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,

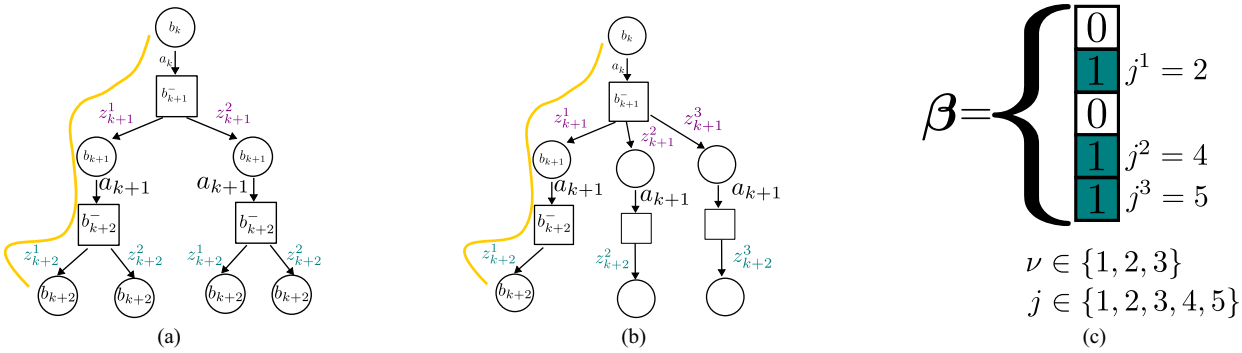


Fig. 1. Possible belief trees in continuous setting given $\boldsymbol{\beta}_{k+1:k+2}$. By **purple** and **teal** colors, we denote possibly different dimensionality of the observation as explained in Section II-B. Thick yellow lace illustrates the observation sequence $\boldsymbol{z}_{k+1:k+2}$ (Section 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:

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

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

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

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

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

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 $\mathbb{E}_{\boldsymbol{z}_{(k+1)+}|b_k,a_{k+},\boldsymbol{\beta}_{(k+1)+}}[-\mathrm{h}(b_{k+L})+\mathrm{h}(b_k)] \leq 0$. 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

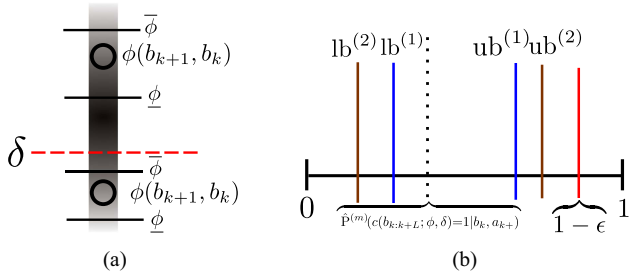


Fig. 2. (a) Conceptual visualization of our *simplification* approach (Section III-F). For clarity we show a myopic setting. Gradient displays the PDF, i.e., a larger number of samples lands in the area of greater intensity. Using the bounds, we want to assess whether the fraction of the sampled observation 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:

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

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

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

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

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

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

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

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 $0 \le \mathrm{lb}$ from below and the upper bound is bounded by one $\mathrm{ub} \le 1$ 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 $\mathrm{P}(c(b_{k:k+L};\phi,\delta)=1|b_k,a_{k+})$. 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

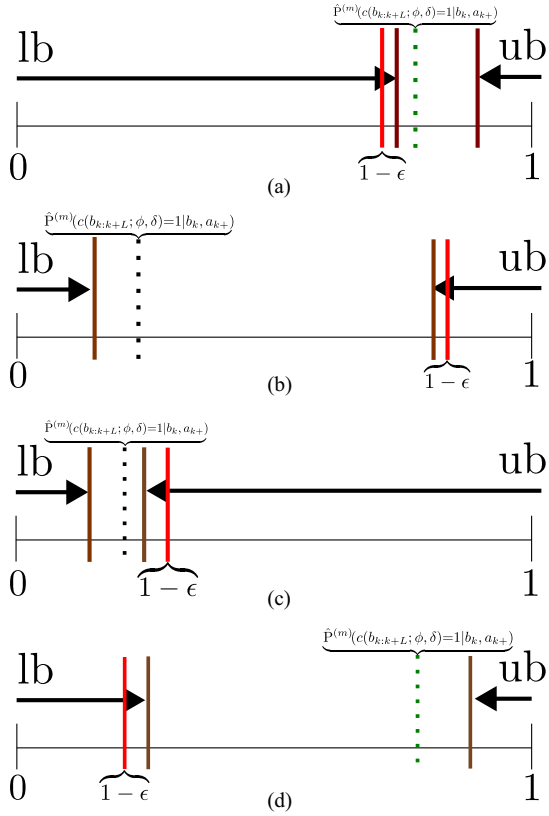


Fig. 3. Visualization of adaptation from Section III-G. Note, in all scenarios the value of dashed line is *unknown*. Red line represents the confidence level $1-\epsilon$ to be satisfied with probabilistic constraint. (a) Conceptual illustration of a challenging scenario. To *accept* such an action 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.

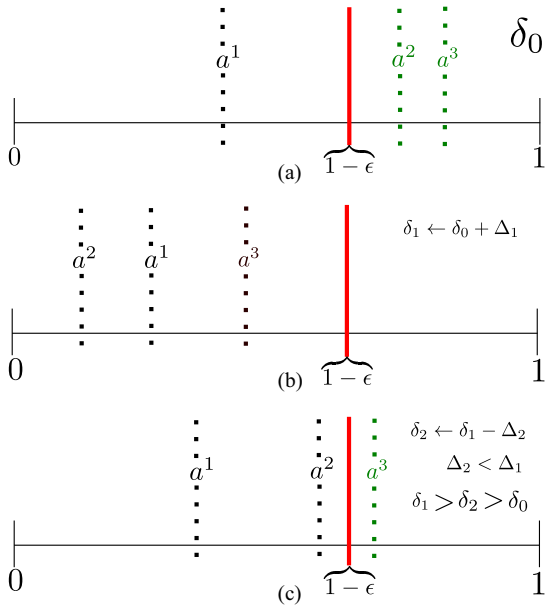


Fig. 4. Visualization of Algorithm 3. We never increase the step size. Therefore, as we see, each candidate action sequence in the bottom visualization (**c**) is shifted to the left relative to the situation displayed in the top (**a**). The action sequence a^1 can be safely discarded in the top illustration (**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])

$$\sum_{\boldsymbol{\beta}_{1:k}} \mathbb{P}(\boldsymbol{x}_k|b_0, a_{0:k-1}, \boldsymbol{z}_{1:k}, \boldsymbol{\beta}_{1:k}) \frac{\text{P}(\boldsymbol{\beta}_{1:k}|b_0, a_{0:k-1}, \boldsymbol{z}_{1:k})}{\sum_{\boldsymbol{\beta}_{1:k}} \text{P}(\boldsymbol{\beta}_{1:k}|b_0, a_{0:k-1}, \boldsymbol{z}_{1:k})} \tag{41}$$

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

$$b_k(\boldsymbol{x}_k) \propto b_0(x_0) \prod_{i=1}^{k} \left(\mathbb{P}_T\left(x_i|x_{i-1}, a_{i-1}\right) \cdot \right.$$
$$\left. \text{P}_{\boldsymbol{\beta}}\left(\boldsymbol{\beta}_i|x_i, \{\ell^j\}_{j=1}^{M(i)}\right) \prod_{\nu_i=1}^{n(\boldsymbol{\beta}_i)} \mathbb{P}_Z\left(z_i^{\nu_i}|x_i, \ell^{j^{\nu_i}}\right) \right). \tag{43}$$

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 $\prod_{i=1}^{k} \text{P}_{\boldsymbol{\beta}}(\boldsymbol{\beta}_i|x_i, \{\ell^j\}_{j=1}^{M(i)})$ terms and remain with

$$b_k(\boldsymbol{x}_k) \propto b_0(x_0) \prod_{i=1}^{k} \left(\mathbb{P}_T\left(x_i|x_{i-1}, a_{i-1}\right) \prod_{\nu_i=1}^{n(\boldsymbol{\beta}_i)} \mathbb{P}_Z\left(z_i^{\nu_i}|x_i, \ell^{j^{\nu_i}}\right) \right). \tag{46}$$

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.

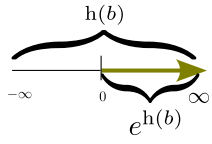


Fig. 5. New Information measure.

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), $\delta^{\max} \triangleq 1$.

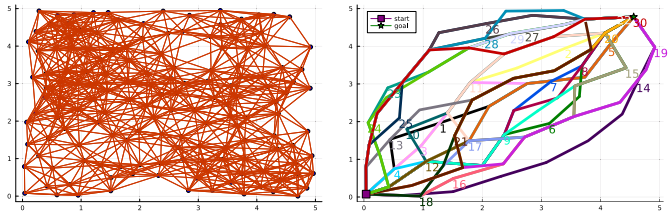
In practice our approach (Algorithm 3) requires $\delta^{\min}$ and $\delta^{\max}$ for the whole return $s(b_{k:k+L}; \cdot)$ for any L. With our definition (56) this is not a problem since we obtain telescopic series. If one uses (57) or deals with infinite horizons approximated by L steps ahead, where $\mathrm{IG}(b, b') = \gamma \mathrm{I}(b') - \mathrm{I}(b)$ [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.

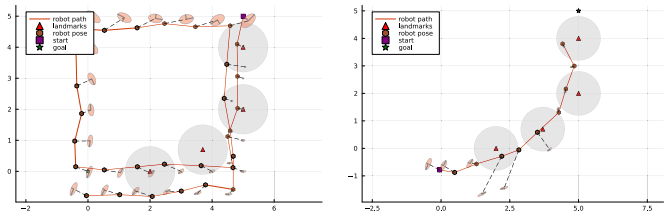
The robot constructs a belief tree of the form presented in Fig. 1(a) for each candidate path within planning session. With each promotion of the depth of the belief tree, we reduce the number of observations at each belief node by factor two, up to a possible single observation at the lowest levels. 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

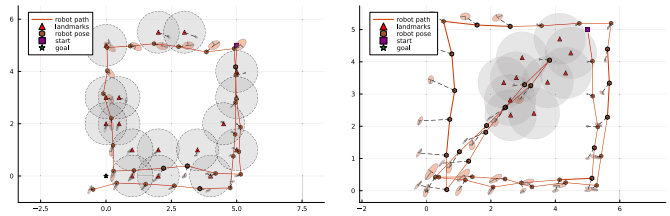
	ϵ	δ	$\mathcal{P}^*$	$\hat{\mathcal{U}}^*_{(m)}$	№discarded paths	time [s] $\pm$ std	speedup (59)	laces frac. (60)	total laces	№paths	№land.
Alg. 2	-	0.0	14	$36.98 \cdot 10^{-5}$	-	1171.21 ± 74.48		0	9000/9000	30	4
Alg. 1	0.023	0.0	no feasible	-	30	77.67 ± 4.01	**0.934**	0.95	459/9000		
Alg. 1	0.3	0.0	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)

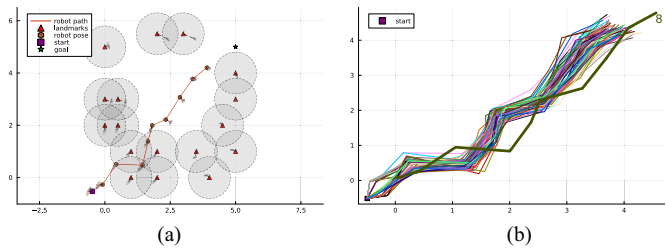
	ϵ	$\mathcal{P}^*$	δ^*	time [s] $\pm$ std	speedup (59)	laces frac. (60)	laces evaluations	№paths with $\widehat{\mathrm{VaR}}_\epsilon^{(m)} > \delta^{\min}$	№paths	№ landmarks
Alg. 4	0.3	8	$1.86 \cdot 10^{-5}$	511.03 ± 22.52	-	0	1920/1920	21	30	17
Alg. 3		8	$1.87 \cdot 10^{-5}$	$\mathbf{349.85 \pm 7.36}$	**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.

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.

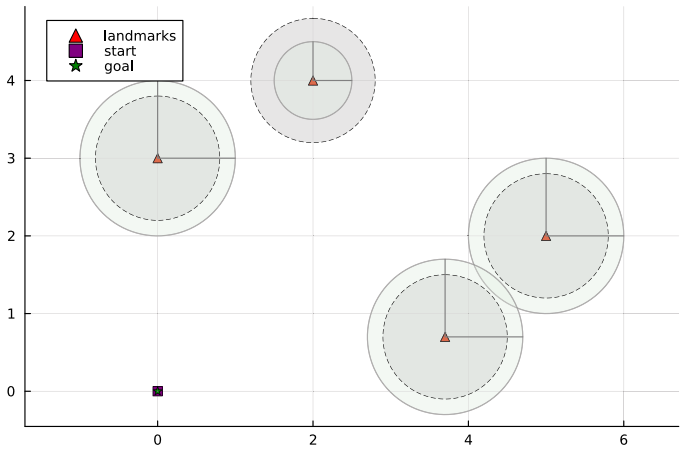


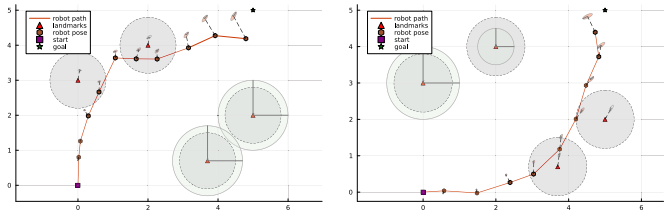
Fig. 10. Visualization of the scenario for verifying that ML observation assumption can be destructive. Robot starts to plan from b_0. Each landmark has prior shown by light green circle and the visibility radius shown by gray circle with dashed line.

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

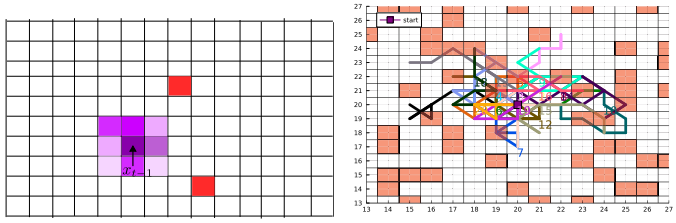
	ϵ	time [s] $\pm$ std	speedup (59)	laces frac. (60)	laces evaluations	L	m	σ^2	no sensors cells	№candidate paths
Alg. 4	0.1	1948.55 ± 259.48	-	0	75000/75000	15	150	0.1	750	20
Alg. 3		$\mathbf{1299.77 \pm 100.82}$	**0.33**	**0.39**	**45761/75000**					
Alg. 4	0.3	1146.21 ± 117.05	-	0	40000/40000	10	200	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)

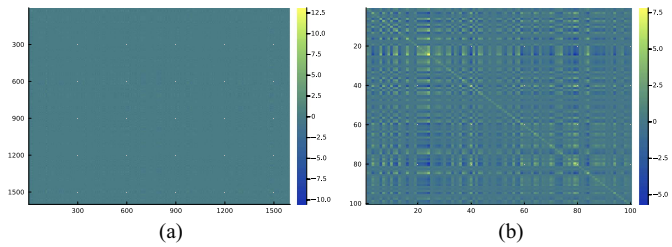
Fig. 12. (a) Conceptual illustration of our scenario and the transition model structure for SD problem. In time index $t-1$ the robot take an action $\uparrow$ and by time t, the robot can be in one of the purple cells. The intensity designate the chance to be there. The red cells are not suitable for 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.

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



(a) (b)

Fig. 13. $\boldsymbol{\xi} \in \mathbb{R}^{1600}$ (a) Covariance of $b_0(\boldsymbol{\xi})$; (b) Zoom in.

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 $\boldsymbol{x}_{t+1}$

$$\mathbb{P}(\boldsymbol{z}_{t+1}|b_t, a_t, \boldsymbol{\beta}_{t+1})\mathrm{P}(\boldsymbol{\beta}_{t+1}|b_t, a_t) = \int_{\boldsymbol{x}_{t+1}} \mathbb{P}(\boldsymbol{z}_{t+1}|b_t, a_t, \boldsymbol{\beta}_{t+1}, \boldsymbol{x}_{t+1}) \cdot \mathbb{P}(\boldsymbol{x}_{t+1}|b_t, a_t, \boldsymbol{\beta}_{t+1})\mathrm{P}(\boldsymbol{\beta}_{t+1}|b_t, a_t)\mathrm{d}\boldsymbol{x}_{t+1} \tag{62}$$

$$= \int_{\boldsymbol{x}_{t+1}} \mathbb{P}(\boldsymbol{z}_{t+1}|b_t, a_t, \boldsymbol{\beta}_{t+1}, \boldsymbol{x}_{t+1}) \cdot \mathbb{P}(\boldsymbol{x}_{t+1}|b_t, a_t)\mathrm{P}_{\boldsymbol{\beta}}(\boldsymbol{\beta}_{t+1}|\boldsymbol{x}_{t+1})\mathrm{d}\boldsymbol{x}_{t+1}. \tag{63}$$

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 $z^l_{k+1:k+L}$:

$$\underline{c}(b^l_{k:k+L}; \underline{\phi}, \delta) \le c(b^l_{k:k+L}; \phi, \delta) \le \overline{c}(b^l_{k:k+L}; \overline{\phi}, \delta). \tag{64}$$

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

$$\underline{c}(b^l_{k:k+L}; \underline{\phi}, \delta) - c(b^l_{k:k+L}; \phi, \delta) > 0. \tag{65}$$

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.

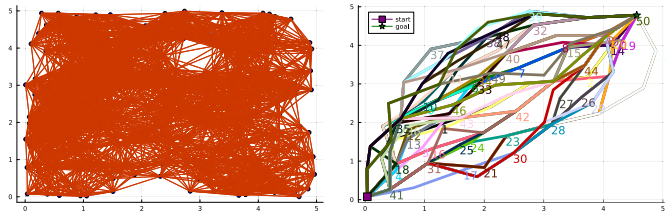
1) SLAM: First, we sample the last pose and the landmarks from the corresponding marginal of the belief. Our belief is Gaussian, thus, we just pull the appropriate portion of the covariance matrix and the mean value followed by sampling from $(x_{t+1}, \{\ell^j\}_{j=1}^{M(k)})^o \sim \mathbb{P}(x_{t+1}, \{\ell^j\}_{j=1}^{M(k)}|b_t, a_t)$. 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.