\hat{Q}^{\pi^*}(ba;\boldsymbol{\rho}')$ by $\hat{Q}(ha)$. We will use the dependence on history h and the corresponding belief $b(h)$ interchangeably since the history h defines the location in the belief tree as opposed to the belief which possibly can be identical for more than single history. It will be clarified in the next section. In the next section we will see why in time zero we have deterministic policy and in future time the policy is stochastic.

C. Monte Carlo Tree Search

MCTS constructs the search tree comprised by belief nodes (transparent circles) and belief-action nodes (black squares), by iteratively descending down the tree and ascending back to the root (See Fig. 1 and 2). On the way down the tree, the exploration mechanics selects an action. The Double Progressive Widening (DPW) manages the sampling of new actions and observations. On the way back to the root MCTS updates action value estimates at each belief action node (Fig. 2a) and relevant visitation counts. In the case of belief-dependent rewards, beliefs represented by particles and continuous setting of states, actions, and observations, MCTS is applied on the level of Belief-MDP (BMDP) and called Particle Filter Tree with DPW (PFT-DPW) [18]. DPW solves the problem of shallow trees in a continuous setting. This problem arises because in this setting it is impossible to sample the same action and observation twice. The DPW technique enables gradually expanding new actions and observations as the tree search progresses. With a slight abuse of notation, we sometimes switch the dependence of various quantities on belief and dependence on the corresponding history. This is because same belief can correspond to different histories. Therefore to properly mark the position at the search tree we shall use history h instead of belief $b(h)$. The exploration score is defined as

$$\mathrm{sc}(h,a)\triangleq\underbrace{\hat{Q}(h,a)}_{\substack{\text{belief action node } ba\\ \text{indexed by } ha}}+k\sqrt{f(n(h))/n(ha)} \tag{7}$$

governs the selection of the actions down the tree, where $n(h)$ is the visitation count of the belief nodes, $n(ha)$ is the visitation count of belief-action nodes and k is the exploration constant (Fig. 2b). The notation ha is the history h with

action a appended to the end, alias to h^- with action a explicitly seen. The function f is log in the case of Upper Confidence Bound (UCB) [19] exploration and power in the case of Polynomial Upper Confidence Tree (PUCT) [20]. The MCTS can be run with rollout and without. In the case of rollout configuration from each new belief node, the rollout is initiated to provide an initial $\hat{V}^*$ of the newly added belief node. This is not mandatory since if no rollout is initiated the MCTS will continue to descend down the tree until the deepest level with the first action from the action space $\mathcal{A}$ (first sampled action in case of continuous action space). **Not in every tree query** the MCTS will expand a new node. In some queries, only visitation counts are promoted (lace already present in the tree incorporated to pertinent $\hat{Q}$). In continuous spaces it happens because of DPW. DPW as well as increasing the visitation counts without adding a new lace introduces observations distribution shift. This is out of the scope of this paper. The $\hat{Q}(b(h),a)$ estimates are assembled from the laces (yellow curve in Fig. 1). Another name for lace is decision epoch or episode or script. Imagine that at the depth ℓ of the belief tree, each belief has a **global** index i_ℓ per depth ℓ, say index runs from left to right over all the belief nodes at level ℓ. Let us define the set of global indices of posterior beliefs which are children of $b_\ell^{i_\ell}(h_\ell^{i_\ell})$ and action a_ℓ by $C(h_\ell^{i_\ell}a_\ell)$. We also define the set of actions emanating from $b_\ell^{i_\ell}$ by $C(h_\ell^{i_\ell})$. Only in time zero we make these sets and visitation counts depend on belief instead of history. In the next equation, we omit the subscript denoting time instance of histories, beliefs, and actions. Suppose MCTS is configured to run without rollout. In this case $\hat{Q}(h_\ell^{i_\ell},a_\ell)$ reads

$$\hat{Q}(h^{i_\ell},a)=\overbrace{\sum_{i_{\ell+1}\in C(h^{i_\ell}a)}\frac{n(h^{\prime,i_{\ell+1}})}{n(h^{i_\ell}a)}\Big(\rho_{\ell+1}(b^{i_\ell},a,b^{\prime,i_{\ell+1}})+}^{\text{single immediate action}}$$
$$\underbrace{\overbrace{\sum_{a'\in C(h^{\prime,i_{\ell+1}})}\frac{n(h^{\prime,i_{\ell+1}}a')}{n(h^{\prime,i_{\ell+1}})}\hat{Q}(h^{\prime,i_{\ell+1}},a')}^{\text{different actions due to eq. (7) approximating the best exploratory future tree policy } \pi^*}\Big)}_{\hat{V}^{\pi^*}(h^{\prime,i_{\ell+1}})}. \tag{8}$$

The future policy highlighted by magenta color is tree query dependent (See Fig. 1). In the same manner, the sets $C(h^{i_\ell}a)$ and $C(h^{\prime,i_{\ell+1}})$ implicitly depend on the tree query number. One of our crucial insights in this paper is the summation over the actions in (8) marked by the red color. This average can also be perceived as a stochastic policy. In **finite** time this summation can include unsafe actions in an unconstrained MCTS approach.

III. Problem Formulation and Rationale

We now proceed to our theoretical problem formulation. To reduce clutter we assume that the planning time index is zero. This is not an inherent limitation of our approach, every further relation can be easily modified to accommodate general planning time index. We endow the BMDP described in Section II-B with belief-dependent operator ϕ and obtain

$$\langle \mathcal{B},\mathcal{A},\mathrm{T}_b,\underbrace{\rho}_{\substack{\text{belief}\\ \text{dependent}\\ \text{reward}}},\underbrace{\phi}_{\substack{\text{belief}\\ \text{dependent}\\ \text{payoff}}},\gamma,b_0\rangle.$$

A. Problem Formulation

Our aim is to tackle the problem presented in [3] and [4] narrowed to the multiplicative form of the inner constraint considering a stochastic future policy. In [3] and [4] we presented our Probabilistic Constraint (PC) defined as such $\mathrm{P}(c{=}1|b_0,a_0,\pi){=}1$ where c is a Bernoulli random variable. In this work c maps to one the event $\bigcap_{\ell=0}^{L}A_\ell^\delta$ such that the problem we want to solve is

$$a_0^*\in\arg\max_{a_0\in\mathcal{A}}Q^{\pi^*}(b_0,a_0;\boldsymbol{\rho}_1)\quad\text{subject to} \tag{9}$$
$$\underbrace{\mathrm{P}\big(\textstyle\bigcap_{\ell=0}^{L}A_\ell^\delta|b_0,a_0,\pi^*_{1:L-1}\big){=}1}_{\substack{\text{outer}\\ \text{constraint}}} \tag{10}$$

In this paper, we define the following sets as said $A_0^\delta\triangleq\{b_0{:}\phi(b_0){\geq}\delta\}$ and for $\ell{\in}[1{:}L]$ the relevant set appears as

$$A_\ell^\delta\triangleq\big\{b_\ell^-,b_\ell{:}b_\ell^-{\in}\mathcal{B}_\ell^-,b_\ell{\in}\mathcal{B}_\ell,\phi(b_\ell^-){\geq}\delta,\phi(b_\ell){\geq}\delta\big\}. \tag{11}$$

One example of an operator ϕ is the probability to be safe given belief, specified as:

$$\phi(b_\ell){=}\mathrm{P}\big(\{x_\ell{\in}\mathcal{X}_\ell^{\text{safe}}\}\big|b_\ell\big){=}\mathbb{E}_{x_\ell\sim b_\ell}[\mathbf{1}_{\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}}] \tag{12}$$
$$\phi(b_\ell^-){=}\mathrm{P}\big(\{x_\ell{\in}\mathcal{X}_\ell^{\text{safe}}\}\big|b_\ell^-\big){=}\mathrm{P}\big(\{x_\ell{\in}\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-\big). \tag{13}$$

Here, $\mathcal{X}^{\text{safe}}$ is the safe space, e.g. the space where a robot can move without inflicting damage on itself. Therefore, we can think about the event $\bigcap_{\ell=0}^{L}A_\ell^\delta$ as the **Safe Belief Space**.

The $\mathcal{B}_\ell^-$ and $\mathcal{B}_\ell$ in (11) are the reachable spaces in time ℓ of propagated beliefs b_ℓ^- and posteriors b_ℓ respectively. The reachable space in time ℓ is the space of all the beliefs in time ℓ that can be reached from a belief given in planning session, using the stochastic execution policy π and changing the actions and the observations in (1) and (2) accordingly. In our case, the belief given in planning session is b_0. By the green color in (11) we highlight that we constrain the propagated beliefs in addition to the posteriors.

The probability of the event $\bigcap_{\ell=0}^{L}A_\ell^\delta$ equals to the probability of the event $\big(\mathbf{1}_{A_0^\delta}(b_0)\prod_{\ell=1}^{L}\mathbf{1}_{A_\ell^\delta}(b_\ell^-,b_\ell)\big){=}1$. In this work, although we use Particle Filter (PF) as the belief update ψ we do not take into account the stochasticity of the belief update operator as opposed to [21],[22] and treat ψ operator as deterministic. Since it would significantly complicate the paper, we leave this aspect to the future work.

One can extract the propagated belief from the belief update ψ, namely $\psi(b,a,z'){\triangleq}\psi^{\text{post}}(\psi^{\text{prop}}(b,a),z')$. Therefore, to make the exposition clearer, from now on the indicator $\mathbf{1}_{A_\ell^\delta}(b_\ell)$ depends solely on the posterior b_ℓ and not both the posterior b_ℓ and the propagated belief b_ℓ^-. Note that in algorithms, for the sake of clarity, we make the indicators dependent on both beliefs, propagated and posterior.

The $\pi^*_{1:L-1}$ is the best future exploratory stochastic policy approximated by our probabilistically-constrained MCTS as we will further see. The approximation of the best future tree policy improves over time as proved by [20] for an unconstrained problem. In our problem, instead of the best future stochastic tree policy, we have the best future stochastic probabilistically-constrained policy. This is because our PC

is automatically enforced in future times due to its recursive nature, as we will see in Section IV. From the discussion above and indicator properties, (10) equals to

$$\mathrm{P}\big(\underbrace{\big(\mathbf{1}_{A_0^\delta}(b_0)\prod_{\ell=1}^{L}\mathbf{1}_{A_\ell^\delta}(b_\ell)\big)=1}_{\substack{\text{inner}\\ \text{constraint}}}|b_0,a_0,\pi\big)= \mathbb{E}^{\mathrm{T,O}}[\mathbf{1}_{A_0^\delta}(b_0)\prod_{\ell=1}^{L}\mathbf{1}_{A_\ell^\delta}(b_\ell)|b_0,a_0,\pi]. \tag{14}$$

The outer condition (10) coupled with inner condition outlined by (14) says that with probability one (almost surely) future propagated and posterior beliefs b^- and b, L steps ahead, will satisfy $\phi(b^-)\geq\delta$ and $\phi(b)\geq\delta$ correspondingly.

Constraining the propagated belief (13) means constraining on average (theoretical expectation) the posterior as discussed in the next section.

B. Implications of Constraining Propagated Belief

In this section we shed light on the question what does it mean to constrain the propagated beliefs alongside with posterior beliefs. To cancel the constraining of the propagated beliefs one must redefine the set A_ℓ^δ for every ℓ as follows

$$A_\ell^\delta \triangleq \Big\{\cancel{b_0^-}, b_\ell: \cancel{b_\ell^-\in\mathcal{B}_\ell^-}, b_\ell\in\mathcal{B}_\ell, \cancel{\phi(b_\ell^-)\geq\delta}, \phi(b_\ell)\geq\delta\Big\}.$$

Further in the paper all the developments are valid for both versions of the set A_ℓ^δ. The probability to be safe given a propagated belief equals to

$$\begin{gathered}\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|b_\ell^-\big)=\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-\big)=\\ \int_{z_\ell\in\mathcal{Z}}\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-,z_\ell\big)\mathbb{P}(z_\ell|h_\ell^-)\mathrm{d}z_\ell=\\ \mathbb{E}_{z_\ell}[\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-,z_\ell\big)|h_\ell^-]=\\ \mathbb{E}_{z_\ell}[\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|b_\ell\big)|b_\ell^-].\end{gathered} \tag{15}$$

The theoretical expectation in (15) is out of the reach. Yet we evaluate it using the propagated belief $b^-(h^-)$. Defining the set A_ℓ^δ as (11), with the propagated beliefs, allows to account for all the possible posterior beliefs in (15). Additionally, we know that $\forall\epsilon>0$

$$\begin{gathered}\lim_{|C(h_\ell^-)|\to\infty}\mathrm{P}\big(|\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-\big)-\\ \tfrac{1}{|C(h_\ell^-)|}\textstyle\sum_{z_\ell^l\in C(h_\ell^-)}\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-,z_\ell^l\big)|>\epsilon\big|h_\ell^-\big)=0.\end{gathered} \tag{16}$$

With a slight abuse of notation, $C(h_\ell^-)$ is now a list of the enumerated observations that are children of h_ℓ^-. Equation (16) means that for any arbitrary small error ϵ, the difference between (15) and its approximation by the children of h_ℓ^- tends to zero as the number of children of h_ℓ^- grows.

Theorem 1 (Necessary condition for entire observation space $\mathcal{Z}$ of children of h_ℓ^- to be safe): Fix $\delta\in[0,1]$ and assume that

$$\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-\big)\geq\delta. \tag{17}$$

Eq. (17) is a necessary condition for the entire observation space $\mathcal{Z}$ of children of h_ℓ^- to be safe. To rephrase that

$$\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-\big)<\delta \tag{18}$$

implies that $\exists b_\ell(h_\ell)$ a child of h_ℓ^- which is not safe, namely, $\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-,z_\ell\big)<\delta$.

See Appendix C for a detailed proof. We still need to check the children posteriors $\{z_\ell^l\}_{l=1}^{|C(h_\ell^-)|}$. This is because the condition (17) is only necessary and not sufficient. In other words, if for all the children $\forall z_\ell\in\mathcal{Z}$ of h_ℓ^-, it holds that $\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-,z_\ell\big)\geq\delta$ it has to be that $\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-\big)\geq\delta$. Since the condition is not sufficient we cannot say that $\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-\big)<\delta$ implies that $\forall b_\ell(h_\ell)$ that are children of h_ℓ^- it will hold that $\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell\big)<\delta$.

Note that if for every sampled observation $\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-,z_\ell^l\big)\geq\delta$, it implies that

$$\Big(\tfrac{1}{|C(h_\ell^-)|}\textstyle\sum_{l=1}^{|C(h_\ell^-)|}\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-,z_\ell^l\big)\Big)\geq\delta. \tag{19}$$

To conclude, by constraining the propagated belief, we constrain the theoretical expectation of the posteriors given h_ℓ^-, and by constraining each posterior we also constrain its sample approximation portrayed by Eq. (19). Without constraining the propagated belief, if the number of children of $b_\ell^-(h_\ell^-)$ is small, namely, $|C(h_\ell^-)|$ is small, we anticipate poor robot's safety in execution of the best action found by our planner (e.g. number of collisions). This is because constraining the propagated belief allows to account in expectation for all the observations in the observation space, and not only the sampled observations. This will happen if the number of MCTS tree queries is small.

It is possible that other definitions of safety of the beliefs can be utilized. While this is outside the scope of this paper, we specified relevant operators ϕ in the Appendix, Section D.

Remark: To assure feasibility of our PC (10) at the limit of MCTS convergence, the robot has to have a bounded support of the belief b_0 and bounded motion models. If we deal with a particle based representation of b_0 we perceive the particles as true robot positions, so it is left only to assure that the motion model is bounded. This is, however, natural since the robot cannot have limitless actuators.

IV. PC-MCTS (ANYTIME APPROACH)

Our constraint depends on a stochastic policy. Similar to the objective (5) in our PC we land at the following result.

Theorem 2 (Representation of PC, recursive form): The PC defined by (14) conforms to the following recursive form.

$$\begin{gathered}\mathbf{1}_{A_0^\delta}(b_0)\mathbb{E}_{b_1}\Big[\mathbf{1}_{A_1^\delta}\mathbb{E}_{a_1}\Big[\mathbb{E}_{b_2}\Big[\mathbf{1}_{A_2^\delta}\dots\\ \mathbb{E}\big[\mathbf{1}_{A_L^\delta}|b_{L-1},a_{L-1}\big]\dots\Big|b_1,a_1\Big]\Big|b_1,\pi_1,\Big|b_0,a_0\Big]=\\ \mathbf{1}_{A_0^\delta}(b_0)\mathbb{E}_{b_1}\Big[\mathbb{E}_{a_1\sim\mathbb{P}_1^\pi(a_1|b_1)}\Big[\\ \mathrm{P}\big(\big(\prod_{\ell=1}^{L}\mathbf{1}_{A_\ell^\delta}(b_\ell)\big)=1\Big|b_1,a_1,\pi\Big)\Big|b_1,\pi_1\Big]\Big|b_0,a_0\Big].\end{gathered} \tag{20}$$

We provide a detailed proof in the Appendix, Section B.

In this section, we present our anytime safety approach. To invalidate the sample approximation of (10) it is sufficient that a single belief (propagated or posterior) in the belief tree fails to be safe and the corresponding indicator is zero. In our methodology, we leverage the classical iterative MCTS scheme of descending down the search tree of histories and

ascending back to the root (Section II-C). Once on the way down the tree an unsafe belief is encountered, we know that the PC enforced from each predecessor belief node is violated. We delete such an action from the search tree and fix the $\hat{Q}$ above. Let us delve into the details.

Suppose the MCTS is configured to run without rollout. Would we construct the estimated counterpart of (20) from the belief tree constructed by MCTS our PC would be as such

$$\begin{aligned}&\Big(\mathbf{1}_{A_0^\delta}(b_0)\sum_{i_1\in C(b_0a_0)}\frac{\mathbf{1}_{A_1^\delta}(b_1^{i_1})}{|C(b_0a_0)|}\\&\sum_{a_1\in C(h_1^{i_1})}\frac{n(h_1^{i_1}a_1)}{n(h_1^{i_1})}\sum_{i_2\in C(h_1^{i_1}a_1)}\frac{\mathbf{1}_{A_2^\delta}(b_2^{i_2})}{|C(h_1^{i_1}a_1)|}\cdots\\&\cdots\frac{\mathbf{1}_{A_{L-1}^\delta}(b_{L-1}^{i_{L-1}})}{|C(h_{L-2}^{i_{L-2}}a_{L-2})|}\sum_{a_{L-1}\in C(h_{L-1}^{i_{L-1}})}\frac{n(h_{L-1}^{i_{L-1}}a_{L-1})}{n(h_{L-1}^{i_{L-1}})}\\&\sum_{i_L\in C(h_{L-1}^{i_{L-1}}a_{L-1})}\frac{\mathbf{1}_{A_L^\delta}(b_L^{i_L})}{|C(h_{L-1}^{i_{L-1}}a_{L-1})|}\Big)=1.\end{aligned}\tag{21}$$

Since our constraint is defined using an indicator, (21) translates to the fact that *each* lace defined by the actions and the observations on the way down the tree shall consist of safe beliefs.

To emphasize that each belief in the search tree has a single parent and the corresponding parent is attainable, let us introduce yet another notation $b_\ell^{i_\ell|i_{\ell-1}}$. This means that the belief $b_\ell^{i_\ell|i_{\ell-1}}$ has global index i_ℓ and parent belief has global index $i_{\ell-1}$. On the way down the tree we ensure that

$$\begin{aligned}&\Big(\mathbf{1}_{A_0^\delta}(b_0^1)\mathbf{1}_{A_1^\delta}(b_1^{i_1|1})\mathbf{1}_{A_2^\delta}(b_2^{i_2|i_1})\cdots\\&\mathbf{1}_{A_{L-1}^\delta}(b_{L-1}^{i_{L-1}|i_{L-2}})\mathbf{1}_{A_L^\delta}(b_L^{i_L|i_{L-1}})\Big)=1,\end{aligned}\tag{22}$$

where the actions along the lace are $a_1\in C(h_1^{i_1}), a_2\in C(h_2^{i_2}),\ldots,a_{L-1}\in C(h_{L-1}^{i_{L-1}})$ and the beliefs are according to the observations indexed by $i_2\in C(h_1^{i_1}a_1), i_3\in C(h_2^{i_2}a_2),\ldots,i_L\in C(h_{L-1}^{i_{L-1}}a_{L-1})$. In other words we require that every propagated and posterior belief along the lace would be safe.

Remark The equivalence of (21) and the fact that every lace in search tree shall be safe is a property of our PC formulation, e.g., this is no happening in case of popular Chance Constraint [15].

Note that in (21) we do not have distributional shift due to progressive widening of observations (and the fact that not in every tree query a new belief node is introduced) as opposed to the objective (8). This is because we do not take into account the statistics dictated by the visitation counts of the observations.

As we see from (21), the recursive form portrayed by (20) transfers to MCTS estimator. For clarity of the exposition let us denote the product of the indicators in the inner constraint (14) by c depending on the current and future beliefs. For example, at the root of the belief tree we have $c(b_{0:L})=\mathbf{1}_{A_0^\delta}(b_0)\prod_{\ell=1}^{L}\mathbf{1}_{A_\ell^\delta}(b_\ell)$. By design, (20) and (21) equals one if and only if, the PC starting from each belief action node ha in the tree is satisfied, namely $\mathrm{P}\big(c{=}1|b(h),a,\pi\big){=}1$. We now define the notion of dangerous action in belief tree.

Remark: We call an action dangerous if it is **believed** to be dangerous. Meaning our notion of dangerous or safe actions based on beliefs and not the possible POMDP states as in Chance Constraint [2].

Definition 1 (Dangerous action): A dangerous action is action a in a place h in a search tree that renders an estimator of (20) smaller than one, namely $\hat{\mathrm{P}}\big(c{=}1|b_0,a_0,\pi\big){<}1$, where the estimator is as in (21).

Note that best stochastic future tree policy is dependent on the number of performed tree queries.

Corollary 1: Each action in a search tree can be dangerous or safe. We define **safe** action a (or a_0) to be the action that is **not** dangerous, namely $\hat{\mathrm{P}}\big(c{=}1|b_0,a_0,\pi\big){=}1$ and safe under the safe future tree policy, namely $\hat{\mathrm{P}}\big(c{=}1|b(h),a,\pi\big){=}1$ for arbitrary future history h as a result of mentioned safe policy. Let us reiterate that we build the search tree solely from the safe actions. Effectively, using our pruning and fixing the values and statistics maintained by the search, to be explained shortly, we assure preemptively that the sample approximation of (20) defined by (21) using the beliefs from the search tree built by MCTS equals to one. To assure that the (21) equals one it is required that every indicator function within is one. This is our mechanism to assure that in **any** finite time the search tree contains only **safe** actions as opposed to duality based methods where the constraint is satisfied only at the convergence limit, namely in infinite time (see Section VI-B). When a newly sampled belief renders the corresponding indicator equal one, we add it to the belief tree. If the indicator is zero, we develop a mechanism to delete an action and fix the search tree upwards.

A. Pushing Forward in Time Only the Safe Trajectories

Even if (21) equals one, meaning every indicator inside equals one, when $\delta{<}1$ and payoff operator as in (12), it is possible that there exist samples that are unsafe, e.g., falling inside an obstacle or a dangerous region. If the robot is operational it means the robot was safe before it commenced an action. Thus, we shall discard the unsafe portion of the belief before we update the belief with action and observation (barring the situation when $\delta{=}1$ and payoff operator as in (12) and (13)). We define $\bar{b}^{\mathrm{safe}}$ as the belief constituted only by the safe particles, namely conditioned on the history and the events $\{x{\in}\mathcal{X}^{\mathrm{safe}}\}$. Such a belief is given by

$$\bar{b}_\ell^{\mathrm{safe}}(x_\ell){=}\mathbb{P}(x_\ell|b_0,a_{0:\ell-1},z_{1:\ell},\textstyle\bigcap_{i=0}^{\ell}\{x_i{\in}\mathcal{X}_i^{\mathrm{safe}}\}).\tag{23}$$

To convert $\bar{b}$ to $\bar{b}^{\mathrm{safe}}$ we remove not safe particles and resample with replacement the safe ones to the initial size. This means that the beliefs and observations in (20) will be not as in objective (9) but as follows. We define $\bar{b}$ as the belief obtained by percolating forward in time belief that has been made safe sequentially, that is $\bar{b}'{=}\psi(\bar{b}^{\mathrm{safe}},a,z')$ where

$$\bar{b}_\ell(x_\ell){=}\mathbb{P}(x_\ell|b_0,a_{0:\ell-1},z_{1:\ell},\textstyle\bigcap_{i=0}^{\ell-1}\{x_i{\in}\mathcal{X}_i^{\mathrm{safe}}\}),\tag{24}$$

and the belief propagated only with action and without an observation

$$\bar{b}_\ell^{-}(x_\ell){=}\mathbb{P}\big(x_\ell|b_0,a_{0:\ell-1},z_{1:\ell-1},\textstyle\bigcap_{i=0}^{\ell-1}\{x_i{\in}\mathcal{X}_i^{\mathrm{safe}}\}\big).\tag{25}$$

Both beliefs are generally unsafe. Our BMDP tuple is now augmented with another space of beliefs $\bar{\mathcal{B}}$ defined by (24). We have now

$$\langle \mathcal{B}, \underbrace{\bar{\mathcal{B}}}_{\substack{\text{space}\\ \text{of the beliefs}\\ \text{defined by (24)}}}, \mathcal{A}, \mathrm{T}_b, \underbrace{\rho}_{\text{reward}}, \underbrace{\phi}_{\text{payoff}}, \gamma, b_0 \rangle.$$

At this point we need to define another safe set

$$\bar{A}^{\delta}_{\ell}=\{\bar{b}^{-}_{\ell}, \bar{b}_{\ell}{:}\bar{b}^{-}_{\ell}\in\bar{\mathcal{B}}^{-}_{\ell}, \bar{b}_{\ell}\in\bar{\mathcal{B}}_{\ell}, \phi(\bar{b}^{-}_{\ell})\geq\delta, \phi(\bar{b}_{\ell})\geq\delta\}.$$

Here the $\bar{\mathcal{B}}^{-}_{\ell}$ and $\bar{\mathcal{B}}_{\ell}$ are reachable by changing observations in (24) and (25) spaces. Do note that only in time 0 the set $A^{\delta}_{0}=\bar{A}^{\delta}_{0}$. This is because in inference we know that robot is still operational. We always make safe the actual robot belief b_0. In planning, the belief is rendering the observation in the next time step (Fig. 4a). Thus in both CC and PC in time $\ell{+}1$, the observation PDF reads $\mathbb{P}(z_{\ell+1}|\bar{b}^{\text{safe}}_{\ell}, a_{\ell})$, whereas in objective we have $\mathbb{P}(z_{\ell+1}|b_{\ell}, a_{\ell})$. We sample from the latter and the normalized ratios of these likelihoods

$$w^{a_\ell}_{\ell+1} \propto {\mathbb{P}(z_{\ell+1}|\bar{b}^{\text{safe}}_{\ell}, a_{\ell})}\big/{\mathbb{P}(z_{\ell+1}|b_{\ell}, a_{\ell})} \tag{26}$$

are the weights in the equation (28). Using Importance Sampling in such a way, we construct a single belief tree. However, for the constraint calculation we use the $\bar{b}$ corresponding to the belief b, please see Fig. 4a. The Eq. (14) transforms into

$$\mathrm{P}\big(\mathbf{1}_{A^{\delta}_{0}}(b_0)\textstyle\prod_{\ell=1}^{L}\mathbf{1}_{\bar{A}^{\delta}_{\ell}}(\bar{b}_{\ell})\big){=}1|b_0, a_0, \pi\big). \tag{27}$$

Let us reiterate, on the way down the tree, we ensure that every belief along the lace lightens up its indicator. Similar to (21), we ensure that under the stochastic policy approximated by the MCTS, the PC is satisfied. We have that

$$\begin{aligned}\Big(\mathbf{1}_{\bar{A}^{\delta}_{0}}(b_0)\textstyle\sum_{i_1\in C(b_0 a_0)} w^{a_0,i_1}_{1}\mathbf{1}_{\bar{A}^{\delta}_{1}}(\bar{b}^{i_1}_{1})\sum_{a_1\in C(h^{i_1}_{1})}\frac{n(h^{i_1}_{1}a_1)}{n(h^{i_1}_{1})}\\ \textstyle\sum_{i_2\in C(h^{i_1}_{1}a_1)} w^{a_1,i_2}_{2}\mathbf{1}_{\bar{A}^{\delta}_{2}}(\bar{b}^{i_2}_{2})\cdots\\ \cdots\mathbf{1}_{\bar{A}^{\delta}_{L-1}}(\bar{b}^{i_{L-1}}_{L-1})\textstyle\sum_{a_{L-1}\in C(h^{i_{L-1}}_{L-1})}\frac{n(h^{i_{L-1}}_{L-1}a_{L-1})}{n(h^{i_{L-1}}_{L-1})}\\ \textstyle\sum_{i_L\in C(h^{i_{L-1}}_{L-1}a_{L-1})} w^{a_{L-1},i_L}_{L}\mathbf{1}_{\bar{A}^{\delta}_{L}}(\bar{b}^{i_L}_{L})\Big){=}1.\end{aligned} \tag{28}$$

Further in this paper we assume that the observation model $\mathrm{O}(z, x)$ has infinite support, to rephrase that $\{z{\in}\mathcal{Z}, x{\in}\mathcal{X}{:}\mathrm{O}(z, x){>}0\}{=}\mathcal{Z}{\times}\mathcal{X}$. This assumption ensures that there are no nullified weights in (28). Further, since the weights in (28) are normalized and all of them are nonzero, even a single weight missing because the inner constraint is violated, renders (28) smaller than one. This means that the constraint with respect to the root b_0 of the belief tree is not satisfied. Since the weights are selfnormalized per action, to verify that (28) equals to one we do not need to calculate weights at all. In fact, we never check the whole PC approximation. In contrast, as we already mentioned we only verify that each indicator equals to one on the way down the tree.

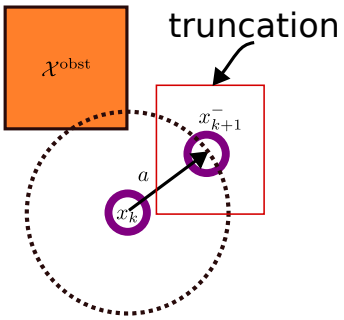


Fig. 3: Illustration of the effect of truncation of motion model T.

B. Bounded Support Motion Model

Suppose $\bar{b}^{\text{safe}}$ is represented by the finite set of particles, belief update ψ is a PF. If motion model has a support encapsulating the whole space $\mathbb{R}^d$, where d is a dimension, for every possible action it will be eventually unsafe belief. This is because eventually, at the limit of MCTS, for every tried action it will be non safe belief. However, we know that robot cannot teleport and truncation of motion model is, therefore, natural. In fact it is assumed often times to be Gaussian to bring infinite support to the table to alleviate the complexity of the solution. Observe Fig. 3. Without truncation for every action a it is possible that the propagated sample will be unsafe and render next in time posterior belief also unsafe.

Using our further presented method we build a tree solely from safe actions. Do note that all our algorithms can be run with various belief dependent operators. It is customary to maintain a pair of posterior beliefs $\bar{b}$ and b as visualized in Fig. 4a or just maintain a single belief b. Further we stick to the former scheme as in Fig. 4a.

C. Constraint Violation and Efficient Tree Cleaning

Before we begin this section we must clarify that from now on we slightly change the notations in text and the algorithms. To recap, we use following variables: h represents a history $\{b, a_0, z_1, \dots a_{\ell-1}, z_\ell\}$, and haz' is the shorthand for history with a and z' appended to the end. In a similar manner, as mentioned earlier, ha is the history h with action a appended to the end. $C(ha)$ is a set of the children of a belief-action node ha. Each such a child, now, is a triple of observation, reward, and posterior belief $\{z', r', b'\}$.

We now explain how we prune all dangerous actions (Def. 1) from the search tree and thereby our search tree always contains only the safe actions (Cor. 1). Actions at the root and tree future policy, which is stochastic due to exploration, are such that the PC is fulfilled starting from each belief node in the search tree. Suppose that our Probabilstically Constrained MCTS (Alg. 2 and 5) is currently at a belief node $b(h)$ in the belief tree, with a corresponding history h defining the unique place h in the belief tree. The algorithm selects an action according to (7) and suppose it creates a new belief. Every time we create a new belief node to be added to the search tree we obtain b' for the reward calculation and corresponding $\bar{b}'^{,-}$ and $\bar{b}'$ for the constraint. We then check if $\phi(\bar{b}'^{,-})\geq\delta$ and $\phi(\bar{b}')\geq\delta$ and if both inequalities are satisfied

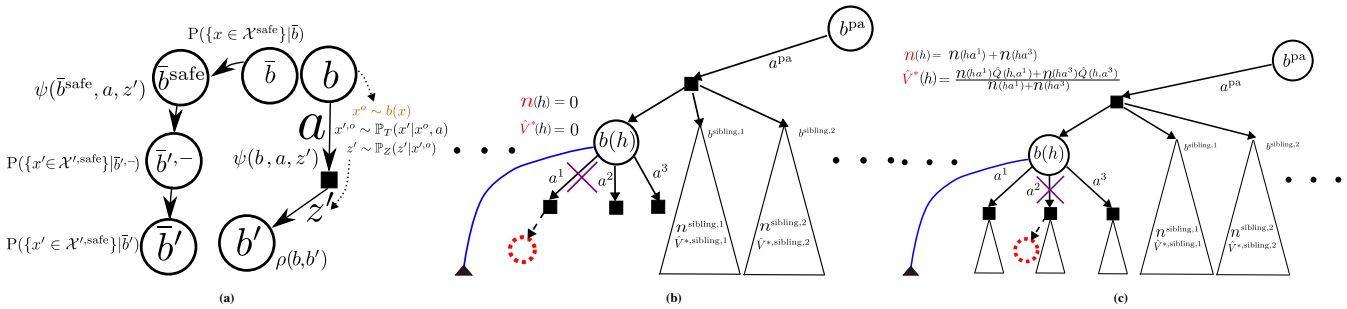


Fig. 4: **(a)** Conceptual illustration of maintaining a pair of posterior beliefs, b and b' are used for the reward operator, while $\bar{b}$, $\bar{b}'^{,-}$ and $\bar{b}'$ for the safety operator. We create observation using belief b highlighted by the brown color. In the subfigures **(b)** and **(c)** we visualize the cleaning of the belief tree in case that new belief violated the inner constraint. By blue color we mark elements related to *optional* rollout. By the thick red dashed circle we mark a newly expanded belief node. **(b)** In this scenario the belief node is a first child expanded from the first selected action a^1. The expanded belief violates the inner constraint. Thus we need to prune action a^1. Because we have not updated the visitation count of b yet, we only need to delete action a^1. **(c)** Harder scenario for cleaning the tree. Here we need to perform appropriate fixes to the action-value-estimates after we prune a^2. Note that UCB or PUCT, tries each action from each belief infinitely many times.

we add the newly created node to belief tree. If $\phi(\bar{b}'^{,-})<\delta$ or $\phi(\bar{b}')<\delta$, we shall prune an action leading to this belief node from the belief tree and fix the $\hat{Q}$ upwards since the laces emanating from the cleaned action participate in $\hat{Q}$ of every ancestor belief action node. Due to the fact that we assemble $\hat{Q}$ at each belief-action node from laces, at node h it holds that $\hat{V}^*(h)=\sum_{a\in C(h)}\frac{n(ha)\hat{Q}(ha)}{n(h)}$ and $n(h)=\sum_{a\in C(h)}n(ha)$. The $\hat{Q}$ of the parent reads

$$\hat{Q}(h^{\mathrm{pa}}a^{\mathrm{pa}})=\frac{(n(h)+1)\big(\rho(b^{\mathrm{pa}},a^{\mathrm{pa}},b)+\hat{V}^*(h)+\mathrm{roll}(h)\big)+\ldots}{n(h^{\mathrm{pa}}a^{\mathrm{pa}})}. \tag{29}$$

where the summation over all the sibling subtrees and the visitation count appears as $n(h^{\mathrm{pa}}a^{\mathrm{pa}})=n(h)+1+n^{\mathrm{sibling},1}+1\ldots$, where by the red red color we denote values of the current belief node and by the blue color we denote optional values related to the activation of the rollout. We now turn to an explanation of how to clean the tree efficiently using subtraction and adding operations instead of assembling action-value estimates and visitation counts from scratch using updated values down the tree. Suppose the action leading to the newly added belief does not have sibling subtrees corresponding to another actions and this belief is the first child of such an action, as depicted in Fig. 4b. In this case the visitation count $n(h)$ and $\hat{V}^*(h)$ are not present yet within $\hat{Q}(h^{\mathrm{pa}}a^{\mathrm{pa}})$. This is because the only rollout was commenced from $b(h)$. We can just delete the action leading to the newly created belief node.

We, now, focus on a more interesting setting depicted in Fig. 4c. After we deleted the subtree defined by the belief node $b(h)$ and action a (a^2 in Fig. 4c), we need to update the visitation count $n(h)$ as such $n(h)\leftarrow n^{\mathrm{intree}}(h)-n^{\mathrm{intree}}(ha)$, where we denote values that are currently in the belief tree by $\square^{\mathrm{intree}}$. As shown in Fig. 4c, we have that $\hat{V}^{*,\mathrm{intree}}(h)=\frac{\sum_{a\in C^{\mathrm{intree}}(h)}n^{\mathrm{intree}}(ha)\hat{Q}^{\mathrm{intree}}(ha)}{\sum_{a\in C^{\mathrm{intree}}(h)}n^{\mathrm{intree}}(ha)}$. At this point we have everything to calculate the updated value function at belief $b(h)$ indexed by history h. To do that we use the relation:

$$\underbrace{\hat{V}^*(h)\cdot n(h)}_{S(h)}\leftarrow\underbrace{\hat{V}^{*,\mathrm{intree}}(h)n^{\mathrm{intree}}(h)}_{S^{\mathrm{intree}}(h)}- \hat{Q}^{\mathrm{intree}}(ha)n^{\mathrm{intree}}(ha). \tag{30}$$

For an efficient update by (30) we need to cache the sum of the cumulative reward laces for each belief node. We define such a sum as $S(h)\triangleq\hat{V}^*(h)\cdot n(h)$. In addition, we need to subtract the deleted action a from the set of children of $b(h)$, namely $C(h)\leftarrow C^{\mathrm{intree}}(h)\backslash\{a\}$. If $b(h)$ is a root node we just update its visitation count $n(h)$ to a new value. We do not need to store $S(h)$ for a root belief. Else, we identify a parent action and node of $b(h)$ marked a^{pa} and b^{pa} respectively. We need to calculate $n(h^{\mathrm{pa}}a^{\mathrm{pa}})$ as such, $n(h^{\mathrm{pa}}a^{\mathrm{pa}})\leftarrow n^{\mathrm{intree}}(h^{\mathrm{pa}}a^{\mathrm{pa}})-n^{\mathrm{intree}}(h)+n(h)$. We then shall update $\hat{Q}(h^{\mathrm{pa}}a^{\mathrm{pa}})$ as such

$$\begin{aligned}\hat{Q}(h^{\mathrm{pa}}a^{\mathrm{pa}})\cdot n(h^{\mathrm{pa}}a^{\mathrm{pa}})\leftarrow\hat{Q}^{\mathrm{intree}}(h^{\mathrm{pa}}a^{\mathrm{pa}})\cdot n^{\mathrm{intree}}(h^{\mathrm{pa}}a^{\mathrm{pa}})-\\ \big(n^{\mathrm{intree}}(h)+1\big)\big(\rho(b^{\mathrm{pa}}a^{\mathrm{pa}}b)+\hat{V}^{*,\mathrm{intree}}(h)+\mathrm{roll}^{\mathrm{intree}}(h)\big)+\\ \big(n(b)+1\big)\big(\rho(b^{\mathrm{pa}}a^{\mathrm{pa}}b)+\hat{V}^*(h)+\mathrm{roll}^{\mathrm{intree}}(h)\big).\end{aligned} \tag{31}$$

Note that (31) encompasses both cases. The case when the subtree is deleted (30) and the case then only the visitation counts down the tree and the $\hat{Q}$ are updated (upper levels of the belief search tree). The value of $\hat{V}^*(h)$ subsumes both cases. In the latter case its update reads as such

$$\underbrace{\hat{V}^*(h)\cdot n(h)}_{S(h)}\leftarrow\underbrace{\hat{V}^{*,\mathrm{intree}}(h)n^{\mathrm{intree}}(h)}_{S^{\mathrm{intree}}(h)}- \hat{Q}^{\mathrm{intree}}(ha)n^{\mathrm{intree}}(ha)+\hat{Q}(ha)n(ha). \tag{32}$$

We now calculate a new visitation count of b^{pa} using $n(h^{\mathrm{pa}})\leftarrow n^{\mathrm{intree}}(h^{\mathrm{pa}})-n^{\mathrm{intree}}(h^{\mathrm{pa}}a^{\mathrm{pa}})+n(h^{\mathrm{pa}}a^{\mathrm{pa}})$, and $S(h^{\mathrm{pa}})$ using (32) and renaming there the history from h to h^{pa} and the action from a to a^{pa}. Now we can treat $b^{\mathrm{pa}}(h^{\mathrm{pa}})$ similarly as we treated $b(h)$. We outlined the tree cleaning procedure in Alg. 4. To conclude, this way our search tree always consists of only safe actions (not dangerous with respect to Def. 1).

V. The Algorithms and Guarantees

This section describes our algorithms followed by convergence guarantees. Alg. 2 summarizes our main result. Similar to [18], we present a provable modified variant summarized by Alg. 5.

Listing 1 Common procedures

```
1: procedure PLAN(belief: b, horizon: L)
2:     for m iterations or timeout do
3:         h ← ∅
4:         SIMULATE(b, b, h, L)        ▷ A single tree query
5:     end for
6:     return ACTIONSELECTION(b, h, 0)
7: end procedure
```

A. Detailed Algorithms Description

The entry point of both these algorithms, listed in Listing 1, is a loop over trials of observation laces. The difference between the algorithms is in the SIMULATE function. We name a single call to SIMULATE a **tree query**. In each trial, we descend with the lace of observations and actions intermittently, calculate the beliefs and rewards along the way, and ascend back to the root of the belief tree. Once, on the way down the tree, the unsafe belief is encountered we clean such action from the search tree and fix the action value estimates of all ancestor belief action nodes. Similar to the classical MCTS our approach can be run with rollout or without. In addition, if we do not want to use safe beliefs for the constraint we only need to remove parts marked by the brown color in Alg. 2 and 5 and use regular belief instead. We also present a Polynomial variant of our approach, Alg. 5 we named PC-MCTS with Polynomial Upper Confidence Tree (PC-MCTS-PUCT). In the next section, we prove the convergence with an exponential rate of Alg. 5 in probability. Note that we cache the values of the summation of the cumulative reward for all belief nodes for both algorithms. This happens in line 31 of Alg. 2 and line 30 of Alg. 5, where we denote $S(h) = \hat{V}^*(h) \cdot n(h)$.

a) Safe Rollout: The rollout is not necessary for applying MCTS. Without rollout, upon opening a new belief node the MCTS would behave as it reached the leaf node. Nevertheless, the rollout helps to provide better results in finite time and apparently helps to accelerate convergence (We did not find any rigorous analysis for that). With this motivation in mind, we present the Safe Rollout routine for our approach summarized by Alg. 3. Our safe rollout selects action randomly which myopically fulfills the sample approximation of myopic PC based on m samples. If no feasible action exists (which is not possible with our method since we always have an action "do not do anything") we select an action maximizing the sample approximation mentioned before.

B. Convergence Guarantees

Although we sample actions and observations and the number of samples of both is marching to infinity in discrete steps, MCTS converges, at each belief action node in probability, to the optimal value of the probailistically constrained problem defined by

$$Q^{\pi^*}(b(h), a; \boldsymbol{\rho}) \quad \text{subject to} \tag{33}$$

$$\mathrm{P}\big(c{=}1 \big| \bar{b}(h), a, \pi\big){=}1. \tag{34}$$

In (33) we omitted the time indices. MCTS approximates the stochastic policy π^* by a discrete but infinite set of sampled

Algorithm 2 Probabilistically Constrained MCTS (PCMCTS)

```
1: procedure SIMULATE(belief: b, belief: b̄, history: h, depth: d)
2:     if d = 0 then
3:         return 0
4:     end if
5:     b̄^safe ← MAKEBELIEFSAFE(b̄)
6:     SafeActionFlag ← false
7:     while not(SafeActionFlag) do
8:         a ← ACTIONSELECTION(h, c)
9:         Calculate propagated belief b'^− from b and b̄'^,− from b̄^safe using a
10:        if |C(ha)| ≤ k_o n(ha)^α_o then ▷ observation Prog. Widening
11:            z' ∼ P_O(z|x^o); x^o ∼ b'^−
12:            b̄' ← ψ(b̄^safe, a, z'),          Calculate 1_{{b̄'^,−, b̄': φ(b̄'^,−)≥δ, φ(b̄')≥δ}}(b̄'^,−, b̄')
13:            if 1_{{φ(b̄'^,−)≥δ, φ(b̄')≥δ}}(b̄'^,−, b̄') == 0 then
14:                CLEANTREE(h, a) ▷ Clean current belief tree to be safe using Alg. 4.
15:                Continue ▷ Jump to line 14
16:            else
17:                SafeActionFlag ← true
18:            end if
19:            b' ← ψ(b, a, z'), r' ← ρ(b, a, z', b') ▷ Regular belief and the reward on top of it are obtained only for not pruned actions
20:            C(ha) ← C(ha) ∪ {z', r', b'}
21:            r^lace ← r' + γ SAFEROLLOUT(b', d − 1)
22:        else
23:            SafeActionFlag ← true
24:            {z', r', b'} ← sample uniformly from C(ha)
25:            r^lace ← r'+γ SIMULATE(b',b̄', haz', d−1)
26:        end if
27:    end while
28:    n(h) ← n(h) + 1 ▷ Initialized to zero
29:    n(ha) ← n(ha) + 1 ▷ Initialized to zero
30:    Q̂(ha) ← Q̂(ha) + (r^lace − Q̂(ha))/n(ha) ▷ Initialized to zero
31:    S(ha) ← S(ha) + r^lace ▷ Initialized to zero
32:    return r^lace
33: end procedure
34: procedure ACTIONSELECTION(b, h, c)
35:    if |C(h)| ≤ k_a n(h)^α_a then ▷ action Prog. Widening
36:        a ← NEXTACTION(h)
37:        C(h) ← C(h) ∪ {a}
38:    end if
39:    return arg max_{a∈C(h)} Q̂(ha) + c√(log n(h)/n(ha)) ▷ UCB
40: end procedure
```

Algorithm 3 Myopically Safe Rollout Action selection

procedure SAFEROLLOUTPOLICY(b, $\mathcal{A}$)
  $\mathcal{A} \leftarrow$ shuffle($\mathcal{A}$). $V^* \leftarrow -\infty$ ▷ by shuffle assure that action is selected randomly
  for $a \in \mathcal{A}$ **do**
    for m iterations **do**
      Calculate propagated belief b'^- from b and a
      $z' \sim \mathbb{P}_{\mathrm{O}}(z|x^o); x^o \sim b'^-$
      $b' \leftarrow \psi(b, a, z')$
      Calculate $\mathbf{1}_{\{\phi(b'^-)\geq\delta, \phi(b')\geq\delta\}}(b'^-, b')$
    end for
    $\hat{V}^{(m)} \leftarrow \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{\{\phi(b'^-)\geq\delta, \phi(b')\geq\delta\}}(b'^{,i,-}, b'^{,i})$
    if $\hat{V}^{(m)} \geq 1-\epsilon$ **then** ▷ Note that we added ϵ here
      return a ▷ First shuffled action satisfying myopic PC approx. is returned
    else if $\hat{V}^{(m)} > \hat{V}_*^{(m)}$ **then**
      $V^* \leftarrow V$, $a^* \leftarrow a$
    end if
  end for
  return a^*
end procedure

Algorithm 4 Cleaning Belief tree to be safe

procedure CLEANTREE(h, a)
  Delete all children b' of ba belief-action node and delete ba itself $C(h) \leftarrow C^{\mathrm{intree}}(h) \setminus \{a\}$
  if $n^{\mathrm{intree}}(h) == 0$ **then**
    return
  end if
  $n(h) \leftarrow n^{\mathrm{intree}}(h) - n^{\mathrm{intree}}(ha)$
  if b is root **then**
    $C^{\mathrm{intree}}(h) \leftarrow C(h)$, $n^{\mathrm{intree}}(h) \leftarrow n(h)$,
    return
  else
    Assemble $\hat{V}^*(h)$
  end if
  while true **do**
    if b is root **then**
      $n^{\mathrm{intree}}(h) \leftarrow n(h)$,
      return
    end if
    Identify a^{pa} which is parent to b
    Identify b^{pa} such that $b^{\mathrm{pa}}a^{\mathrm{pa}}$ is a belief action node which is parent of b
    $n(h^{\mathrm{pa}}a^{\mathrm{pa}}) \leftarrow n^{\mathrm{intree}}(h^{\mathrm{pa}}a^{\mathrm{pa}}) - n^{\mathrm{intree}}(h) + n(h)$ ▷ History h^{pa} corresponds to belief b^{pa}
    Reconstruct $\hat{Q}(h^{\mathrm{pa}}a^{\mathrm{pa}})$ and put $n^{\mathrm{intree}}(h) \leftarrow n(h)$ and $S^{\mathrm{intree}}(h) \leftarrow S(h)$
    $n(h^{\mathrm{pa}}) \leftarrow n^{\mathrm{intree}}(h^{\mathrm{pa}}) - n^{\mathrm{intree}}(h^{\mathrm{pa}}a^{\mathrm{pa}}) + n(h^{\mathrm{pa}}a^{\mathrm{pa}})$,
    Assemble $\hat{V}^*(h^{\mathrm{pa}})$ and put $\hat{Q}^{\mathrm{intree}}(h^{\mathrm{pa}}a^{\mathrm{pa}}) \leftarrow \hat{Q}(h^{\mathrm{pa}}a^{\mathrm{pa}})$ ▷ We have $V^*(h^{\mathrm{pa}})$ and $n(h^{\mathrm{pa}})$ for the next iteration
    $b \leftarrow b^{\mathrm{pa}}$
  end while
end procedure

continuous actions and statistics defined by the visitation counts. To give an intuition, the convergence in probability is a result of the fact that we are dealing with expectations (See Lemma 1 and Theorem 2) and the fact that every belief action node is visited infinite amount of times due to (7) even in continuous domains where the new nodes are endlessly expanded. Moreover, the polynomial variant of DPW used in Alg. 5 allows to each belief-action node be visited a sufficient amount of times before the next new belief is introduced. Thus, convergence in probability happens with an exponential rate. Although the tree policy is tree query dependent and improves over time, [20] showed the convergence is from the leafs and upwards to the value under the optimal stochastic policy. Further we show that if the actions are continuous and have some natural distribution, MCTS will eventually sample an unsafe action (line 36 in Alg. 2 and line 35 in Alg. 5).

Theorem 3: Suppose, at the node h, we have an action a sampler on top of the continuous probability space $(\mathcal{A}, \mathcal{F}, \mathrm{P})$ where $\mathcal{A}$ is the outcomes space, $\mathcal{F}$ events space and P is the probability. At the limit of convergence in infinite time (after an infinite number of tree queries), it holds that $C(h)$ includes an action sampled from $\mathcal{A}$ that is arbitrary close to the optimal action with respect to the theoretical action-value function at each belief node.

To prove that we take an arbitrary set $S \in \mathcal{A}$ such that $\mathrm{P}(S|h) > 0$. Note that the probability here depends on the history (belief) we focus on. Time marches to infinity in countable steps so as the samples are countable. Denote by $|C(h)|$ the number of $i.i.d.$ samples (line 36 in Alg. 2 and line 35 in Alg. 5). The probability of sampled action $a \sim \mathbb{P}(a|h)$ not to be in S is $\mathrm{P}(\{a \in \mathcal{A} : a \notin S\}|h) = 1 - \mathrm{P}(S|h)$. When the number of samples tends to infinity we have that

$$(1-\mathrm{P}(S|h))^{|C(h)|} \underbrace{\rightarrow}_{|C(h)|\rightarrow\infty} 0. \tag{35}$$

It holds that the probability not to sample an action in S tends to zero with number of samples tending to infinity. Therefore, in the unconstrained MCTS approach, the action value function estimates $\hat{Q}$ will include unsafe actions if they are exist in the action space $\mathcal{A}$. On the contrary in our safe approach we remove the actions sampled from the unsafe sets of arbitrary small positive measure.

We now show that the cleaning tree routine is necessary due to fact that we will sample an infinite amount of unsafe actions and this can shift the expectations with respect to actions in values maintained in search tree.

Without our pruning we will obtain infinitely many unsafe actions in each unsafe set S. It can be seen using the second Borel Cantelli Lemma. Towards this end let us define the event $E^i \triangleq \{a^i \sim \mathbb{P}(a|h) : a^i \in S\}$, sampled action i is a member of set S. The events E^i are independent since we sample actions independently. The series $\sum_{i=1}^{\infty} \mathrm{P}(E^i|h)$ are divergent since $\mathrm{P}(E^i|h) = \mathrm{P}(S|h) > 0$ by definition. Thus, $\mathrm{P}\big(\bigcap_{j=1}^{\infty} \bigcup_{i=j}^{\infty} E^i \big| h\big) = 1$, namely the event sampled action is

a member of set S occurs infinitely often times. To rephrase that, without our pruning we would have sampled infinitely many dangerous actions and, therefore, the expectations can undergo a shift and even if at the root of the belief tree we select an optimal safe action, the influence of unsafe future actions can be substantial.

In this section, we prove convergence in probability with an exponential rate of Polynomial Upper Confidence Tree (PUCT) version of our approach (Alg. 5). We now list down the changes between Alg. 2 and Alg. 5:

- In Alg. 5 we have Polynomial Double Progressive Widening with depth dependent parameters defined in [20];
- The rollout in Alg. 5 is missing;
- If Alg. 2 decided not to open a new branch it samples the triple $\{z', r', b'\}$ uniformly from $C(ha)$ (line 24 highlighted by the blue color). In contrast Alg. 5 selects the child with a minimal visitation count (line 23 highlighted by the blue color).

The following theorem provides its soundness.

Theorem 4 (Convergence with Exponential Rate in Probability): Every belief h and belief action node ha of Alg. 5, equipped with our pruning mechanism from Section IV-C and summarized by Alg. 4 converges in probability and with an exponential convergence rate to the optimal value function $V^*(b(h))$ and action-value function $Q(b(h)a)$, respectively, while satisfying the PC starting from the belief action node ha, namely $\mathrm{P}\big(c{=}1|\bar{b}(h), a, \pi^*\big){=}1$.

Next, we provide the proof under rather mild assumptions. To be specific we must assume that reward lies in a bounded interval and that sampling of actions covers the entire space with an arbitrary precision. For more precise definition see Def. 2. Our proof is valid for both approaches, namely with making belief safe before pushing forward in time with action and observation and without (in this case the constraint at each belief-action node is $\mathrm{P}\big(c{=}1|b(h), a, \pi^*\big){=}1$). Similar to [18] we leverage the proof by [20].

Before we proceed let us mention that DPW of Alg. 2 with $k_z = k_a = 1$ and depth dependent α_d as described in [20] are identical to the one used in Alg. 5.

Lemma 2: Fix belief node $b(h)$ in belief tree and belief action node ha, $k_a = k_o = 1$ in Alg. 2 and select in Alg. 2 and Alg. 5 same $\alpha_{o,d}$ and $\alpha_{a,d} \in (0, 1)$ in both algorithms (can be depth dependent). The condition $|C(ha)| \leq n(ha)^{\alpha_{o,d}}$ is equivalent to $\lfloor n(ha)^{\alpha_{o,d}} \rfloor > \lfloor (n(ha) - 1)^{\alpha_{o,d}} \rfloor$. In a similar manner $|C(h)| \leq n(h)^{\alpha_{a,d}}$ is equivalent to $\lfloor n(h)^{\alpha_{a,d}} \rfloor > \lfloor (n(h) - 1)^{\alpha_{a,d}} \rfloor$.

We provide sketch of the proof. It is sufficient to prove that the first claim is identical since both have identical structure. Let us focus on $|C(ha)| \leq n(ha)^{\alpha_{o,d}}$. The new child is added if and only if the visitation $n(ha)^{\alpha_{o,d}}$ passes the subsequent integer at some visitation of node ha. This is happening if and only if $\lfloor n(ha)^{\alpha_{o,d}} \rfloor > \lfloor (n(ha) - 1)^{\alpha_{o,d}} \rfloor$. ■

Now we would like to pay attention to the fact that since we clean the belief tree from the unsafe actions, we have $n(h) \geq \sum_{a \in C(h)} n(h, a)$. This is in contrast to the classical MCTS, where $n(h) = \sum_{a \in C(h)} n(h, a)$. Therefore, we shall fix the visitation count of each belief node and belief action node that have been affected by pruning. This is done by Alg. 4.

Algorithm 5 Probabilistically Constrained MCTS PUCT

1: **procedure** SIMULATE(belief: b, belief: $\bar{b}$, history: h, depth: d)
2: **if** $d = 0$ **then**
3: **return** 0
4: **end if**
5: $\bar{b}^{\text{safe}} \leftarrow$ MAKEBELIEFSAFE($\bar{b}$)
6: SafeActionFlag $\leftarrow$ false
7: **while** not(SafeActionFlag) **do**
8: $a \leftarrow$ ACTIONSELECTION(h, c)
9: Calculate propagated belief b'^{-} from b and $\bar{b}'^{,-}$ from $\bar{b}^{\text{safe}}$ using a
10: **if** $\lfloor n(ha)^{\alpha_{o,d}} \rfloor > \lfloor (n(ha) - 1)^{\alpha_{o,d}} \rfloor$ **then** ▷ observation Prog. Widening
11: $z' \sim \mathbb{P}_{\mathrm{O}}(z|x^o); x^o \sim b'^{-}$
12: $\bar{b}' \leftarrow \psi(\bar{b}^{\text{safe}}, a, z')$, Calculate $\mathbf{1}_{\{\bar{b}'^{,-}, \bar{b}': \phi(\bar{b}'^{,-}) \geq \delta, \phi(\bar{b}') \geq \delta\}}(\bar{b}'^{,-}, \bar{b}')$
13: **if** $\mathbf{1}_{\{\phi(\bar{b}'^{,-}) \geq \delta, \phi(\bar{b}') \geq \delta\}}(\bar{b}'^{,-}, \bar{b}') == 0$ **then**
14: CLEANTREE(h, a) ▷ Clean current belief tree to be safe using Alg. 4.
15: **Continue** ▷ Jump to line 14
16: **else**
17: SafeActionFlag $\leftarrow$ true
18: **end if**
19: $b' \leftarrow \psi(b, a, z')$, $r' \leftarrow \rho(b, a, z', b')$ ▷ Regular belief and reward on top of it are obtained only for not pruned actions
20: $C(ha) \leftarrow C(ha) \cup \{z', r', b'\}$
21: **else**
22: SafeActionFlag $\leftarrow$ true
23: $\{z', r', b'\} \leftarrow \underset{\{z', r', b'\} \in C(ha)}{\arg\min} \frac{n(hao)}{n(h)}$
24: **end if**
25: **end while**
26: $r^{\text{lace}} \leftarrow r' + \gamma$ SIMULATE(b', $\bar{b}'$, haz',$d{-}1$)
27: $n(h) \leftarrow n(h) + 1$ ▷ Initialized to zero
28: $n(ha) \leftarrow n(ha) + 1$ ▷ Initialized to zero
29: $\hat{Q}(ha) \leftarrow \hat{Q}(ha) + \frac{r^{\text{lace}} - \hat{Q}(ha)}{n(ha)}$ ▷ Initialized to zero
30: $S(ha) \leftarrow S(ha) + r^{\text{lace}}$ ▷ Initialized to zero
31: **return** r^{lace}
32: **end procedure**
33: **procedure** ACTIONSELECTION(b, h, c)
34: **if** $\lfloor n(h)^{\alpha_{a,d}} \rfloor > \lfloor (n(h) - 1)^{\alpha_{a,d}} \rfloor$ **then** ▷ action Prog. Widening
35: $a \leftarrow$ NEXTACTION(h)
36: $C(h) \leftarrow C(h) \cup \{a\}$
37: **end if**
38: return $\arg\max_{a \in C(h)} \hat{Q}(ha) + \sqrt{n(h)^{c_d}/n(ha)}$ ▷ PUCT
39: **end procedure**

The following claim is required to understand [20] and we give now an informal proof missing in [20].

Lemma 3: The k^{th} child of node ha in Alg. 5 is added on visit $n(ha) = \lceil k^{\frac{1}{\alpha}} \rceil \triangleq n_k(ha)$.

Observe that the left hand side of $\lfloor n(ha)^{\alpha_{o,d}} \rfloor > \lfloor (n(ha) - 1)^{\alpha_{o,d}} \rfloor$ jumped $\lfloor n(ha)^{\alpha_{o,d}} \rfloor$ times and right hand side

lagged exactly by single visitation. So the inequality is fulfilled exactly $\lfloor n(ha)^{\alpha_{o,d}} \rfloor$ times. Moreover, the *first* $n(ha)$ such that $n(ha)^{\alpha_{o,d}}$ passes a subsequent integer assures the jump. Meaning, we have two cases. The first case is $n(ha)^{\alpha_{o,d}}{=}\lfloor n(ha)^{\alpha_{o,d}} \rfloor{=}k$ and taking $k^{\frac{1}{\alpha_{o,d}}} = \lceil k^{\frac{1}{\alpha_{o,d}}} \rceil$ is returning us back to $n(ha)$. The second case is $\lfloor n(ha)^{\alpha_{o,d}} \rfloor + 1 > n(ha)^{\alpha_{o,d}} > \lfloor n(ha)^{\alpha_{o,d}} \rfloor = k$. But we know that if $k^{\frac{1}{\alpha}}$ would be an integer, it would the previous case with a smaller $n(ha)$. The $k^{\frac{1}{\alpha}}$ has to be slightly larger than integer so as ceil operator return the right natural $n(ha)$. Similarly number of actions expanded from node h is $\lfloor n(h)^{\alpha} \rfloor$. ■

Our cleaning routine prunes only the actions and fixes the affected visitation counts so the proof by [20] is not broken. Further we establish the definitions and the assumptions from [20] in order to assure the validity of the proof. Some of them we take directly from [20], [18].

Definition 2 (Regularity Hypothesis): The Regularity hypothesis is the assumption that for any $\Delta{>}0$, there is a non zero probability to sample an action that is optimal with precision Δ. More precisely, there is a $\theta > 0$ and a $p > 1$ (which remain the same during the whole simulation) such that for all $\Delta > 0$,

$$Q(ha){\geq}V^*(h){-}\Delta \text{ with probability of at least } \min(1, \theta\Delta^P). \tag{36}$$

Definition 3 (Exponentially sure in n): We say that some property depending on an integer n is exponentially sure in n if there exist positive constants C, h, and η such that the probability that the property holds is at least

$$1 - C\exp(-hn^{\eta}). \tag{37}$$

In addition we need to assume that the belief dependent reward is bounded from below and above, namely it lies in the closed interval $[\rho^{\min}, \rho^{\max}]$. Instead of $\rho : \mathcal{B} \times \mathcal{A} \times \mathcal{Z} \times \mathcal{B} \mapsto \mathbb{R}$ we require the mapping to be $\rho : \mathcal{B} \times \mathcal{A} \times \mathcal{Z} \times \mathcal{B} \mapsto [\rho^{\min}, \rho^{\max}]$ Under these assumptions the convergence result of Alg. 5 summarized by Theorem 4 holds.

VI. SOTA Continuous Constrained MCTS

We now firm up the loose ends and turn to the description of the existing constrained POMDP considered in an anytime setting which will serve as our baseline.

A. Expectation Constrained Belief-dependent POMDPs

The averaged constraint formulated with **payoff** operator and including the propagated beliefs would be

$$\mathbb{E}^{\mathrm{T,O}}\big[\textstyle\sum_{\ell=0}^{L} \phi(b_\ell^-, b_\ell)\big|b_0, \pi\big] \geq \delta. \tag{38}$$

One possibility is to define $\phi(b_\ell^-, b_\ell){=}\phi(b_\ell^-){+}\phi(b_\ell)$. Clearly the cumulative averaged formulation (38) is not suitable for safety since it permits deviations of the individual safety operators ϕ. It can happen that with the low probability of future observation the resulting posterior belief will be **extremely** unsafe. However, sometimes the operator ϕ is naturally bounded from above. It holds that $\mathrm{P}(\{x_\ell{\in}\mathcal{X}_\ell^{\text{safe}}\}|\square){\leq}1$. Thus, if we select an operator ϕ as in (12) and $\delta{=}2L$ it is sufficient to ensure safety. If $\delta{<}2L$, it permits deviations of the individual belief dependent operators. Therefore, the averaged with respect to observations episodes and stochastic policy cumulative constraint is not sufficient to assure safety. The works [7], [6] impose the averaged cumulative constraint at the root of the belief tree as

$$V^{\pi}(b_0;\boldsymbol{\theta}_0){\triangleq}\mathbb{E}^{\mathrm{T,O}}\big[\textstyle\sum_{\ell=0}^{L} \theta(b_\ell^-, b_\ell)\big|b_0, \pi\big] \leq \delta^{\theta}. \tag{39}$$

We introduced the optional dependence on b^- of cost operator θ (emphasized by turquoise color), e.g

$$\theta(b_\ell^-, b_\ell){=}\theta(b_\ell^-){+}\theta(b_\ell) \tag{40}$$

where

$$\theta(\square){=}1{-}\overbrace{\mathrm{P}\big(\{x_\ell \in \mathcal{X}_\ell^{\text{safe}}\}\big|\square\big)}^{\phi(\square) \text{ from (12)}} = \mathrm{P}\big(\{x_\ell \notin \mathcal{X}_\ell^{\text{safe}}\}\big|\square\big){=}\mathbb{E}_{x_\ell\sim\square}[\mathbf{1}_{\{x_\ell\notin\mathcal{X}_\ell^{\text{safe}}\}}] \tag{41}$$

with values in $\square$ are substituted by b_ℓ^- and b_ℓ respectively. Similar to the behavior of bounded payoff operator, here we can assure safety if $\delta^{\theta}{=}0$. This will assure that (39) is satisfied if and only if **all** $\theta(b_\ell^-, b_\ell)$ inside are zero. This is because $\mathrm{P}\big(\{x_\ell{\notin}\mathcal{X}_\ell^{\text{safe}}\}\big|\square\big){\geq}0$. In the light of the discussion about deviation of the cost values, further in this paper we assume that $\delta^{\theta}{=}0$. Now, if we set $\delta{=}1{-}\delta^{\theta}{=}1$ in our PC (10) and payoff as in (12) two formulations are equivalent. Yet, this will happen solely with payoff operator being as in (12), cost as in (41) and $\delta{=}1$. Another possibility is to define cost in (39) as

$$\theta_\ell(b_\ell^-, b_\ell){\triangleq}1{-}\mathbf{1}_{A_\ell^\delta}(b_\ell^-, b_\ell) \tag{42}$$

with $\delta^{\theta}{=}0$ and

$$A_\ell^\delta{\triangleq}\left\{b_\ell^-, b_\ell{:}b_\ell^-{\in}\mathcal{B}_\ell^-, b_\ell{\in}\mathcal{B}_\ell, \phi(b_\ell^-){\geq}\delta, \phi(b_\ell){\geq}\delta\right\}.$$

We obtain that the (39) is satisfied if and only if our PC (10) is satisfied and in both formulations we have the freedom to select operator ϕ and δ (we still need to assure that the δ is the same in both formulations). Importantly, unlike the cost from (41), the cost from (42) can not be represented as expectation over the state dependent cost. This cost is general belief dependent operator even if the payoff inside is as (12). **Remark**: In general the transition between cost constraint and payoff constraint is not trivial. To do that one must use the linearity of the expectation and the relation between the cost and payoff operators.

B. Duality Based Approach

We now turn to the discussion about duality based approach in continuous spaces suggested in [5]. Suppose that $\delta^{\theta}{=}0$ in (39); The iterative scheme of duality based approach subsumes two steps iteratively solving the following objective

$$\max_{\pi}\min_{\lambda\geq 0}\big(V^{\pi}(b_0;\boldsymbol{\rho}_1){-}\lambda\underbrace{\overbrace{V^{\pi}(b_0;\boldsymbol{\theta}_0)}^{\text{as in (39)}}}_{\geq 0}\big), \tag{43}$$

where one step minimizes for λ and another maximizes for execution stochastic policy π. Here, $\boldsymbol{\theta}_0$ is a vector of cost operators (starting from time 0). The Dual ascend goes towards

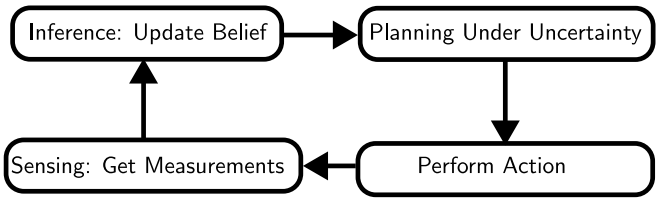


Fig. 5: Autonomy loop.

$V^{\pi}(b_0;\boldsymbol{\theta}_0)=0$. The policy is feasible only in this case. In the λ minimization step, since $V^{\pi}(b_0;\boldsymbol{\theta}_0)\geq 0$, the larger λ will yield smaller objective. Thus, this part of the objective is becoming increasinlgy important with the iterations of the step of the minimization of λ. In practice, the (43) is approximated by the MCTS estimator. Many different suboptimal actions participate within every $\hat{Q}$ in the search tree. This is a direct result of the exploration exploitation tradeoff portrayed by the (7) and the assembling each $\hat{Q}$ from the laces (e.g. if the search tree rooted at b_0, the corresponding action value is $\hat{Q}(b_0a_0)={}^{1}/_{n(b_0a_0)}\sum_j q(b^j_{0:L})$). Another possibility would be on the way up the tree to take the maximum of the previously calculated $\hat{Q}(b(h),a)$ with respect to actions with visitation count $n(ha)>0$. We need to exclude the actions with $n(ha)=0$ in order not to take the initial values q^{init} (Fig. 2b). If we do that, on the way up the tree, instead of completing the lace with future cumulative reward we will complete it with the result of the maximum. Still, the problem of many actions participating in the $\hat{Q}$ remains. Because the result of the maximum is changing as MCTS progresses, still suboptimal actions are participating in each $\hat{Q}$ besides the leaves. This aspect is detrimental to online planning under the safety constraint. If the safety is formulated as in eq. (39) and (41) with $\delta^{\theta}=0$ and the objective is (43), the robot will prefer to depart from unsafe regions as far as possible to ensure that the all expanded actions with at least single posterior are safe at as many beliefs as possible in the search tree. The importance of the all actions being safe increases closer to the root because closer to the root actions participate within more laces. Our approach does not suffer from such a problem since we prune the dangerous actions in the first place.

VII. Simulations and Results

We are now eager to demonstrate our findings in simulations. We compare our approach (Alg. 2 with CPFT-DPW suggested in [5] with our modifications in terms of constraining propagated beliefs as described in Section VI-A. For our approach named PC-PFT-DPW we select the payoff operator ϕ as in (12) and simulate for $\delta=1$. Our baseline follows the averaged constraint formulation in the cost form as in (39) with cost operator as in (42), payoff operator ϕ and δ inside (42) identical to one used in PC-PFT-DPW. As described in Section VI-A we set $\delta^{\theta}=0$. In PC-PFT-DPW we use our safe rollout (Alg. 3) whereas in CPFT-DPW the rollout is set per problem.

We always simulate the trials of a number of autonomy loop cycles. A single cycle of autonomy loop is depicted at Fig. 5. We now specify our problems under consideration.

A. Problems Composition

We present the Safe Lidar Roomba problem.

a) Safe Lidar Roomba: Roomba is a robotic vacuum cleaner that attempts to localize itself in a familiar room and reach the target region. The POMDP state is the position of the agent x, its orientation angle θ, and the status. The status is a binary variable and it tells of whether the robot has reached goal state or stairs. The Roomba action space is defined as $\mathcal{A}=\{a^1,a^2,a^3,a^4,a^5,a^6\}$. The action space $\mathcal{A}$ comprises the pairs (v,ω^v). Each Roomba action is a pair (v,ω^v). It comprises a velocity v and a corresponding angular velocity $\omega^v={}^{\mathrm{d}\theta}/_{\mathrm{d}t}$. We discretized the velocities and the angular velocities and selected the following action space $a^1=(0,-\pi/2)$, $a^2=(0,0)$, $a^3=(0,\pi/2)$, $a^4=(5,-\pi/2)$, $a^5=(5,0)$, $a^6=(5,\pi/2)$. We also have $v^{\text{noise\_coeff}}=0.2$ and $\omega^{\text{noise\_coeff}}=0.05$ such that $v^{\max}=5+0.5\cdot v^{\text{noise\_coeff}}$ and $\omega^{\max}=\pi/2+0.5\cdot\omega^{\text{noise\_coeff}}$. In our simulations we selected $\mathrm{d}t=0.5$ sec. We set $\sigma_{\text{ray}}=0.01$, $\mathrm{rl}_{\min}=0.001$, the stairs penalty is -10000, the goal reward is 10000, the time penalty is -1000.

The robot motion is deterministic with a predefined time step $\mathrm{d}t$, but the action is noisy. When we apply PF each particle is propagated with a noisy action. The velocity noise is drawn from a uniform distribution over the interval $(-0.5v^{\text{noise}},0.5v^{\text{noise}})$. In a similar manner, the angular velocity noise is uniform over the interval $(-0.5\omega^{\text{noise}},0.5\omega^{\text{noise}})$. We draw the noise for each particle and add to the action $a\in\mathcal{A}$ before we apply the motion model. To do so, we first clamp velocity v in the interval $[0,v^{\max}]$. We then clamp ω^v in the interval $[-\omega^{\max},\omega^{\max}]$. The next $\theta'=\theta+\omega^v\cdot\mathrm{d}t$ is wrapped to the interval $(-\pi,\pi]$. After the turn, next position of the agent is $x'=x+v\cdot\mathrm{d}t\cdot(\cos(\theta),\sin(\theta))^T$. If the robot hits the wall, it stops. The status becomes 1 if the robot hits the goal wall (green color in Fig. 6a) and -1 if the robot hits a stairs wall (red color in Fig. 6a). At the end of the motion step, the status is updated and the agent takes an observation. It first determines the ray length rl using the known workspace (room) and the position and heading direction $(\cos(\theta),\sin(\theta))^T$ of the robot. The distribution of the observation conditioned on the robot pose is then Gaussian $\mathcal{N}(\mathrm{rl},\sigma(\mathrm{rl}))$ truncated from the left at zero, where $\sigma(\mathrm{rl})=\sigma_{\text{ray}}\max(\mathrm{rl},\mathrm{rl}_{\min})$. To introduce the safety aspect, similar to [6], we add a rectangular avoid region (Fig. 6). The reward is the expectation over the state reward that is a large reward for reaching the goal, large penalty for reaching the stairs and for each time instance.

b) Dangerous Light Dark: We take inspiration from the one dimensional problem from [5]. The agent lives in a one dimensional space. We reach versatility of action space by the length of actions, such that $\mathcal{A}=\{0,\pm0.5,\pm1,\pm1.5,\pm2,\pm2.5,\pm6\}$. The agent's reward is the multi-objective and subsumes the expected state-dependent reward and the belief-dependent reward to localize itself

$$\rho_{\ell+1}(b_\ell,a_\ell,b_{\ell+1})=\mathbb{E}_{x\sim b_\ell}[r(x,a)]-\mathrm{tr}(\Sigma(b_{\ell+1})) \tag{44}$$

where $\Sigma(b_{\ell+1})$ is the covariance matrix of $b_{\ell+1}$. The agent's state dependent goal is to get to location defined by interval $[-0.75,0.75]$ as fast as possible and execute the action 0 to

Fig. 6: Plot of one of the trials of the execution of the actions from planning in Lidar Roomba problem. Illustration of departing from the unsafe region problem in Lagrangian based methods. The yellow rectangle represents 500 particles of b_0 sampled from a uniform distribution, the pink rectangle is the unsafe region to avoid. The green line is the exit area and the red line is the stairs. On both figures we plotted the ground truth robot positions and the beliefs that transit from yellow to red as time indexes progress; **(a)** CPFT departs as far as possible from the unsafe region. **(b)** Our method behaves as expected: the agent goes to the green area while avoiding the unsafe region.

stay there. Executing it within the interval $[-0.75, 0.75]$ will give the agent a reward of 100, and executing it outside the radius will yield a negative reward of -100. For all other actions the state dependent reward function is $-\text{abs}(x)$. The agent's motion model T is specified as

$$x_{k+1}{=}x_k + a_k + w_k, \tag{45}$$

where w_k follows truncated Gaussian with $\sigma{=}0.1$ and truncation with $\Delta{=}0.5$ around nominal value $x_k{+}a_k$. The light region is located at $x{=}2$ and the observation model is $z_k = x_k{+}v_k$ where $v_k{\sim}\mathcal{N}(0, \sigma(x_k))$ and $\sigma(x){=}\mathbf{1}_{\{x:|x-2|\leq 1\}}(x) \cdot 10^{-10}{+}\mathbf{1}_{\{x:|x-2|>1\}}(x){\cdot}|x{-}2|$. At $x{=}{-}0.75$ there is a cliff such that if agent falls it crashes. In addition around the light source there is a pit. The safe space is $\mathcal{X}^{\text{safe}}{=}\{-0.75 < x < 1\} \cap \{x > 3\}$. The prior belief $b_k(x_k)$ is Gaussian $\mathcal{N}(7, 20)$ truncated such that its support is $[6, 8]$.

c) Simultaneous Localization and Mapping with Certain and Uncertain Obstacles (SLAM): Our action space comprises motion primitives and zero action, $\mathcal{A} = \{\rightarrow, \nearrow, \uparrow, \nwarrow, \leftarrow, \swarrow, \downarrow, \searrow, \mathbf{0}\}$. If robot selected zero action $\mathbf{0}$, we do not apply motion model to each particles but do resampling to take into account received observation. This allows to robot not move if it is too dangerous. In this problem the agent and the uncertain obstacles (landmarks) have circular form. The motion model T for the agent is

$$x_{k+1} = x_k + a_k + w_k. \tag{46}$$

Our goal is to epitomize the importance of safe state trajectories versus solely safe beliefs trajectory. Towards this end we draw randomly many tiny obstacles so as one way or another the unsafe trajectory will be encountered by the robot if planning was done with pushing forward in time also the unsafe particles. Our observation model is bearing range with the noise inversely proportional to the distance to uncertain obstacle, the landmark l. The motion model for the landmark is

$$l_{k+1} = l_k. \tag{47}$$

We maintain belief over the last robot pose and the landmark. The observation model reads $z_k = x_k - l_k + v_k$ where $v_k \sim \mathcal{N}(0, \Sigma_k(x_k, l_k))$. The $\Sigma_k(x_k, l_k)$ is a diagonal matrix with main diagonal $\sigma_k^2(x_k, l_k){=}\|x_k - l_k\|_2$.

1) Pushbox 2D Problem: In this section we first describe our variation of PushBox2D problem with soft safety. We then transfer the soft safety to our formulation described in Section III. Clearly soft safety is not good enough. In cases there is no feasible solution exists we do not want robot to do any operations. Instead, it is desirable that robot decide that the goal is not achievable. The Pushbox2D problem is motivated by air hockey. A disk-shaped robot (blue disk) must push a disk-shaped puck (red disk) into a goal area (green circle) by bumping into it while avoiding any collision of itself and the puck with an edge area (black area). The state space consists of the xy-locations of both the robot and the puck, i.e., $\mathcal{X}{=}\mathbb{R}^4$, while the action space is defined by motion primitives of unit length . The action Null is terminal. If the robot is not in contact with the puck during a move, the state evolves according to

$$\begin{aligned} x'{=}(f(x,a) + w, (x^p, y^p))^T, \quad w{\sim}\mathcal{N}(0, W), \\ f(x,a){=}(x^r + a^x, y^r + a^y)^T, \end{aligned} \tag{48}$$

where (x^r, y^r) and (x^p, y^p) are the xy-coordinates the robot and the puck respectively, corresponding to state x, and (a^x, a^y) is the displacement vector corresponding to action a. If the robot bumps into the puck, the next position (x'^p, y'^p) of the puck is

$$\begin{pmatrix} x'^p \\ y'^p \end{pmatrix} = \begin{pmatrix} x^p \\ y^p \end{pmatrix} {+}5r_s \left(\left(\begin{pmatrix} a^x \\ a^y \end{pmatrix} \cdot n \right) \left(n{+} \begin{pmatrix} r^x \\ r^y \end{pmatrix} \right) \right), \tag{49}$$

where n is the unit directional vector from the center of the robot to the center of the puck at the time of contact, and r_s is a random variable drawn from a truncated Gaussian distribution $\mathcal{N}(\mu, \sigma^2, l, u)$, which is the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ truncated to the interval $[l, u]$. The variables r_x and r_y are random variables drawn from a truncated Gaussian distribution $\mathcal{N}(0.0, 0.12, -0.1, 0.1)$. The prior belief $b_0(x_0)$ is a Gaussian over the robot position and deterministic over the puck position. The robot has access to a noisy bearing sensor to localize itself observing the puck and a noise-free collision sensor which detects contacts between the robot and the puck. Specifically, given a state $x{\in}\mathcal{X}$, an observation (o_c, o_b) consists of a binary component o_c which indicates

(a) (b)

Fig. 7: **(a)** Conceptual visualization of the PushBox2D problem. The agent is the blue circle. The puck is the red circle. The goal is the green circle. **(b)** The observation noise intensity map. Light green color denotes the lower noise intensity.

whether or not a contact between the robot and the puck occurred, and a bearing range component o^{br} calculated as

$$o^{\mathrm{br}}=h(x)+v,\ \ h(x)=(x^r-x^p, y^r-y^p)^T, \tag{50}$$

where x^r , y^r and x^p , y^p are the xy-coordinates of the robot and the puck corresponding to the state x, and r_o is a random angle (expressed in radians) drawn from a truncated Gaussian distribution with magnitude of the variance dependent on the position on the map of the robot as in the Fig. 7b). The reward for the MCTS baseline is the distance to goal of the puck with boundary region and other obstacles

$$\begin{aligned}\rho(b)=-\,\mathbb{E}_{x\sim b}[\|x^p-x^g\|_2]-1000\cdot\mathbf{1}_{\{o_c==1\}}(o_c)+\\ \mathrm{P}(\{x\in\mathcal{X}^{\mathrm{safe}}\}|b)-\mathcal{H}(b).\end{aligned} \tag{51}$$

Here we have a soft chance constraints since it is not clear how to enforce chance constraints to MCTS. In case of our approach we shift the $\mathrm{P}(\{x\in\mathcal{X}^{\mathrm{safe}}\}|b)$ component to our probabilistic constraint.

B. Experiments

We benchmarked our approach using the Lidar Roomba problem, the famous Light Dark problem, active SLAM problem and PushBox2D problem. We have shown the issue described in Section VI-B on Lidar Roomba and the satisfiability of the constraint solely at the limit of MCTS convergence situation on a Light Dark problem. Considering an active SLAM problem, we visualized the importance of making belief safe in planning, as described in Section IV-A. With PushBox2D problem we verified the importance of making the propagated belief safe and simulated for several values of δ.

In the Lidar Roomba problem the robot performs at most 50 cycles of autonomy loop. In the Light Dark problem, the robot performs 5 cycles of autonomy loop. We do 70 trials of each such a scenario and approximate the $\mathrm{P}(S|b_0)\approx\hat{\mathrm{P}}(S|b_0)=\sum_{i=1}^{70}\mathbf{1}_S(\tau_0^i)/70$ using the simulated trajectories. The event $S=\{\tau_0\in\times_{\ell=0}^{L}\mathcal{X}_\ell^{\mathrm{safe}}\}$, where $\tau_0=x_{0:L}$ means each state in the actual robot trajectory starting at time 0 was safe. $\hat{V}^*(b_0;\boldsymbol{\rho}_1)=\frac{1}{70}\sum_{i=1}^{70}\sum_{\ell=0}^{L(i)}\rho_{\ell+1}(b_\ell^i,a_\ell^i,b_{\ell+1}^i)$, where in Roomba $L(i)\leq 50$ since we have terminal state and in Light Dark $L(i)\equiv 5$.

In SLAM problem the robot makes 50 trials of at most 20 cycles of autonomy loop. In PushBox2D problem the robot performs 20 trials of at most 20 cycles of autonomy loop.

In all four problems we take 500 belief particles.

C. Discussion and Results Interpretation

Before we proceed it shall be noted that the number of collisions and approximated probability that the trajectory is safe in relevant tables are connected as follows

$$\hat{\mathrm{P}}(\{\tau_0\in\times_{\ell=0}^{L}\mathcal{X}_\ell^{\mathrm{safe}}|b_0)=\hat{\mathrm{P}}(S|b_0)=1-\tfrac{\text{num. coll}}{\text{num. trials}}. \tag{52}$$

Let us interpret the results.

a) Roomba: Table I corresponds to Roomba problem. From Table I we behold that the cumulative reward yielded by CPFT is much lower than our method. In particular, the Roomba never reaches the goal and not stairs. We also calculate an empirical mean of the distance between the terminal Roomba position and the middle of the goal region. As we see, CPFT makes Roomba to depart from the obstacle as far as possible.

b) Light Dark: Table II corresponds to Light Dark problem. In Table II we see that with a small number of MCTS iterations, CPFT makes 16 collisions from 70 trials in contrast to 0 collisions with our technique. We illustrate the scenario in Fig. 8. In this problem it is dangerous to the agent to jump to desired interval. This is because the width of the belief b_0 is larger than the desired area and robot can fall off the cliff or to the pit (assuming the motion model as in (45) and without the stochastic noise w_k). Our approach prevent the robot to jump to desired area since any belief particle can be the ground truth.

c) SLAM: For the SLAM problem we study the influence of making posterior belief safe before pushing forward in time. Since this aspect stemmed from Chance Constraints, to differentiate between two approaches, in this study we change the name of our approach with nullifying the unsafe part of belief to CC-PC-PFT (Alg. 2). We call our method, with pushing forward in time with action and the observation generally unsafe belief, PC-PFT. Let us remind that the difference in the inner constraint as such. Instead of payoff operator as (12) we set

$$\begin{aligned}\phi(\bar{b}_\ell)=\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\}\big|\bar{b}_\ell\big)=\\ \mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\}\big|b_0,a_{0:\ell-1},z_{1:\ell},\textstyle\bigcap_{i=0}^{\ell-1}\{x_i\in\mathcal{X}_i^{\mathrm{safe}}\}\big)\end{aligned} \tag{53}$$

$$\begin{aligned}\phi(\bar{b}_\ell^-)=\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\}\big|\bar{b}_\ell^-\big)=\\ \mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\mathrm{safe}}\}\big|b_0,a_{0:\ell-1},z_{1:\ell-1},\textstyle\bigcap_{i=0}^{\ell-1}\{x_i\in\mathcal{X}_i^{\mathrm{safe}}\}\big).\end{aligned} \tag{54}$$

Table III presents the results for SLAM in our first setup with tiny obstacles. Here our setup is as follows. We have

Fig. 8: Illustration of faulty scenario of CPFT as opposed to our approach **(a)** CPFT **(b)** Our Alg. 2.

Model	tree queries	$\hat{V}^*(b_0;\boldsymbol{\rho}_1)$	$\hat{\mathbb{E}}[\|\|x^t - x^g\|\|_2^2]\pm$std	$\hat{\mathrm{P}}(S\|b_0)$	num. coll
CPFT-DPW	1000	-46500.0 ± 136.31	829.78 ± 525.88	1	0/70
PCPFT-DPW	1000	-28086 ± 14399	56.86 ± 215.12	1	0/70

TABLE I: The Lidar Roomba problem.

Model	tree queries	$\hat{V}^*(b_0;\boldsymbol{\rho}_1)$	$\hat{\mathrm{P}}(S\|b_0)$	num. coll
CPFT-DPW	15	-75.67 ± 57.66	0.77	16/70
PCPFT-DPW	15	-115.27 ± 94.28	1	0/70

TABLE II: The Light Dark problem.

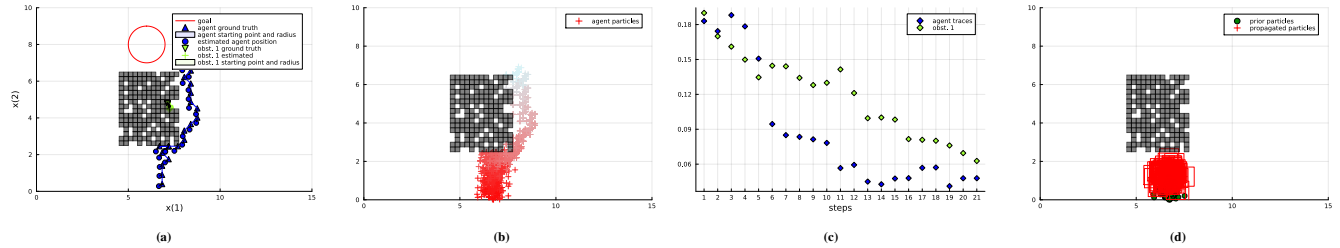


Fig. 9: This simulation setup is associated with Table III. In this figure we plot one of the trials shown in Table III. Here we nullify unsafe part of the belief in planning and run CCPC-PFT. **(a)** Here, we plot the goal, agent ground truth, estimated agent positions and the obstacles; **(b)** Belief particles, where the colors symbolize the time instance; **(c)** Traces of the agent and the landmark (obstacle); **(c)** Visualization of the truncation. Here we move each particle of b_0 with action selected by the agent and plot the truncation region of the stochastic motion model.

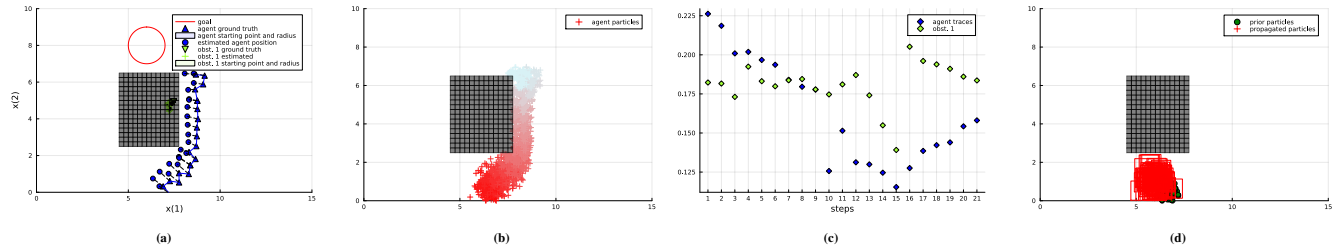


Fig. 10: This simulation setup is associated with Table IV columns related to CCPC-PFT-DPW and here we show one of the trials. In this figure we nullify unsafe part of the belief in planning. **(a)** Here, we plot the goal, agent ground truth, estimated agent positions and the obstacles; **(b)** Belief particles, where the colors symbolize the time instance; **(c)** Traces of the agent and the landmark (obstacle); **(c)** Visualization of the truncation. Here we move each particle of b_0 with action selected by the agent and plot the truncation region of the stochastic motion model.

a rectangular area where we randomly sow rectangular tiny obstacles without replacement. It means if we randomly sow the number of tiny obstacles equal to the number of cells within the large rectangle we will obtain a complete large rectangle. We randomly sow the tiny obstacles in each trial. We make transition model of the agent (46) deterministic by nullifying the noise. With drawing 80% of tiny obstacles (Fig. 9) the PC-PFT-DPW, without making belief safe and maintaining a pair of the beliefs, the scenario in Fig. 9 reached the belief node where **all the actions were claimed unsafe and pruned**, even the $\mathbf{0}$ action. As we have seen in the simulation $\mathbf{0}$ action was pruned the last and this is a direct result of the fact that unsafe belief particles were propagated with $\mathbf{0}$ action and updated with received observation. When we do the same operation previously making belief safe, we obtain again the safe belief since particles were propagated with $\mathbf{0}$ action and, therefore, stay at the same places.

In our second setup we fill the complete rectangle with tiny obstacles in a random manner as previously debated (Fig. 11). We show our results in Table IV. We did not obtained a significant difference in two approaches. Interestingly, as we see the safety is much challenging in this problem due to challenging robot localization with simultaneous mapping of uncertain single landmark. Our prior b_0 in SLAM problem is

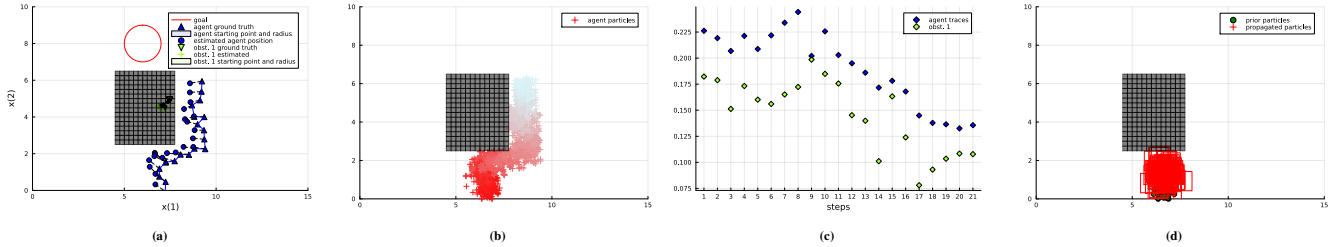


Fig. 11: This simulation setup is associated with Table IV columns related to PC-PFT-DPW and here we show one of the trials. In this figure we **do not** nullify unsafe part of the belief in planning. **(a)** Here, we plot the goal, agent ground truth, estimated agent positions and the obstacles; **(b)** Belief particles, where the colors symbolize the time instance; **(c)** Traces of the agent and the landmark (obstacle); **(c)** Visualization of the truncation. Here we move each particle of b_0 with action selected by the agent and plot the truncation region of the stochastic motion model.

TABLE III: 50 Trials of at most 20 cycles of autonomy loop Fig. 5 where planning sessions implemented by Algorithm CCPC-PFT-DPW versus PC-PFT-DPW. Same seed in both algorithms. This problem is the **SLAM** described in Section VII-A0c in **our first scenario** shown at Fig. 9. Here we study the number of collisions and the reward value.

Parameters		$\hat{\mathrm{P}}(S\|b_0)$		num coll.		mean cum. rew. $\pm$ std	
Operator ϕ	δ	CCPC-PFT-DPW	PC-PFT-DPW	CCPC-PFT-DPW	PC-PFT-DPW	CCPC-PFT-DPW	PC-PFT-DPW
(12)	0.8	0.64	-	18/50	-	-106.37 ± 12.37	-

TABLE IV: 50 Trials of at most 20 cycles of autonomy loop Fig. 5 where planning sessions implemented by Algorithm CCPC-PFT-DPW versus PC-PFT-DPW. Same seed in both algorithms. This problem is the **SLAM** described in Section VII-A0c in **our second scenario** shown at Fig. 10 and Fig. 11. Here we study the number of collisions and the reward value.

Parameters		$\hat{\mathrm{P}}(S\|b_0)$		num coll.		mean cum. rew. $\pm$ std	
Operator ϕ	δ	CCPC-PFT-DPW	PC-PFT-DPW	CCPC-PFT-DPW	PC-PFT-DPW	CCPC-PFT-DPW	PC-PFT-DPW
(12)	0.8	0.6	0.6	28/70	28/70	-109.92 ± 11.55	-106.68 ± 12.77

Gaussian with diagonal variances of 0.1.

d) PushBox2D: We constructed a challenging scenario where evading the obstacle significantly complicates putting the puck into the hole (the goal). Let us contemplate the results presented in Table. V. We selected $m = 10$ and $\epsilon = 0$ in rollout summarized by Alg. 3. As we see, constraining the propagated belief significantly improves safety while preserving reaching the goal by the puck.

VIII. Conclusions

In this work, we introduced an anytime online approach to perform Safe and Risk Aware Belief Space Planning in continuous domains in terms of states, actions, and observations. We rigorously analyzed our approach in terms of convergence. Our prominent novelty is assuring safety with respect to the belief tree expanded so far. As opposed to SOTA in continuous domains, we are not mixing safe and dangerous actions in the search tree. Our belief tree is safe with respect to our PC and consist solely of the safe actions. Moreover, when our PC is satisfied, it is satisfied starting from each belief action node, ensuring a match in a current planning session and future planning sessions. We corroborated our theoretical development by simulating **four** different problems in continuous domains. Each problem exhibited a different phenomenon caught by our methodology.

IX. Acknowledgment

This work was supported by the Israel Science Foundation (ISF).

References

[1] O. Madani, S. Hanks, and A. Condon, "On the undecidability of probabilistic planning and related stochastic optimization problems," *Artificial Intelligence*, vol. 147, no. 1-2, pp. 5–34, 2003.

[2] P. Santana, S. Thiébaux, and B. Williams, "Rao*: An algorithm for chance-constrained pomdp's," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[3] A. Zhitnikov and V. Indelman, "Risk aware adaptive belief-dependent probabilistically constrained continuous pomdp planning," *arXiv preprint arXiv:2209.02679*, 2022.

[4] ——, "Simplified continuous high dimensional belief space planning with adaptive probabilistic belief-dependent constraints," *IEEE Trans. Robotics*, 2024.

[5] A. Jamgochian, A. Corso, and M. J. Kochenderfer, "Online planning for constrained pomdps with continuous spaces through dual ascent," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 33, no. 1, 2023, pp. 198–202.

[6] A. Jamgochian, H. Buurmeijer, K. H. Wray, A. Corso, and M. J. Kochenderfer, "Constrained hierarchical monte carlo belief-state planning," *arXiv preprint arXiv:2310.20054*, 2023.

[7] J. Lee, G.-H. Kim, P. Poupart, and K.-E. Kim, "Monte-carlo tree search for constrained pomdps," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[8] E. Altman, *Constrained Markov decision processes*. CRC Press, 1999.

[9] A. Beck, *First-order methods in optimization*. SIAM, 2017.

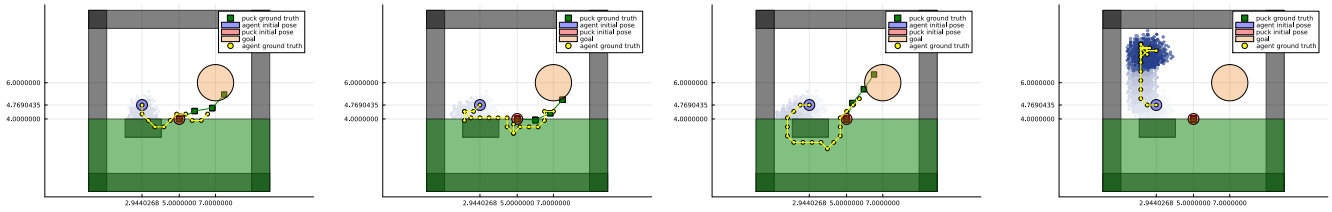
[10] Q. H. Ho, T. Becker, B. Kraske, Z. Laouar, M. Feather, F. Rossi, M. Lahijanian, and Z. N. Sunberg, "Recursively-constrained partially observable markov decision processes," *arXiv preprint arXiv:2310.09688*, 2023.

[11] R. Munos, *From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning*, 2014.

[12] M. Ajdarów, Š. Brlej, and P. Novotnỳ, "Shielding in resource-constrained goal pomdps," in *AAAI Conf. on Artificial Intelligence*, vol. 37, no. 12, 2023, pp. 14 674–14 682.

[13] G. Mazzi, A. Castellini, and A. Farinelli, "Risk-aware shielding of partially observable monte carlo planning policies," *Artificial Intelligence*, vol. 324, p. 103987, 2023.

[14] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2010, pp. 2164–2172.



(a) $\delta = 0.0$ **(b)** $\delta = 0.3$ **(c)** $\delta = 0.7$ **(d)** $\delta = 1.0$

Fig. 12: Visualization of actual PushBox2D simulation with several values of δ.

TABLE V: 20 Trials of at most 20 autonomy loop cycles of Algorithms CCPC-PFT-DPW versus PC-PFT-DPW. Same seed in both algorithms. This problem is the **2DPushBox** described in Section VII-A1. The operator ϕ conforms to (12) and (13).

Parameters					$\hat{\mathrm{P}}(S\|b_0)$ in accord to (52)		Est. prob. of puck reaching the goal		mean cum. rew. $\pm$ std	
tree queries	δ	L	propagated constr.	rollout	CCPC-PFT-DPW	PC-PFT-DPW	CCPC-PFT-DPW	PC-PFT-DPW	CCPC-PFT-DPW	PC-PFT-DPW
1500	0.0	20	Yes	Yes	0.1	0.0	1.0	1.0	-20.39 ± 6.12	-20.73 ± 4.32
1500	0.3	20	Yes	Yes	0.45	0.3	0.85	0.85	-37.33 ± 11.93	-32.31 ± 11.58
1500	0.7	20	Yes	Yes	0.65	0.7	0.65	0.8	-43.04 ± 6.82	-41.56 ± 9.31
1500	0.7	20	No	Yes	0.2	0.3	0.9	0.9	-38.38 ± 9.69	-34.07 ± 9.20
1500	0.7	20	Yes	No	0.55	0.65	0.0	0.0	-59.82 ± 0.25	-59.85 ± 0.26
1500	1.0	20	Yes	Yes	-	1.0	-	0.0	-	-60.18 ± 0.18
1500	1.0	20	No	Yes	-	0.2	-	0.9	-	-35.81 ± 7.59

[15] R. J. Moss, A. Jamgochian, J. Fischer, A. Corso, and M. J. Kochenderfer, "Constrainedzero: Chance-constrained pomdp planning using learned probabilistic failure surrogates and adaptive safety constraints," *arXiv preprint arXiv:2405.00644*, 2024.

[16] G. Chou, N. Ozay, and D. Berenson, "Safe output feedback motion planning from images via learned perception modules and contraction theory," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2022, pp. 349–367.

[17] S. Dean, A. Taylor, R. Cosner, B. Recht, and A. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," in *Conference on Robot Learning*. PMLR, 2021, pp. 654–670.

[18] Z. Sunberg and M. Kochenderfer, "Online algorithms for pomdps with continuous state, action, and observation spaces," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 28, no. 1, 2018.

[19] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *European conference on machine learning*. Springer, 2006, pp. 282–293.

[20] D. Auger, A. Couetoux, and O. Teytaud, "Continuous upper confidence trees with polynomial exploration–consistency," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part I 13*. Springer, 2013, pp. 194–209.

[21] A. Zhitnikov and V. Indelman, "Simplified risk aware decision making with belief dependent rewards in partially observable domains," *Artificial Intelligence, Special Issue on "Risk-Aware Autonomous Systems: Theory and Practice"*, 2022.

[22] M. H. Lim, T. J. Becker, M. J. Kochenderfer, C. J. Tomlin, and Z. N. Sunberg, "Optimality guarantees for particle belief approximation of pomdps," *Journal of Artificial Intelligence Research*, vol. 77, pp. 1591–1636, 2023.

APPENDIX A
PROOF OF LEMMA 1(REPRESENTATION OF THE VALUE FUNCTION).

Before we begin, let us clarify that when we write $\{\mathbb{P}^{\pi}_{\ell}(a_\ell|b_\ell)\}_{\ell=1}^{L-1}$, the a_ℓ and b_ℓ inside the $\{\mathbb{P}_\ell(a_\ell|b_\ell)\}_{\ell=1}^{L-1}$ can be a random variables for all relevant ℓ or corresponding realizations. However, $\{\mathbb{P}^{\pi}_{\ell}\}_{\ell=1}^{L-1}$ is the series of distributions of length $L-1$ and corresponding actions and beliefs are unknown. In addition, we remind to the reader that $\pi_\ell(a_\ell, b_\ell) = \mathbb{P}^{\pi}_{\ell}(a_\ell|b_\ell) \quad \forall \ell \in 1:L-1$ and $\pi=\{\mathbb{P}^{\pi}_{\ell}\}_{\ell=1}^{L-1}$.

$$\mathbb{E}^{\mathrm{T,O}}\big[\sum_{\ell=0}^{L-1}\rho_{\ell+1}(b_\ell,a_\ell,b_{\ell+1})\big|b_0,\pi\big]= \sum_{\ell=0}^{L-1}\mathbb{E}^{\mathrm{T,O}}\big[\rho_{\ell+1}(b_\ell,a_\ell,b_{\ell+1})\big|b_0,\pi\big]=\sum_{\ell=0}^{L-1}\mathbb{E}^{\mathrm{T,O}}\big[\rho_{\ell+1}\big|b_0,\pi\big] \tag{55}$$

$$\mathbb{E}^{\mathrm{T,O}}\big[\rho_{\ell+1}\big|b_0,\pi\big]=\int_{\rho_{\ell+1}}\rho_{\ell+1}\mathbb{P}(\rho_{\ell+1}|b_0,\{\mathbb{P}^{\pi}_{i}\}_{i=0}^{L-1})\mathrm{d}\rho_{\ell+1}=$$

$$\int_{\rho_{\ell+1}}\rho_{\ell+1}\int_{\substack{b_{1:\ell}\\ a_{0:\ell}\in\times_{i=1}^{\ell}\mathcal{A}}}\mathbb{P}(\rho_{\ell+1},b_{1:\ell},a_{1:\ell}|b_0,\{\mathbb{P}^{\pi}_{i}\}_{i=0}^{L-1})\mathrm{d}b_{1:\ell}\mathrm{d}a_{0:\ell}\mathrm{d}\rho_{\ell+1}=$$

$$\int_{\rho_{\ell+1}}\rho_{\ell+1}\int_{\substack{b_{1:\ell}\\ a_{0:\ell}\in\times_{i=1}^{\ell}\mathcal{A}}}\mathbb{P}(\rho_{\ell+1}|b_{0:\ell},a_{0:\ell})$$

$$\mathbb{P}(b_{1:\ell},a_{0:\ell}|b_0,\{\mathbb{P}^{\pi}_{i}\}_{i=0}^{L-1})\mathrm{d}b_{1:\ell}\mathrm{d}a_{0:\ell}\mathrm{d}\rho_{\ell+1}=$$

$$\int_{\substack{b_{1:\ell}\\ a_{0:\ell}\in\times_{i=1}^{\ell}\mathcal{A}}}\left(\int_{\rho_{\ell+1}}\rho_{\ell+1}\mathbb{P}(\rho_{\ell+1}|b_{0:\ell},a_{0:\ell})\mathrm{d}\rho_{\ell+1}\right)$$

$$\mathbb{P}(b_{1:\ell},a_{0:\ell}|b_0,\{\mathbb{P}^{\pi}_{i}\}_{i=0}^{L-1})\mathrm{d}b_{1:\ell}\mathrm{d}a_{0:\ell} \tag{56}$$

We now use a chain rule from the future time back on $\mathbb{P}(b_{1:\ell},a_{0:\ell}|b_0,\{\mathbb{P}^{\pi}_{i}\}_{i=0}^{L-1})$ an got

$$\mathbb{E}^{\mathrm{T,O}}\big[\rho_{\ell+1}\big|b_0,\pi\big] = \mathop{\mathbb{E}}_{a_0}\Big[\mathop{\mathbb{E}}_{b_1}\Big[\mathop{\mathbb{E}}_{a_1}\Big[\mathop{\mathbb{E}}_{b_2}\Big[\dots \mathop{\mathbb{E}}_{a_\ell}\big[\mathbb{E}\big[\rho_{\ell+1}|b_\ell,a_\ell\big]\big|b_\ell,\pi_\ell\big]\dots \Big|b_1,a_1\Big]\Big|b_1,\pi_1\Big]\Big|b_0,a_0\Big]\Big|b_0,\pi_0\Big]. \tag{57}$$

APPENDIX A
PROOF OF LEMMA 1(REPRESENTATION OF THE VALUE FUNCTION).

■

APPENDIX B
PROOF OF THEOREM 2(REPRESENTATION OF OUR OUTER CONSTRAINT).

Before we begin, let us clarify that when we write

$$\mathrm{P}\left(\left(\mathbf{1}_{A_0^\delta}(b_0)\prod_{\ell=1}^{L}\mathbf{1}_{A_\ell^\delta}(b_\ell)\right)=1|b_0,a_0,\{\mathbb{P}_\ell^\pi\}_{\ell=1}^{L-1}\right),$$

the actions a_ℓ and the beliefs b_ℓ inside $\{\mathbb{P}_\ell^\pi\}_{\ell=1}^{L-1}$ are unknown random quantities. In addition, we remind to the reader that $\pi_\ell(a_\ell,b_\ell)=\mathbb{P}_\ell^\pi(a_\ell|b_\ell)\quad\forall\ell\in 1{:}L-1$ and $\pi=\{\mathbb{P}_\ell^\pi\}_{\ell=1}^{L-1}$. Moreover, in this paper each posterior belief is associated with corresponding propagated belief. Therefore we can rescind the explicit dependence of the indicator on propagated belief.

$$\begin{aligned}\mathbb{E}\Big[\mathbf{1}_{A_0^\delta}(b_0)\prod_{\ell=1}^{L}\mathbf{1}_{A_\ell^\delta}(b_\ell)|b_0,a_0,\{\mathbb{P}_\ell^\pi\}_{\ell=1}^{L-1}\Big]=\\ \int\limits_{\substack{b_{1:L}\\ a_{1:L-1}\in\times_{\ell=1}^{L-1}\mathcal{A}}}\mathbf{1}_{A_0^\delta}(b_0)\prod_{\ell=1}^{L}\mathbf{1}_{A_\ell^\delta}(b_\ell)\\ \mathbb{P}(b_{1:L},a_{1:L-1}|b_0,a_0,\{\mathbb{P}_\ell^\pi\}_{\ell=1}^{L-1})\mathrm{d}b_{1:L}\mathrm{d}a_{1:L-1}.\end{aligned}\tag{58}$$

Now, we need to handle $\mathbb{P}(b_{1:L},a_{0:L-1}|b_0,a_0,\{\mathbb{P}_\ell^\pi\}_{\ell=1}^{L-1})$. It holds that

$$\mathbb{P}(b_{1:L},a_{1:L-1}|b_0,a_0,\{\mathbb{P}_\ell^\pi\}_{\ell=1}^{L-1})$$

equals to

$$\begin{aligned}\mathbb{P}(b_{2:L},a_{2:L-1}|b_0,a_0,b_1,a_1,\mathbb{P}_1^\pi(a_1|b_1),\{\mathbb{P}_\ell^\pi\}_{\ell=2}^{L-1})\\ \mathbb{P}(b_1,a_1|b_0,a_0,\{\mathbb{P}_\ell\}_{\ell=1}^{L-1})=\\ \mathbb{P}(b_{2:L},a_{2:L-1}|b_1,a_1,\{\mathbb{P}_\ell^\pi\}_{\ell=2}^{L-1})\mathbb{P}_1^\pi(a_1|b_1)\mathbb{P}(b_1|b_0,a_0)=\\ \mathbb{P}(b_L|b_{L-1},a_{L-1})\prod_{\ell=1}^{L-1}\mathbb{P}_\ell^\pi(a_\ell|b_\ell)\mathbb{P}(b_\ell|b_{\ell-1},a_{\ell-1}).\end{aligned}\tag{59}$$

We now merge (58) and (59), and land at the desired result

$$\begin{aligned}\mathbf{1}_{A_0^\delta}(b_0)\int\limits_{\substack{b_{1:L}\\ a_{1:L-1}\in\times_{i=1}^{L-1}\mathcal{A}}}\mathbf{1}_{A_L^\delta}(b_L)\mathbb{P}(b_L|b_{L-1},a_{L-1})\\ \prod_{\ell=1}^{L-1}\Big(\mathbb{P}_\ell^\pi(a_\ell|b_\ell)\mathbb{P}(b_\ell|b_{\ell-1},a_{\ell-1})\mathbf{1}_{A_\ell^\delta}(b_\ell)\Big)\mathrm{d}b_{1:L}\mathrm{d}a_{1:L-1}=\\ \mathbf{1}_{A_0^\delta}(b_0)\int_{b_1}\mathbb{P}(b_1|b_0,a_0)\mathbf{1}_{A_1^\delta}(b_1)\int_{a_1}\mathbb{P}_1^\pi(a_1|b_1)\Big(\dots\\ \int_{b_L}\mathbf{1}_{A_L^\delta}(b_L)\mathbb{P}(b_L|b_{L-1},a_{L-1})\mathrm{d}b_L\dots\Big)\mathrm{d}a_1\mathrm{d}b_1=\\ \mathbf{1}_{A_0^\delta}(b_0)\mathop{\mathbb{E}}_{b_1}\Bigg[\mathbf{1}_{A_1^\delta}(b_1)\mathop{\mathbb{E}}_{a_1}\Bigg[\mathop{\mathbb{E}}_{b_2}\Big[\mathbf{1}_{A_2^\delta}(b_2)\dots\\ \mathbb{E}\big[\mathbf{1}_{A_L^\delta}(b_L)|b_{L-1},a_{L-1}\big]\dots\Big|b_1,a_1\Big]\Bigg|b_1,\pi_1\Bigg]\Bigg|b_0,a_0\Bigg]=\\ \mathbf{1}_{A_0^\delta}(b_0)\mathop{\mathbb{E}}_{b_1}\Bigg[\mathop{\mathbb{E}}_{a_1\sim\mathbb{P}_1^\pi(a_1|b_1)}\Bigg[\\ \mathrm{P}\big(\big(\prod_{\ell=1}^{L}\mathbf{1}_{A_\ell^\delta}(b_\ell)\big)=1\Big|b_1,a_1,\pi\Big)\Bigg|b_1,\pi_1\Bigg]\Bigg|b_0,a_0\Bigg].\end{aligned}$$

■

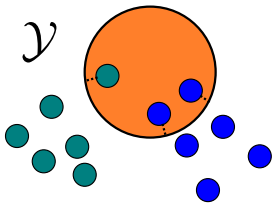


Fig. 13: Illustration of complex safety operators in multirobot setting.

APPENDIX C
PROOF OF THEOREM 1 (NECESSARY CONDITION FOR THEORETICAL POSTERIORS TO BE SAFE)

For the necessary condition we prove the inverse implication. Suppose that $\forall z_\ell\in\mathcal{Z}$ it holds that $\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-,z_\ell\big)\geq\delta$. We arrive at

$$\Big(\int_{z_\ell\in\mathcal{Z}}\mathrm{P}\big(\{x_\ell\in\mathcal{X}_\ell^{\text{safe}}\}\big|h_\ell^-,z_\ell\big)\mathbb{P}(z_\ell|h_\ell^-)\mathrm{d}z_\ell\Big)\geq\delta.\tag{60}$$

■

APPENDIX D
VAR AND CVAR AS SAFETY COST OPERATORS.

Suppose we have particle represented belief and the obstacle of the circular form Fig. 13. In addition we have two robots teal and blue. Each particle of the belief is a concatenated position of each robot such that if x is a particle, the $x[1{:}2]$ corresponds to the first robot and $x[3:4]$ corresponds to the second robot. We shall check such a constraint for each robot separately. For clarity let x denote the position of the one of the robots. Suppose the map $\mathcal{M}$ is given. We first define a distance from the safe space $\mathcal{Y}\subseteq\mathcal{M}$ as $\mathrm{dist}(x,\mathcal{Y})=\min_{y\in\mathcal{Y}}\|x-y\|_2$. We then define Value at Risk (VaR) as

$$\begin{aligned}\theta(b)\triangleq\mathrm{VaR}_\alpha^b[\mathrm{dist}(x,\mathcal{Y})]=\\ \min\{\xi|\mathrm{P}(\mathrm{dist}(x,\mathcal{Y})\leq\xi)\geq 1-\alpha\}.\end{aligned}\tag{61}$$

The Conditional Value at Risk (CVaR) is specified as

$$\begin{aligned}\theta(b)\triangleq\mathrm{CVaR}_\alpha^b[\mathrm{dist}(x,\mathcal{Y})]=\\ \mathbb{E}[\mathrm{dist}(x,\mathcal{Y})|\{x:\mathrm{dist}(x,\mathcal{Y})\geq\mathrm{VaR}_\alpha^b[\mathrm{dist}(x,\mathcal{Y})]\}].\end{aligned}\tag{62}$$

Both of these operators are **cost** operators.

Chapter 5

Discussion

This research has three main components: belief-dependent rewards, risk awareness, and the simplification paradigm. This discussion elaborates on these components.

We recognize a gap in current state-of-the-art algorithms that utilize anytime planning approaches. These approaches leverage an assumption that the belief-dependent reward is merely the expected value of state-dependent reward with respect to a belief. This assumption is unrealistic in many real-life problems such as autonomous localization, SLAM, and Informative Planning. An inspiring challenge is then to allow fast and efficient POMDP planning with belief-dependent rewards while keeping optimality guarantees.

Another aspect that requires attention is risk awareness. An indispensable part of POMDP planning is the distribution of the long-term rewards given a candidate policy. The goal of decision-making is to select an optimal policy. Due to the fact that distributions of the rewards are not comparable, it is necessary to apply an additional operator. Commonly, this operator is averaging (expectation). The expectation is not truly distribution-aware nor risk-aware since many distributions can have the same expectation. In this thesis, we aim to tackle this aspect while maintaining the mathematical soundness of the solution.

Belief-dependent rewards and risk-aware operators, instead of averaging, increase the computational burden even more. This is because most of the algorithms extensively use the assumptions mentioned above. To reiterate, the first assumption is that the belief-dependent reward is nothing more than the expected value of state-dependent reward with respect to belief. The second assumption is the objective operator being the expectation. Moreover, in many problems in robotics, the constraints naturally arise. The most renowned constraint in robotics is **collision avoidance**. Another important constraint is Information Gain. The question of when the robot should stop to explore the terrain is still considered an open problem.

Therefore, in these settings, the simplification is invaluable. Simplification is the substitution of any part of the original decision-making problem by cheaper-to-calculate counterparts, that can be utilized in decision-making instead of original elements, while providing guarantees on the impact of such a substitution.

The first three works are under the umbrella of **simplification paradigm**, the first two, the fourth, and the fifth papers are also in the area of risk awareness. The second, fourth, and fifth works are on the subject of Probabilistically Constrained Belief Dependent POMDPs. Each work is discussed individually below.

Simplified Risk-aware Decision Making with Belief-dependent Rewards in Partially Observable Domains This work started with an attempt to quantify loss incurred by the simplification of belief-dependent POMDP. At first, we created the random variable we called `PLoss` [49]. This variable is accessible using only the unsimplified problem. In this work, we assumed that only the simplified problem is online accessible. We wanted to provide guarantees over the simplification impact without accessing the unsimplified problem and come up with stochastic bounds over the return and online accessible random variable which we called `PbLoss`. In this work, we substitute the return with simple stochastic bounds. Surprisingly these bounds induce deterministic bounds on the specific risk-aware operator, being Value at Risk (VaR). Using such a simplification we were able to speed up online risk-aware decision-making. In this work, we studied empirically that the information-theoretic reward pose a significant computational difficulty as the number of belief particles grows. Notably, while working on [50] we realized that [40] makes a decision using a single sample of observations episode. This means that here `PLoss` [49] can become a usable tool. It can express the probability of sampling a pair of observations per candidate action sequence and obtain loss from simplification larger than some value.

In addition in this work, we introduced the **extended setting**. In the extended setting, both, the belief update is stochastic as well as the reward operator on top of the given belief. Note that in [31] the belief update is also stochastic. This extension allowed us to formulate a simplification paradigm such as lowering the number of particles in belief representation and account for the simplification impact by providing guarantees.

We now summarize our key contributions as they appear in the paper [49]:

1. We extend ρ-POMDP to probabilistic ρ-POMDP ($\mathbb{P}\rho$-POMDP) by relaxing the assumption that the reward operator and the belief update are deterministic;
2. We introduce novel stochastic bounds on the return/reward and rigorously formulate the simplification framework on top of general objective operators and returns/rewards;
3. Using our formulations we present the simplification of risk averse decision making under uncertainty;
4. We present a novel objective utilizing joint distribution of the rewards corresponding to two candidate policies and describe a method to simplify such decision making while preserving *action consistency*;
5. We introduce the general concept of `PLoss` and provide its online description with `PbLoss` and utilize it to provide guarantees in terms of deterministic bounds;
6. Finally, we exemplify our framework on a particular simplification technique, which is reducing the number of samples within planning.

Simplified Continuous High Dimensional Belief Space Planning with Adaptive Probabilistic Belief-dependent Constraints In this work, we tackle the high-dimensional setting of POMDP with an unknown robot workspace. We were able to innovate the stopping exploration criterion and drastically speed up such decision-making. We introduced Probabilistic Belief-dependent Constraints (PC), which is two-staged and constrains what the robot believes. The first stage is the inner constraint. This constraint is belief-dependent and operates on a given history. We have two flavors of the inner constraint, namely cumulative and multiplicative. In this work, we focus on Information Gain and the cumulative form of the inner constraint. The second stage is the outer constraint, thresholding the probability that the inner constraint is satisfied, stemming from future histories simulated in planning session. We also analyze the averaged belief-dependent constraint, which surprisingly, due to computational burden, in the relevant literature, was approximated by Maximum Likely (ML) observation. Here, research was also carried out in the domain of Risk Awareness because we present an adaptive approach to maximize VaR. Additionally, in this paper, we presented a simplification mechanics for our PC. We did not simulate the simplification since the adaptive bounds over Information Gain in high dimensional spaces are missing.

Our contributions are fourfold. Below we list them down in the same order as they are presented in the manuscript.

1. First, we utilize our Probabilistically Constrained Belief-dependent POMDP in the context of an information-theoretic constraint. We focus on the IG, however, our theory supports any of the other belief-dependent operators, e.g. the difference between traces of covariance matrices of two consecutive-in-time beliefs. We analyze the Mutual Information (MI) constraint and ML observation approach versus our novel probabilistic constraint. Notably, we did not find any works shifting the MI from the reward operator to the constraint.
2. Second, we rigorously derive a theory of *simplification* in the constrained setting. We emphasize that the simplification paradigm has not been considered in this setting before. Given a monotonically converging to the belief-dependent constraint or/and reward bounds, depending on context, our approach can be simplified, gaining substantial speedup without any loss in performance quality.
3. Third, we present an algorithm to maximize Value at Risk adaptively utilizing the suggested theory. As we unveil in this paper, this enables the decision maker to save time by adaptively expanding the lowest required number of observations episodes without compromising the quality of the solution.
4. Fourth, we apply our technique to a high-dimensional BSP. In particular, our case studies are active SLAM and SD problems.

No Compromise in Solution Quality: Speeding Up Belief-dependent Continuous POMDPs via Adaptive Multilevel Simplification This work is a journal extension of the conference paper [44]. In the conference version of the paper adaptive bounds over a differential entropy estimator [5] were derived. This estimator leverages the given models, however, it assumes a low-dimensional setting. To rephrase that, it does not utilize the structure of a high-dimensional belief. Therefore, we did not use it in [50]. In addition in [44] the proposed bounds were utilized within the suggested novel simplification framework in the setting of a given belief tree. The main point of the simplification framework is that the adaptive bounds over the reward induce simplification levels. These levels then naturally transit to the objective function. When the objective bounds intervals corresponding to different policies overlap we need to **resimplify**. In [44] a single **resimplification strategy** was proposed. In our journal paper, we refine the general theory of the simplification. In the setting of a given belief tree, for instance, in the SS algorithm or when the predefined static action sequences are assumed,

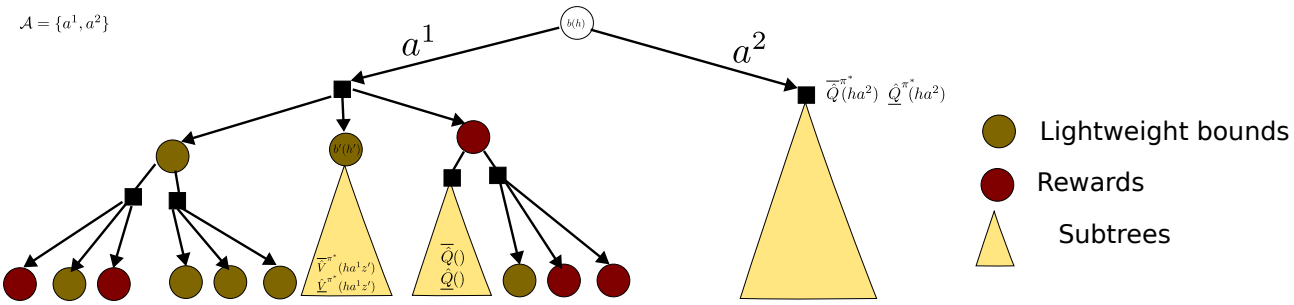


Figure 5.1: Illustration of the approach taken by [46] with a branching factor of three and the action space constituted by two actions. The bounds are substituted by rewards as the search progresses. Here, we visualize a full tree (action-wise) with both actions down the root.

we added an additional resimplification strategy which we call LAZY. We took inspiration from [46]. Let us briefly describe the approach taken by [46]. We illustrated the behavior in Fig. 5.1. In Fig. 5.1 four beliefs at the deepest level were switched from bounds to rewards. Suppose a two-steps ahead (Fig. 5.1) setting and suppose that we are given belief $b(h)$ with history h, reward for action a and one step ahead belief $b'(h')$ is bounded from below and above as such $\underline{\rho}(b,a,b') \leq \rho(b,a,b') \leq \overline{\rho}(b,a,b')$ and suppose we know that we are going to expand m children to belief action node ha. The objective for action a is bounded as

$$\frac{1}{m}\sum_{i=1}^{m}\Big(\underline{\rho}(b,a,b'^{,i})+\underline{\hat{V}}^{\pi}(b'^{,i})\Big) \leq \frac{1}{m}\sum_{i=1}^{m}\Big(\rho(b,a,b'^{,i})+\hat{V}^{\pi}(b'^{,i}(h'^{,i}))\Big) \leq \frac{1}{m}\sum_{i=1}^{m}\Big(\overline{\rho}(b,a,b'^{,i})+\overline{\hat{V}}^{\pi}(b'^{,i}(h'^{,i}))\Big) \tag{5.1}$$

where the value function estimator is

$$\hat{V}^{\pi}(b'^{,i}) = \frac{1}{m}\sum_{j=1}^{m}\rho(b'^{,i},\pi(h'^{,i}),b''^{,j}) \tag{5.2}$$

and the action-value function estimator

$$\hat{Q}^{\pi}(b(h)a) = \frac{1}{m}\sum_{i=1}^{m}\Big(\rho(b(h),a,b'^{,i})+\hat{V}^{\pi}(b'^{,i})\Big). \tag{5.3}$$

The bounds over the value function estimator can be of two flavors. The first is as follows. Define

$$a^* = \pi^*(b(h)) = \arg\max_{a\in\mathcal{A}} \hat{Q}^{\pi^*}(b(h)a).$$

We have the policy tree flavor if

$$\underbrace{\frac{1}{m}\sum_{j=1}^{m}\underline{\rho}(b'^{,i},a'^{,*},b''^{,j})}_{\underline{\hat{V}}^{\pi^*}(b'^{,i})} \leq \hat{V}^{\pi^*}(b'^{,i}) \leq \overbrace{\frac{1}{m}\sum_{j=1}^{m}\overline{\rho}(b'^{,i},a'^{,*},b''^{,j})}^{\overline{\hat{V}}^{\pi^*}(b'^{,i})}, \quad a'^{,*} = \pi^*(b'^{,i}). \tag{5.4}$$

Here, we need to know the best action for each $b'^{,i}$.The LAZY flavor reads

$$\underbrace{\max_{a\in\mathcal{A}}\frac{1}{m}\sum_{j=1}^{m}\underline{\rho}(b'^{,i},a,b''^{,j})}_{\underline{\hat{V}}^{\pi^*}(b'^{,i})} \leq \hat{V}^{\pi^*}(b'^{,i}) \leq \overbrace{\max_{a\in\mathcal{A}}\frac{1}{m}\sum_{j=1}^{m}\overline{\rho}(b'^{,i},a,b''^{,j})}^{\overline{\hat{V}}^{\pi^*}(b'^{,i})}. \tag{5.5}$$

Similarly we define bounds over the action value estimate

$$\underline{\hat{Q}}^{\pi}(b(h)a) \leq \hat{Q}^{\pi}(b(h)a) \leq \overline{\hat{Q}}^{\pi}(b(h)a). \tag{5.6}$$

The works [46], [38] select the best action with the **maximal upper bound** of the objective, and if the overlap of the objective bounds, corresponding to selected action and other actions, is present, they delve into the belief tree using forward search heuristic to select an action a and observation z' to bring the bounds of $Q^{\pi}(b(h)a)$ closer to each other and eventually eliminate the overlap. We, on the contrary, select the best action using the **largest lower bound** and use the resimplification strategy to delve into the belief tree and promote some of the reward bounds simplification levels. Let us go over these possibilities to select the best action:

- $a^* = \arg\max_{a\in\mathcal{A}} \overline{Q}^{\pi}(b(h)a)$ [46], [30], [38];
- $a^* = \arg\max_{a\in\mathcal{A}} \underline{Q}^{\pi}(b(h)a)$ [51].

In both cases above, the action consistent (same action is selected using the bounds instead of the objectives) decision is made with a^* if $\underline{Q}^{\pi}(b(h)a^*) \geq \max_{a\in\mathcal{A}\setminus\{a^*\}} \overline{Q}^{\pi}(b(h)a)$. Else, we shall resimplify and promote the simplification level. The work [46] substitutes some reward bounds by the reward itself, thereby tightening the objective bounds and effectively jumping to the maximal simplification level at these reward bounds. To resimplify, we shall delve into the tree, namely, select an action and belief (observation) to go down. Let us exemplify several possibilities to select an action leading to the subtree to resimplify

- $a^\dagger = \arg\max_{a\in\mathcal{A}} \overline{Q}^{\pi}(b(h)a)$ [46], [30], [38];
- $a^\dagger = \arg\max_{a\in\mathcal{A}} \overline{Q}^{\pi}(b(h)a) - \underline{Q}^{\pi}(b(h)a)$ [51].

For the next observation/belief both papers, [46] and [51] use the same heuristics

$$i = \arg\max_{i\in 1\ldots m} \overline{V}^{\pi^*}(b'^{,i}) - \underline{V}^{\pi^*}(b'^{,i}).$$

In this paper we also utilize the adaptive multilevel simplification within an MCTS approach.

To summarize, we list down the contributions of this work, in the order they are presented in the manuscript:

1. Building on **any** adaptive monotonically convergent bounds over belief-dependent reward, in this paper we present a **provable** general theory of adaptive multilevel simplification with deterministic performance guarantees.

2. For the case of a given belief tree as in Sparse Sampling, we develop two algorithms, Simplified Information Theoretic Belief Space Planning (SITH-BSP) and a faster variant, LAZY-SITH-BSP. Both are complementary to any POMDP solver that does not couple belief tree construction with an objective estimation while exhibiting a significant speedup in planning with a guaranteed same planning performance.

3. In the context of MCTS, we embed the theory of simplification into the PFT-DPW algorithm and introduce SITH-PFT. We provide stringent guarantees that exactly the same belief tree is constructed by SITH-PFT and PFT-DPW. We focus on a UCB exportation technique, but with minor adjustments, an MCTS with any exploration method will be suitable for acceleration.

4. We derive novel lightweight adaptive bounds on the differential entropy estimator of [5] and prove the bounds presented are monotonic and convergent. Moreover, these bounds can be incrementally tightened. We believe these bounds are of interest on their own. The bounds are calculated using the simplified belief (See Fig. 1) of [51]. We emphasize that any other bounds fulfilling assumptions declared in Section 3.3 of [51] can be utilized within our framework.

5. We present extensive simulations that exhibit a significant improvement in planning time without any sacrifice in planning performance.

To be precise, we explicitly clarify how this work differs from the conference version of this paper [44]. In this version, we extend the simplification framework to the rewards depending on a pair of consecutive-in-time beliefs, e.g., Information Gain, as opposed to the conference version where such an extension was only mentioned. In this version, we provide alternative proof of these bounds and prove that these reward bounds are monotonic. In the setting of a given belief tree we present an additional algorithm, that we call LAZY-BSP. This algorithm is faster than SITH-BSP suggested in [44]. Importantly, we extend our simplification framework to support also anytime MCTS planners. Additionally, we provide extensive performance evaluation of our methods in simulations.

Risk Aware Adaptive Belief-dependent Probabilistically Constrained and Chance Constrained Continuous Approximate POMDP Planning In this paper, we suggested our Probabilistically Constrained Belief-dependent POMDP. The rationale behind the name Belief-dependent POMDP is that we make all possible operators belief-dependent. This work researches safety and risk awareness using constraints.

In this work, we outline meaningful belief-dependent operators, not only ones related to safety, to serve as inner constraint in our formulation. We stay in the setting of the given belief tree in this paper. Specifically, we build upon SS and Open Loop (OL) planning. These are belief trees built with deterministic policy. Notably, SS algorithm builds a full belief tree in terms of actions on the way down the tree.

We rigorously analyze our approach versus popular chance constraints. Interestingly, in section 5.7 of this paper, we arrived at the conclusion that in the case of policies, the Chance Constraint can be viewed as a special

case of Probabilistic Constraint. In this special case, the belief-dependent operator looks into the future and calculates the Chance Constraint starting from each belief in the tree using in addition to the belief, the policy as input and the history corresponding to the belief to calculate the threshold per future history simulated in the planning session.

In the setting of static candidate action sequences, we compare Chance Constraint and Probabilistic Constraint. In the OL setting the Chance Constraint enforces the constraint on where the robot can be in the future, effectively assuming perfect observability MDP. On the contrary, we constrain what a robot believes about the POMDP state.

In both settings, Open Loop (OL) and Closed Loop (CL), we contribute algorithms adaptively evaluating the constraint. All our algorithms converge in probability as the number of sampled observations episodes and belief particles grows.

Below, we detail the contributions of this paper in the order they appear in the article.

- Firstly, in Section 3.1, we formulate a risk-averse belief-dependent Probabilistically Constrained continuous POMDP. Averaging the state-dependent reward/constraint to obtain the belief-dependent reward/constraint is a severe hindrance that we relax. We are unaware of prior works addressing POMDP with risk-averse belief-dependent constraints. In particular, our probabilistic belief-dependent constraint supports risk-averse operators, such as CVaR, and leads to a novel safety constraint formulation.

- Secondly, on top of our probabilistic formulation, in Section 4.1.3, we contribute a novel, efficient actions-pruning mechanism. SOTA pruning technique proposed by [36] constitutes only a necessary condition such that it is possible that after pruning, actions violating the CC are kept in the belief tree. Therefore, the feasibility of CC has to still be inspected for each not-pruned action. On the contrary, our pruning condition is necessary and sufficient. No additional checks are needed after the pruning of the belief tree is complete.

- In Section 4.2, we contribute algorithms for online solutions of Probabilistically constrained belief-dependent POMDP in continuous domains. Our algorithms are adaptive given a budget of observation episodes laces (Fig. 1 of the paper) and beliefs within the lace to expand in the belief tree. In other words, we provide a way to guide the belief tree construction while planning. Our framework is universal for challenging continuous domains and can be applied in nonparametric and parametric settings. We innovate algorithms for CL setting with policies as well as for OL setting with candidate action sequences.

- Another contribution on our end is a rigorous analysis of our probabilistic formulation versus chance-constrained in Section 5. Despite recent algorithmic developments [36], there has been relatively little effort devoted to the theoretical aspects of Chance-constrained continuous belief-dependent POMDP. Surprisingly, in Section 5.7 we obtained that in the CL setting, CC is a specific case of our PC when the belief-dependent operator is CC itself. It shall be noted as a contribution that we spotted the fact that belief shall be defined differently within CC. To the best of our knowledge, no paper addresses this discrepancy.

- We uplift a chance-constrained solver to continuous domains in terms of states and observations and general belief-dependent rewards through Importance Sampling (IS) in Section 6.

- In an OL setting, we contribute an adaptive, in terms of trajectories and states, algorithm (Alg. 6) for chance-constrained continuous ρ-POMDP. This algorithm can be used with exceptionally long horizons and a high dimensional setting.

Anytime Probabilistically Constrained Provably Convergent Online Belief Space Planning This work is a continuation of the previous paper. Here we embed our Probabilistic safety into MCTS. Our contributions in this work are as follows. We assure that the constraint is fulfilled not only at the convergence of MCTS as in the case of [20] but at every moment in time. Our search tree expanded by MCTS always consists of solely the safe actions. In addition, in this paper we constraint the propagated beliefs along the posteriors as in our previous work. The polynomial variant of our approach is provably convergent. In particular, similar to [41], we utilize the proof by [4]. We show that our modifications do not break the proof.

Below we list down our contributions in the same order as they appear in the manuscript.

1. By directly constraining the problem space and not the dual space we present an anytime MCTS based algorithm for safe online decision making with safety governed by Probabilistic Constraint (PC). Our approach enjoys anytime safety guarantees with respect to the belief-tree expanded so far and works in continuous state, action and observation spaces. When stopped anytime, the action returned can be considered as the best safe action under the safe future policy (tree policy) expanded so far. Our search tree consists of **solely** the safe actions. We prove convergence in probability of our approach.

2. Another one of our contributions is the constraining the beliefs with incorporated outcome uncertainty stemming from an action performed by the robot and without incorporating the received observation.

3. We also spot a problem happening in duality based approach arising from averaging unsafe actions in MCTS phase. Therefore, an additional contribution of ours is an analysis of this phenomenon.

Bibliography

[1] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato. FIRM: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *Intl. J. of Robotics Research*, 33(2):268–304, 2014. pages 9

[2] Mauricio Araya, Olivier Buffet, Vincent Thomas, and Françcois Charpillet. A pomdp extension with belief-dependent rewards. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 64–72, 2010. pages 8

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002. pages 7

[4] David Auger, Adrien Couetoux, and Olivier Teytaud. Continuous upper confidence trees with polynomial exploration–consistency. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part I 13*, pages 194–209. Springer, 2013. pages 175

[5] Y. Boers, H. Driessen, A. Bagchi, and P. Mandal. Particle filter based entropy. In *2010 13th International Conference on Information Fusion*, pages 1–8, 2010. pages 9, 172, 174

[6] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 723–730, 2011. pages 5, 9

[7] Louis Dressel and Mykel J. Kochenderfer. Efficient decision-theoretic target localization. In Laura Barbulescu, Jeremy Frank, Mausam, and Stephen F. Smith, editors, *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS 2017, Pittsburgh, Pennsylvania, USA, June 18-23, 2017*, pages 70–78. AAAI Press, 2017. pages 8

[8] Khen Elimelech and Vadim Indelman. Simplified decision making in the belief space using belief sparsification. *Intl. J. of Robotics Research*, 41(5):470–496, 2022. pages 9

[9] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006. pages 9, 10

[10] E. I. Farhi and V. Indelman. ix-bsp: Belief space planning through incremental expectation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2019. pages 9

[11] Mathieu Fehr, Olivier Buffet, Vincent Thomas, and Jilles Dibangoye. rho-pomdps have lipschitz-continuous epsilon-optimal value functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6933–6943. Curran Associates, Inc., 2018. pages 8

[12] Johannes Fischer and Omer Sahin Tas. Information particle filter tree: An online algorithm for pomdps with belief-based rewards on continuous domains. In *Intl. Conf. on Machine Learning (ICML)*, Vienna, Austria, 2020. pages 8

[13] Neha P Garg, David Hsu, and Wee Sun Lee. Despot-α: Online pomdp planning with large state and observation spaces. In *Robotics: Science and Systems (RSS)*, 2019. pages 5, 6, 7, 8

[14] Astghik Hakobyan, Gyeong Chan Kim, and Insoon Yang. Risk-aware motion planning and control using cvar-constrained optimization. *IEEE Robotics and Automation Letters*, 4(4):3924–3931, 2019. pages 5

[15] Milos Hauskrecht. Value-function approximations for partially observable markov decision processes. *Journal of artificial intelligence research*, 13:33–94, 2000. pages 7

[16] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994. pages 8

[17] Marcus Hoerger, Hanna Kurniawati, and Alberto Elfes. Multilevel monte-carlo for solving pomdps online. In *Proc. International Symposium on Robotics Research (ISRR)*, 2019. pages 9

[18] V. Indelman. No correlations involved: Decision making under uncertainty in a conservative sparse information space. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):407–414, 2016. pages 9

[19] Masoumeh T Izadi and Doina Precup. A planning algorithm for predictive state representations. In *IJCAI*, pages 1520–1521. Citeseer, 2003. pages 4

[20] Arec Jamgochian, Anthony Corso, and Mykel J Kochenderfer. Online planning for constrained pomdps with continuous spaces through dual ascent. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 33, pages 198–202, 2023. pages 175

[21] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998. pages 5

[22] Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine learning*, 49(2):193–208, 2002. pages 7, 10

[23] Sung-Kyun Kim, Rohan Thakker, and Ali-Akbar Agha-Mohammadi. Bi-directional value learning for risk-aware planning under uncertainty. *IEEE Robotics and Automation Letters*, 4(3):2493–2500, 2019. pages 9

[24] A. Kitanov and V. Indelman. Topological information-theoretic belief space planning with optimality guarantees. *arXiv preprint arXiv:1903.00927*, 3 2019. pages 9

[25] M. Kochenderfer, T. Wheeler, and K. Wray. *Algorithms for Decision Making*. MIT Press, 2022. pages 6, 7, 8

[26] Mykel J. Kochenderfer. *Decision making under uncertainty: theory and application*. MIT press, 2015. pages 6, 7

[27] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006. pages 7

[28] D. Kopitkov and V. Indelman. No belief propagation required: Belief space planning in high-dimensional state spaces via factor graphs, matrix determinant lemma and re-use of calculation. *Intl. J. of Robotics Research*, 36(10):1088–1130, August 2017. pages 9

[29] Dmitry Kopitkov and Vadim Indelman. General purpose incremental covariance update and efficient belief space planning via factor-graph propagation action tree. *Intl. J. of Robotics Research*, 38(14):1644–1673, 2019. pages 9

[30] H. Kurniawati, D. Hsu, and W. S. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems (RSS)*, 2008. pages 5, 6, 174

[31] Michael H Lim, Tyler J Becker, Mykel J Kochenderfer, Claire J Tomlin, and Zachary N Sunberg. Optimality guarantees for particle belief approximation of pomdps. *Journal of Artificial Intelligence Research*, 77:1591–1636, 2023. pages 171

[32] Michael Littman and Richard S Sutton. Predictive representations of state. *Advances in neural information processing systems*, 14, 2001. pages 4

[33] Nils J Nilsson. *Principles of artificial intelligence*. Morgan Kaufmann, 1982. pages 9

[34] J. Pineau, G. J. Gordon, and S. Thrun. Anytime point-based approximations for large POMDPs. *J. of Artificial Intelligence Research*, 27:335–380, 2006. pages 6

[35] S. Prentice and N. Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *Intl. J. of Robotics Research*, 28(11-12):1448–1465, 2009. pages 9

[36] Pedro Santana, Sylvie Thiébaux, and Brian Williams. Rao*: An algorithm for chance-constrained pomdp's. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. pages 5, 9, 175

[37] David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2164–2172, 2010. pages 5, 8

[38] T. Smith and R. Simmons. Heuristic search value iteration for pomdps. In *Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 520–527, 2004. pages 4, 6, 7, 8, 10, 173, 174

[39] T. Smith and R. Simmons. Point-based pomdp algorithms: Improved analysis and implementation. In *Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 542–547, 2005. pages 7

[40] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Robotics: Science and Systems (RSS)*, pages 65–72, 2005. pages 171

[41] Zachary Sunberg and Mykel Kochenderfer. Online algorithms for pomdps with continuous state, action, and observation spaces. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28, 2018. pages 5, 6, 8, 175

[42] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. pages 5

[43] Ori Sztyglic and Vadim Indelman. Online pomdp planning via simplification. *arXiv preprint arXiv:2105.05296*, 2021. pages 9

[44] Ori Sztyglic and Vadim Indelman. Speeding up online pomdp planning via simplification. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022. pages 9, 172, 174

[45] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT press, Cambridge, MA, 2005. pages 5, 7

[46] Thomas Walsh, Sergiu Goschin, and Michael Littman. Integrating sample-based planning and model-based reinforcement learning. In *AAAI Conf. on Artificial Intelligence*, volume 24, 2010. pages , 10, 173, 174

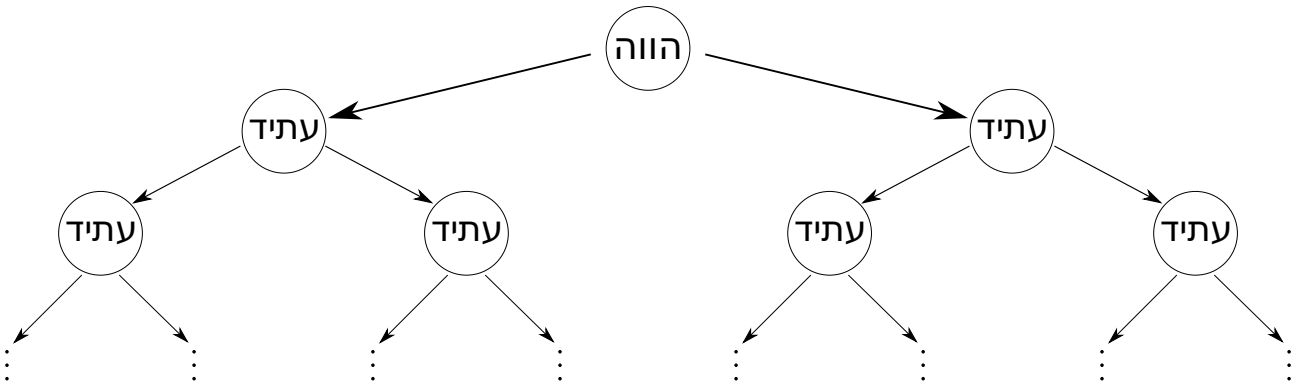
[47] Yue Wang, Abdullah Al Redwan Newaz, Juan David Hernández, Swarat Chaudhuri, and Lydia E Kavraki. Online partial conditional plan synthesis for pomdps with safe-reachability objectives: Methods and experiments [-5pt]. *IEEE Transactions on Automation Science and Engineering*, 2021. pages 9

[48] Nan Ye, Adhiraj Somani, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. *JAIR*, 58:231–266, 2017. pages 5, 8

[49] A. Zhitnikov and V. Indelman. Simplified risk aware decision making with belief dependent rewards in partially observable domains. *Artificial Intelligence, Special Issue on "Risk-Aware Autonomous Systems: Theory and Practice"*, 2022. pages 171

[50] Andrey Zhitnikov and Vadim Indelman. Simplified continuous high dimensional belief space planning with adaptive probabilistic belief-dependent constraints. *IEEE Trans. Robotics*, 2024. pages 171, 172

[51] Andrey Zhitnikov, Ori Sztyglic, and Vadim Indelman. No compromise in solution quality: Speeding up belief-dependent continuous pomdps via adaptive multilevel simplification. *Intl. J. of Robotics Research*, 2024. pages 174



איור 1: עץ החלטה

תקציר

מוטיבציה הביקוש למערכות אוטונומיות גדל באופן דרסטי במגוון תחומים שונים כגון ניווט אוטונומי, פעולות של זרוע רובוטית, רובוט אנושי ובינה מלאכותית. אחד היישומים הרצויים ביותר הוא כלי רכב אוטונומי. לדוגמה, כלי טיס בלתי מאויש לפיקוח ובדיקה, רובוטים חקלאיים, וכו'. היישומים נמצאים בכל מקום. ראיה ממוחשבת היא מקור משמעותי למידע וידע על העולם בו המערכת האוטונומית פועלת. כיום כלי חישה הולכים וקטנים ואיכות התמונה הולכת וגדלה. לכן, טבעי להשתמש במקור זה באופן דומה לבן אנוש. קבלת החלטות על ידי סוכן מהווה גרעין בבעיות ויישומים המתוארים לעיל. על מנת לבחור פעולה אופטימלית הסוכן צריך לפתור בזמן אמת בעיית אופטימיזציה כבדה מאוד חישובית. יחד עם זאת בהרבה מקרים מספיק לפתור בעיה מקורבת שקלה יותר חישובית מבלי לפגוע בביצועים. לדוגמה במקרה של רובוט חקלאי, כביש יכול להיות רחב וריק כך שבאם והרובוט יעבור יותר קרוב לקצה הכביש או יותר קרוב למרכז אין הרבה משמעות. מטרת מחקר זה הינה לייעל את תהליך קבלת ההחלטות על ידי הפשטה של הבעיה ולכמת הפסד בביצועים, אם קיים.

פעילות בסביבה לא ידועה הקלט לבעיית קבלת החלטה אופטימלית הוא מידע. כאשר נהג מפעיל רכב הוא משתמש בידע מקדים לגבי מפה של המקום, ראיה ושמיעה שלו על מנת להסיק את מיקומו ולבחור פעולה אופטימלית שמקרבת אותו ליעד. בינה מלאכותית מנסה לדמות את תהליך זה. לפני תחילת פעולתו, לרובוט מספקים בצורת פילוג הסתברותי מידע מקדים על פרמטרים שהוא יצטרך. בזמן הפעולה בנגישות הרובוט מידע מקדים, היסטוריה של תצפיות שהוא קיבל ופעולות שעשה. על סמך קלט זה הוא צריך לבצע הסקה הסתברותית על פרמטרים הנדרשים לו כגון מיקומו ומיקום של חפצים בעולם. משמע להסיק התפלגות הסתברותית של פרמטרים האלה בהינתן כל מה שנתון לו. התפלגות הסתברותית זאת נקראת אמונה. אמונה היא פילוג הסתברותי שנושא את כל האינפורמציה שנגישה ונדרשת בצורה קומפקטית. הרובוט מתחזק אמונה ומעדכן אותה עם קבלת תצפית חדשה וביצוע פעולה שבחר. בפרט, אמונה נותנת גישה לאי ודאות.

קבלת החלטות תחת חוסר ודאות - במרחב האמונה המערכות העקרוניות של הרובוט הן מערכת חישה כגון מצלמה או מכ"ם פולט אור ומערכת התנועה. כאשר הרובוט פועל בשטח, הוא חש את העולם עולם בעזרת חיישנים שזמינים לו ומבצע תכנון תנועתו בצורה מחזורית. לאחר חישוב פעולה מיטבית הוא מבצע אותה, מקבל תצפית חדשה ומעדכן את האמונה. האמונה משמשת כקלט לשלב ביצוע תכנון בזמן האמת. גורם הכרחי של תכנון מוצלח זה כמות הצעדים קדימה בעתיד (הוריזון) שסוכן מתחשב בו במסגרת תהליך קבלת החלטות. ככל שההוריזון גדל, התכנון נעשה יותר מדויק ואיכותי. נדגים את בעיית התכנון במרחב אמונה על ידי בעיית ניווט ליעד בסביבה לא ידועה. על מנת להגיע ליעד, הרובוט צריך לאסוף מידע על סביבתו ולבצע תהליך תכנון על מנת לבחור פעולה אופטימלית, שוב לאסוף מידע ושוב לבצע תהליך תכנון. ככה הוא ממשיך בצורה מחזורית עד להגעתו ליעד הנדרש. על מנת לתכנן, הרובוט מסמלץ התממשויות אפשרויות של העתיד. כלומר תצפיות שהוא יכול לקבל בעתיד כתוצאה מפעולה מסוימת. במילים אחרות הרובוט חושב על סדרת פעולות מועמדת להיבחר כאופטימלית, מסמלץ התממשות תצפיות שהוא יכול לקבל בזמן ביצוע סדרת פעולות זאת ומעדכן את האמונות העתידיות. ככה הוא מקבל סדרת אמונות בגודל של הוריזון, על כל אחד הוא מחשב תגמול וסוכם את התגמולים. התגמול המצטבר מהווה התממשות אחת של העתיד. בדוגמה של נווט, התגמול מורכב ממרחק ממוצע מהיעד ומדד אי וודאות של המצב של הרובוט. לא מספיק למזער מרחק ממוצע מהרובוט ליעד, צריך גם שאי וודאות של מצב הרובוט תהייה נמוכה. כמות התגמולים המצטברים שווה לכמות התצפיות אפשרויות שרובוט יכול לקבל.

משמע מתקבלת התפתחות אקספוננציאלית עם ההוריזון כפי שמודגם באיור הבא 1 . הסיבוכיות החישובית נעשית עוד יותר כבדה כאשר המצב שלגביו מתחזקים אמונה הינו רב ממדי.

בשל כך לא ניתן לפתור בעיית תכנון במרחב האמונה בזמן אמת באופן מדויק. נדרשת **הפשטה**. כלומר החלפה של בעיית תכנון המדויק בבעיה אחרת הניתנת לפתרון בזמן האמת. לדוגמה הפשטה מקובלת היא לקחת בחשבון רק קבלת סדרת התצפיות הכי מסתברת כתוצאה מבחירת סדרת פעולות מועמדת לאופטימלית. מחקר זה עוסק בניתוח ותכנון הפשטות לבעיות תכנון תחת אי ודאות בתרחישים מאתגרים, כגון התפלגות אמונה מורכבת ופונקציות תגמול כלליות. החזון שלנו שרובוטים יהיו בכל מקום, יבצעו פעולות רבות בזריזות וקלילות רבה.

המחקר נעשה בהנחיית פרופסור חבר ואדים אינדלמן

במסגרת התוכנית הבין-יחידתית למערכות אוטונומיות ורובוטיקה

מחבר חיבור זה מצהיר כי המחקר, כולל איסוף הנתונים, עיבודם והצגתם, התייחסות והשוואה למחקרים קודמים וכו׳, נעשה כולו בצורה ישרה, כמצופה ממחקר מדעי המבוצע לפי אמות המידה האתיות של העולם האקדמי. כמו כן, הדיווח על המחקר ותוצאותיו בחיבור זה נעשה בצורה ישרה ומלאה, לפי אותן אמות מידה.

מחבר חיבור זה קיבל מלגות הבאות:

- מלגת זף
- מלגת ג׳ייקובס
- מלגת מצויינים בממון הפקולטה

תודות

אני רוצה להודות למנחה שלי פרופ. חבר ואדים אינדלמן על הנחייתו ותמיכתו הבלתי פוסקת.

בנוסף אני רוצה להודות לאמי אולגה זיטניקוב עבור תמיכה מתמשכת לאורך עבודתי לקראת תואר דוקטור.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי

הפשטה לקבלת החלטות יעילה תחת חוסר ודאות עם התפלגויות כלליות

חיבור על מחקר

לשם מילוי חלקי של דרישות לקבלת התואר דוקטור לפילוסופיה

אנדריי זיטניקוב

הוגש לסנט הטכניון – מכון טכנולוגי לישראל

סיוון התשפ"ד חיפה יוני 2024

הפשטה לקבלת החלטות יעילה תחת חוסר ודאות עם התפלגויות כלליות

אנדריי זיטניקוב