Measurement Simplification in ρ-POMDP with Performance Guarantees

Tom Yotam

Measurement Simplification in ρ-POMDP with Performance Guarantees

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Applied Mathematics

Tom Yotam

Submitted to the Senate
of the Technion — Israel Institute of Technology
Tamuz 5783 Haifa June 2023

This research was carried out under the supervision of Associate Professor Vadim Indelman, in the Faculty of Applied Mathematics.

Some results in this thesis have been published or submitted as articles by the author and research collaborators in conferences and journals during the course of the author's research period, the most up-to-date versions of which being:

T. Yotam and V. Indelman. Measurement Simplification in BSP with Performance Guarantees. In *Israel Conference Robotics (ICR)*, 2023.

T. Yotam and V. Indelman. Measurement Simplification in ρ-POMDP with Performance Guarantees. In *IEEE Transactions on Robotics (T-RO)*, 2023. To be submitted.

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

Acknowledgements

I would like to thank my advisor, Assoc. Prof. Vadim Indelman, for countless hours of fruitful discussions, and for granting me freedom to explore while leading me in the right direction. I would also like to thank my family and my partner Liat, for their continuous support throughout this journey.

The generous financial help of the Technion is gratefully acknowledged.

Contents

List of Figures

List of Tables

Abstract

Partially Observable Markov Decision Process (POMDP) planning is one of the fundamental problems an agent must solve when operating with imperfect information about its environment, dynamics, and measurements. The agent maintains a probability density function (belief) over its true state which is unknown. A motion model describes to probability of moving from one state to another and an observation model describes the probability of obtaining measurements from a given state. In planning, the agent is set to find the optimal action policy such that a certain reward function is maximized.

The rapid exponential growth of posterior beliefs makes the planning problem NP-complete. We present a novel approach of simplification to the POMDP problem, specifically by partitioning the underlying observation space. Using the partitioned observation space, we show the relation to the original problem by deriving analytical bounds on the expected entropy that hold for all families of belief distributions. We show that these bounds are adaptive and that they converge to the original solution. We show one possible realization of this general framework, specifically for Gaussian distributions, that speeds up the planning time by a factor of 4 with the same optimal action, or with bounds over the loss.

We later extend this concept to introduce a partition hierarchy, where each partitioned set is divided further into children sets. This hierarchy is encoded in a partition tree, which opens the door for a family of efficient implementations for the expected reward bounds.

Finally, we demonstrate this new concept in an active, high-dimensional, Simultaneous Localization and Mapping (SLAM) scenario, both simulated and real. We show substantially speed-up compared to other State of the Art (SOTA) methods.

Abbreviations and Notations

BSP	:	Belief Space Planning
POMDP	:	Partially Observable Markov Decision Process
$\mathcal{X}$	:	State space
$\mathcal{A}$	:	Action space
$\mathcal{T}$	:	Probabilistic transition model
$r(\cdot)$	:	Reward function (over state or belief)
$\rho(\cdot)$	:	Reward function over the belief
$\mathcal{Z}$	:	Observation space
$\mathcal{O}$	:	Probabilistic observation model
x_k	:	State at time index k
a_k	:	Action at time index k
z_k	:	Observation at time index k
h_k	:	History of all actions observations and prior belief up to time k
h_k^-	:	Similar to h_k but not including observations from time step k
$b[x]$	:	Belief - Posterior distribution over the state x
b_k	:	Belief at time index k given h_k
b_k^-	:	Belief at time index k given h_k^-
b_0	:	Initial/Prior belief (time index $k = 0$)
η	:	Normalization term
$\square_{k:k+L}$	:	Sequence from time index k to time index $k + L$
π_{k+}	:	Policy sequence from time index k to the predefined horizon
β	:	Data association vector
$X_i^{\beta_i(j)}$	:	Involved state in the jth measurement, at the ith time step
X^{inv}	:	Set of all involved state variables for a given time step

$\mathcal{N}(\cdot,\cdot)$	:	Gaussian distribution
Σ	:	Covariance matrix
Λ	:	Information matrix
W_{k+1}	:	Noise covariance matrix of the motion model
V_{k+1}	:	Noise covariance matrix of the observation model
F	:	Jacobian of the motion function
H	:	Jacobian of the observation function
$\tilde{A}_k$	:	Collective jacobian for time index k
A_{k+i}	:	Collective jacobian for time indices $k : k+i$
Z	:	Multivariate random observations vector
Z^i	:	Component of Z, defined by observation model
Z^s	:	Partition of Z
$Z^{\bar{s}}$	:	Partition of Z, complementary to Z^s
$Z^{n_i\|m_j}$	:	nth node at ith partitioning level, descendant of mth node at jth partitioning level
$J(b_k, \pi_{k+})$	:	Objective (Value) function over the belief at time k given a policy
$J^{\star}(b_k)$	:	Optimal objective function
$\pi^{\star}$	:	Optimal Policy
$V^{\pi}(b_k)$	:	Value (objective) function over the belief at time k given a policy
$Q^{\pi}(b_k, a_k)$	:	Belief-Action value function given a policy
γ	:	Discount factor
$\mathcal{H}(\cdot)$	:	Differential Entropy
$\mathcal{LB}, \mathcal{UB}$	:	Lower and upper bounds over the reward functions

Chapter 1

Introduction

1.1 Planning under uncertainty

Autonomous agents must operate with imperfect information about their environments, dynamics, and measurements. *Belief Space Planning* (BSP) is one of the fundamental problems one must solve for these autonomous agents to interact with the environment successfully. The modeling of the problem is such that we maintain a probability density function over the true state of the agent, which is unknown. We then reason about the evolution of this distribution in the future for different actions and possible observations. BSP uses two main models to evolve the distribution over the state: motion and measurement models. The two models often have very distinctive and different effects. The motion model introduces noise by state uncertainty. The measurement model on the other hand, although noisy by itself, usually provides valuable information about the agent's pose and the map of the environment, which in turn reduces state uncertainty.

One advantage of using the BSP formulation is the capacity to incorporate belief dependent reward functions, particularly, information-theoretic rewards, which is important for tasks such as; search and rescue, informative path planning and active classification. The ability to measure belief uncertainty and reduce it is key in solving such tasks, however, it comes with added computational complexity, especially when the observation space is high-dimensional. In a typical active *Simultaneous Localization And Mapping* (SLAM) setting, future observations may include hundreds or even thousands of landmarks. Moreover, in a visual-based POMDP setting, even a single observation can be high-dimensional when the measurement model uses raw image inputs.

Solving the corresponding POMDP problem involves reasoning about different actions or policies, and for each, account for different possible observations. This leads to an exponential growth of the posterior beliefs, which in turn makes the planning problem NP complete [18].

A possible approach to addressing this issue is to simplify the planning problem. One specific simplification method of interest, is forming analytical bounds on the expected reward and using the bounds to decide on the optimal action. Given that the bounds

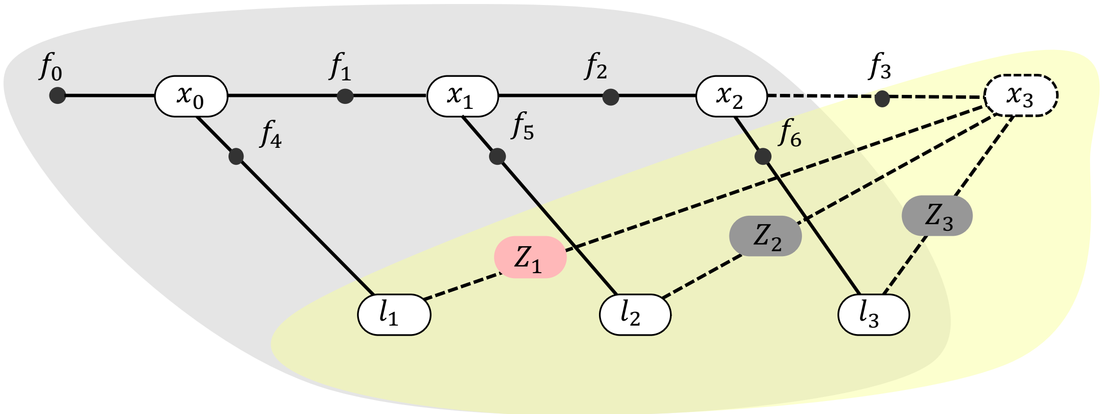

Figure 1.1: A prior factor graph is shown in the gray blob, considering an action that leads to the factor f_3. The posterior graph includes the future measurements Z_1, Z_2, Z_3, the addition to the posterior factor graph is shown in yellow. Different sets of measurements are assigned different colors which represents one possible partitioning, these sets are used to bound the expected Entropy of the entire posterior graph.

are easier to calculate than the expected reward, planning becomes more efficient.

In this work, we present a novel approach of simplification of the POMDP planning problem, specifically to the *multivariate observation space*. When the observation space is high-dimensional, the calculation of information-theoretic rewards becomes expensive. We show that by partitioning the random variables that model future possible measurements into sets, this calculation becomes more efficient. To illustrate, Consider a factor graph representation for a given belief at planning time $t = 2$, as in Fig 1.1. In this toy example, we sketch out how in planning, a specific action leads to 3 new random observations involving different landmarks. One can think of a simplified setting, that takes into account only a subset of these random observations. This subset of observations is a partition of the multivariate random variable representing the 3 original observations. This multivariate random variable defines in this case the observation space for this specific time step, such that, its partitioning is a partition of the observation space. Similarly, in visual based POMDP setting, a partition of the observation space may correspond to a partition of the random variables representing future image pixels into subsets.

Using the partitioned observations, we show the relation to the original problem by deriving analytical bounds on the expected entropy that hold for all families of belief distributions. We present a partition tree that allows greater efficiency as we go down its hierarchy. We show that these bounds are adaptive, computationally cheaper, and that they converge to the original solution. Moreover, we show one possible realization of this general framework for an active SLAM scenario involving multivariate Gaussian distributions and present a hierarchy of efficient implementations. The speed up ranges from a factor of 4 for the least efficient one, and a speed up from a cube to linear time for the most efficient one.

1.2 Related Work

1.2.1 POMDP

POMDP has been widely used as a model for decision-making under uncertainty, despite the fact that obtaining the optimal solution for the planning problem is known to be intractable [18] [8]. While the standard POMDP is formulated with state dependent reward functions, it is possible to extend this framework to include belief dependent reward as well; examples for such frameworks are ρ-POMDP proposed by [1] and BSP by [19] and [27]. This extension is essential for tasks such as terrain monitoring [20], information gathering [25], and active SLAM [13]. Various approximation methods to solving POMDPs were proposed, however even approximating a solution is challenging since most real-world problems incorporate continuous spaces. [24] combined a sparse tree representation and Monte Carlo Tree Search to approximate near-optimal policies, however, it is not suitable for information-theoretic rewards. A particle-based approach to represent the belief was taken by [26], proposing two different algorithms, one of them catering specifically to belief-dependent rewards. An abstraction of the observation model was studied in [2], allowing a speed-up compared with other approximation methods.

Early works have identified the importance of quantifying the information contained in observations. The quantitative value of possible measurements is presented in [4], in an effort to incorporate this value measure in inference. While it did allow for better selection of measurements using heuristics, it did not provide any optimality guarantees. Using the sub-modularity of mutual information, [16] formulated a near-optimal algorithm for sensor placement, which can alternatively cast as a myopic planning algorithm, however it was limited to Gaussian processes.

1.2.2 Inference

Probabilistic graphical models have gained substantial traction in the world of inference, one of the most popular one was introduced by [10] and was later extended at [9]. The latter utilized the structure of SLAM problems to be encoded in a Bayes-tree, which allowed for an incremental update of the posterior belief with incoming information. The computational complexity for big SLAM problems is still very high, and there have been many works that have tried to reduce the computational complexity of inference, considering probabilistic graphical models. Some of the more popular SLAM methods using graphical models for were introduced by [12] presented a graph-theoretic approach to the problem of designing sparse reliable pose-graph SLAM in the context of measurement selection, both [3] and [17] showed methods for compressing a factor graph, and [28] reviewed how feature selection based on some defined scores can improve localization and data association, proposing a greedy algorithm that relies sub-modularity as well.

1.2.3 Simplification

Other works have studied simplification method for planning problems; [23] presented a heuristic-based bounds on the value function, to guide local updates; a belief compression method was proposed in [21], but it lacked guarantees on planning performance. Several works have put forward simplified methods while providing guarantees; For a Gaussian high-dimensional state, [6] proposed a transformation of the original information space to a conservative one, by decoupling all state variables. More general approaches were studied in [5] and [29], the former outlined a theoretical framework for simplification in general while demonstrating said framework for a sparse approximation of the initial belief, while the latter studied a simplification in risk averse planning, while considering a distributional perspective. However, none of these works considered a simplification to the observation space itself.

1.3 Contributions

We put forth a novel concept of *observation space partitioning* that is used to speed up POMDP planning with continuous or discrete spaces. We lay the theoretical foundations for this concept by outlining the partitioning in the common case where observations are modeled as a multivariate random variable. For general belief distributions, we show bounds on the expected reward which is differential entropy over the state. We show that these bounds; are computationally more efficient, can be adaptively changed, and converge to the actual reward. In addition, we show a specific form for these bounds when the belief is normally distributed.

Further more, we double down on the concept of observation space partitioning by introducing a *partition tree*, which encodes a hierarchy of partitions. We show that this hierarchy too can be used to form bounds on the expected reward and we present a hierarchy of efficient implementations for both general and Gaussians beliefs.

Finally, we demonstrate the speed up that can be obtained by this method. We show a significant performance gains for a simulated active SLAM scenario as well as on an actual robot using visual odometery for planning.

Chapter 2

Background

2.1 ρ-POMDP

A discrete-time POMDP models an agent decision process by outlining the dynamics of the interaction between the agent and its environment. It is defined as the tuple $(\mathcal{X}, \mathcal{A}, \mathcal{Z}, T, O, R)$, consisting of a state, action and observation spaces, a transition and observation models, and a reward function. We assume a Markovian transition model, i.e. $T(X, a, X') = \mathbb{P}(X'|X, a)$, and that each measurement is conditionally independent given the state, i.e. $O(X, z) = \mathbb{P}(z|X)$.

Since the agent only observes the environment through noisy measurements, it must maintain a probability distribution over the true state; we denote this distribution as a *belief*. The belief and the propagated belief, are defined as, respectively:

$$b_k \triangleq b[X_k] = \mathbb{P}(X_k|z_{0:k}, a_{0:k-1}) \triangleq \mathbb{P}(X_k|h_k) \tag{2.1}$$

$$b_k^- \triangleq b[X_k^-] = \mathbb{P}(X_k|z_{0:k-1}, a_{0:k-1}) \triangleq \mathbb{P}(X_k|h_k^-). \tag{2.2}$$

At each discrete time step this belief is updated with new motion and observation information according to Bayes' rule. Given an action a_k and observation z_{k+1} the belief is updated according to:

$$b_{k+1} = \eta \int_{X_k} \mathbb{P}(X_{k+1}|X_k, a_k)\mathbb{P}(z_{k+1}|X_{k+1}) b_k dX_k,$$

where η is some normalization factor. A policy $\pi : \mathcal{B} \mapsto A$ maps belief states to actions. Usually, only state dependent reward are considered in the POMDP setting. BSP and later on ρ-POMDP, extend the POMDP model to include belief dependent rewards. For some finite planning horizon $\ell \in [1, L]$, the *value* of a policy π, is defined as the expected cumulative reward received by following π with initial belief b_k:

$$V^\pi(b_k) = R(b_k, \pi_k(b_k)) + \mathop{\mathbb{E}}_{z_{k+1:k+\ell}} \left[\sum_{i=k+1}^{k+\ell} R(b_i, \pi_i(b_i)) \right]. \tag{2.3}$$

Solving a POMDP is equivalent to finding the optimal policy π^* such that the value function is maximized.

In this work, we consider information-theoretic rewards, specifically, differential Entropy:

$$R(b, \pi(b)) \triangleq -\mathcal{H}(X) \equiv \underset{X \sim b}{\mathbb{E}} (\log b[X]) \,, \tag{2.4}$$

where X is a random variable distributed according to $b[X]$.

If both X, Z, are treated as random variables, the expected reward becomes the conditional entropy of these random variables, i.e.

$$\underset{Z}{\mathbb{E}}[R(b)] = -\mathcal{H}(X \mid Z) = -\underset{Z}{\mathbb{E}}[\mathcal{H}(X \mid Z = z)]. \tag{2.5}$$

Thus, the expected reward at each ith look ahead step, can be equivalently written as:

$$\underset{Z_{k+1:i}}{\mathbb{E}}[R(b_i, a_{i-1})] = -\mathcal{H}(X_i | Z_{k+1:i}), \tag{2.6}$$

where the future observations are drawn from the distribution $\mathbb{P}(Z_{k+1:i} \mid b_k, \pi)$ and $i \in [k+1, k+\ell]$.

2.2 Active SLAM

Let x_k be the state of the agent at time k, and X_k be the joint state of the agent's trajectory and environment, e.g. landmarks, up to, and including time k. We define $z_k \triangleq \{z_k^0, ..., z_k^m\}$ as the set of all measurements observed at time k, and $z_{0:k} \triangleq \{z_0, ..., z_k\}$ as the set of all measurements until time k. Similarly we define $a_{0:k} \triangleq \{a_0, ..., a_k\}$ as the set of all actions until time k. Assuming static landmarks, the motion of the agent, and the observations it receives are modeled as:

$$x_{k+1} = f(x_k, a_k) + w_k \quad , \quad z_k = h(X_k) + v_k, \tag{2.7}$$

where f and h are some deterministic functions, and w_k and v_k are their process noise, respectively.

We denote the Data Association vector at time k as β_k. The dimensionality of β_k, is equal to X_k excluding x_k, and is composed of binary entries, where each entry indicates whether the corresponding state was involved in a measurement at that given time step. We assume that each measurement involves the current state x_i, such that β does not account for it. For example, at time step $k = 3$, for a prior state vector of dimensionality 5 and a measurement involving the third and fifth components of the state (e.g. observation of two landmarks): $\beta_3 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \end{pmatrix}^T$.

Alternatively, we can define the objective function using the data association vector:

$$J(b_k, \pi_k(b_k)) \triangleq \mathop{\mathbb{E}}_{\tilde{\beta}}\left[\mathop{\mathbb{E}}_{\tilde{Z}|\tilde{\beta}}\left[\sum_{i=k+1}^{k+\ell} R(b_i, \pi_i(b_i))\right]\right], \tag{2.8}$$

where $\tilde{\beta} \triangleq \beta_{k+1:k+\ell}$ and $\tilde{Z} \triangleq Z_{k+1:k+\ell}$. In this scenario, $\tilde{\beta}$ dictates the number of measurements and the states involved, while $\tilde{Z}$ encodes the information about the distribution of those measurements. Combining Bayes rule and the properties of the models, we can factorize a posterior belief $b[X_{k+\ell}]$ given $\tilde{\beta}$, into prior belief, motion and measurement factors:

$$b[X_{k+\ell}] \propto b[X_k] \prod_{i=k+1}^{k+\ell} \mathbb{P}(x_i|x_{i-1}, a_{i-1})\mathbb{P}(z_i \mid X_i, \beta_i), \tag{2.9}$$

where

$$\mathbb{P}(z_i \mid X_i, \beta_i) = \prod_{j=1}^{m_i(\beta_i)} \mathbb{P}(z_{i,j}|x_i, X_i^{\beta_i(j)}), \tag{2.10}$$

where m_i is the number of measurements at the ith time step, and $X_i^{\beta_i(j)}$ represents the involved state in the jth measurement, at the ith time step, both as a function of β_i. We denote the set of all involved state variables for a given time step as:

$$X_k^{\text{inv}} = \{X_k^{\beta_k(j)} | j \in \mathcal{J}\}, \tag{2.11}$$

where $\mathcal{J} = \{1, 2, .., m_k(\beta_k)\}$.

When the models in (2.7) are linear, with zero-mean Gaussian noise, i.e. $w_k \sim \mathcal{N}(0, W_k)$ and $v_k \sim \mathcal{N}(0, V_k)$, and the prior belief is Gaussian, it can be shown that the posterior belief is also Gaussian. In such case, the Entropy of the posterior belief can be expressed as:

$$\mathcal{H}(X) = \frac{1}{2}(\ln|\Sigma| + N\ln(2\pi e)). \tag{2.12}$$

where $b[X] \sim \mathcal{N}(\mu, \Sigma)$, with mean $\mu \in \mathbb{R}^N$ and covariance matrix $\Sigma \in \mathbb{R}^{N\times N}$. The inverse of the covariance is known as the information matrix, such that $\Sigma_k^{-1} = \Lambda_k$. At each given time step, the information matrix of a posterior belief b_{k+1}, can be decomposed into:

$$\Lambda_{k+1} = \Lambda_k^{\text{Aug}} + F^T W_{k+1}^{-1} F + H^T V_{k+1}^{-1} H, \tag{2.13}$$

where Λ_k^{Aug} is the prior information matrix of b_k, augmented with zeros to accommodate new states, W_{k+1} and V_{k+1} are the noise covariance matrices of the motion and measurement models respectively, and $F \triangleq \nabla f$ and $H \triangleq \nabla h$ are the Jacobians of the motion and measurement functions respectively. The stacked matrices of $F^T W_{k+1}^{-1} F$ and $H^T V_{k+1}^{-1} H$ are denoted as the collective Jacobian $\tilde{A}_k$, see [7] for details.

If we combine the collective Jacobians of consecutive time steps, i.e. $\tilde{A}_{k+1}, \tilde{A}_{k+2}, \ldots, \tilde{A}_{k+i}$ we get the following update rule:

$$\Lambda_{k+i} = \Lambda_k^{\text{Aug}} + \tilde{A}_{k+1:k+i}^T \cdot \tilde{A}_{k+1:k+i}, \tag{2.14}$$

where $\tilde{A}_{k+1:k+i}$ is the collective Jacobian of the motion and measurement factors of (2.9), from the time step $k+1$ until $k+i$.

For the sake of readability, we drop the notation of the history h_i from now on, but assume all distributions of a given time step are conditioned on the history available at the beginning of the planning session. We denote the collective Jacobian of a given horizon $\tilde{A}_{k+1:k+i}$ simply as A_{k+i}.

2.3 Planning using reward bounds

In this section we show how to use measurement selection to simplify the BSP problem. In order to choose the optimal action from a pool of candidate actions, we need to evaluate the reward function for each action. Instead, one can evaluate bounds on the expected reward function as a proxy,

$$\mathcal{LB}_i \leq \mathop{\mathbb{E}}_{Z_{k:i}} (R(b_i)) \leq \mathcal{UB}_i. \tag{2.15}$$

In the same manner we can bound the objective function by summing up the bounds over the reward function for each of the time steps,

$$\sum_{i=k+1}^{k+\ell} \mathcal{LB}_i \leq J\left(b_k, a_{k:k+\ell-1}\right) \leq \sum_{i=k+1}^{k+\ell} \mathcal{UB}_i. \tag{2.16}$$

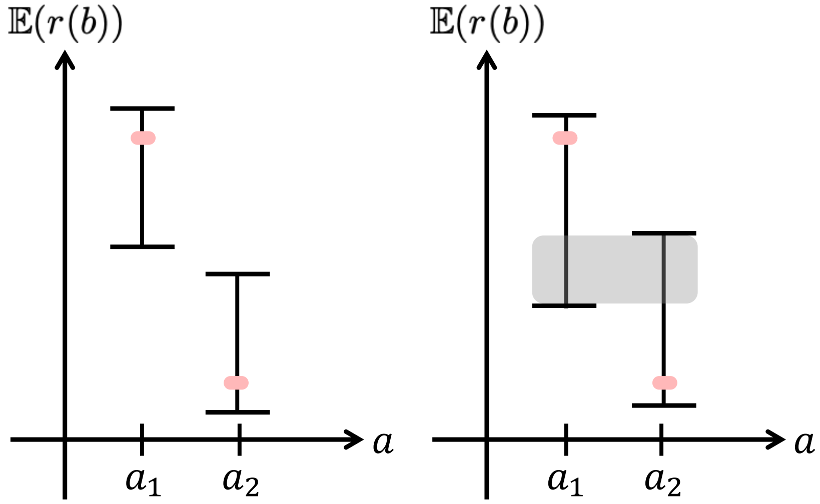

Figure 2.1: The expected reward is highlighted with the bounds. On the left we can select the optimal action based on the bounds alone, on the right the worst-case loss is shaded in gray.

Under the assumption that these bounds can be efficiently calculated, it is easier to select actions based on the reward bounds; see illustration in Fig. 2.1. We can think about two distinct actions, a_1 and a_2 . For each action we calculate the expected reward bounds and face two cases: in the first, the bounds do no overlap and we can select the optimal action, in the second, the bounds do overlap and we choose between tightening the bounds such that they do not overlap, or selecting an action while bounding the loss. Previous works have developed such bounds considering various simplification methods, as discussed in Section 1.2.

In this work, we put forward a fundamental simplification that applies to the observation space itself. Specifically, we propose a partitioning of the observation space and develop expected reward bounds that are a function of this partitioning.

Chapter 3

Approach

3.1 Partitioning of Multivariate Observation Space

Consider a multivariate random variable Z that represents future observations and the corresponding observation space $\mathcal{Z}$. In general, Z can be represented by

$$Z = (Z^1, Z^2, \ldots, Z^m), \tag{3.1}$$

where Z^i is a random variable defined by a given measurement model, and m is the number of such random variables. We can now partition Z into different subset of components, for example, consider the partitioning $Z^s \in \mathcal{Z}^s$ and $Z^{\bar{s}} \in \mathcal{Z}^{\bar{s}}$, such that:

$$\begin{aligned} Z^s &= \{Z^1, Z^2, \ldots, Z^n\}, \\ Z^{\bar{s}} &= \{Z^{n+1}, Z^{n+2}, \ldots, Z^m\}, \end{aligned} \tag{3.2}$$

where $Z = Z^s \cup Z^{\bar{s}}$, and their corresponding subspaces $\mathcal{Z} = \mathcal{Z}^s + \mathcal{Z}^{\bar{s}}$.
In the subsequent section, we derive bounds on the expected reward that are a function of this partitioning, such that $\mathcal{LB}$ and $\mathcal{UB}$ from (2.15) become:

$$\mathcal{LB}_i(b_i, Z^s_{k:i}, Z^{\bar{s}}_{k:i}) \tag{3.3}$$

$$\mathcal{UB}_i(b_i, Z^s_{k:i}). \tag{3.4}$$

The partitioning can be applied (but not limited) to two different observation spaces —a raw measurement such as pixels in an image, or to a SLAM scenario where the measurement model is defined by (2.7), and β dictates the dimension of the measurements, e.g. number of observed landmarks considering a future camera pose.

Taking the former as a toy example, we apply partitioning to a raw image measurement of size 20×20 binary pixels. Each pixel is represented by a random variable $Z^{x,y} \in \{0, 1\}$, where x, y denote the pixel location on the sensor, and $Z \in \mathcal{Z} \subseteq (\mathbb{F}_2)^{400}$. We must consider all of the different permutations for each of those pixels, 2^{400} in total, which defines $|\mathcal{Z}|$ in this case. For example, if we partition Z^s to represent the left half of the image, and $Z^{\bar{s}}$ to represent the right half, we need only to consider 2^{200} permutations for Z^s, and another 2^{200} for $Z^{\bar{s}}$, 2^{201} in total.

Hierarchical Partitioning there can be higher levels of partitioning, breaking down further a given measurement set into two sets. To encode this partitioning scheme we index the partitioning depth, and the number of nodes at a given depth. Each partitioned set is given a unique encoding denoted as $Z^{n_i|m_j}$, where n is the node number at the ith partitioning level, and m is the node number at the parent partitioning level j. We note this slight abuse of notation in regards to (3.2), but it should be clear from the context, which notation is used. If two sets share a parent set, we consider them a base subset and its compliment. For example, $Z^{4_3|2_1}$ represents the 4th node of level 3, where the parent set is the 2nd node of level 1. In this new notation, $Z^s, Z^{\bar{s}}$ becomes $Z^{1_1|1_0}, Z^{2_1|1_0}$ or equivalently $Z^{1_1|1_0}, Z^{\bar{1}_1|1_0}$. Overall, for $Z \in \mathbb{R}^m$, it is possible to create a partition hierarchy of depth $\log_2 m$, as illustrated in Fig. 3.1.

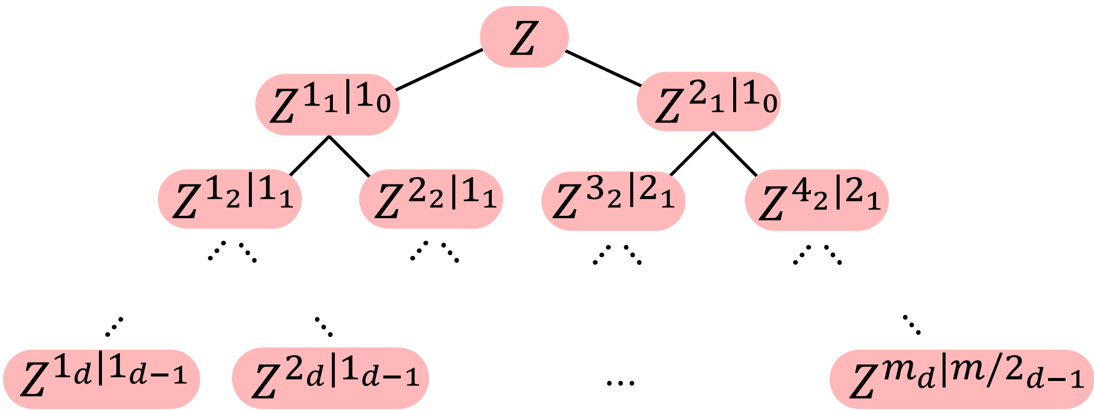

Figure 3.1: An Illustration of a possible partition tree. At each level of partitioning, we split a measurement set into two. For $Z \in \mathbb{R}^m$, the depth of the tree is $d = \log_2 m$.

3.2 Bounds on Expected Entropy

In this section we use measurement partitioning to derive information theoretic reward bounds considering arbitrary distributions.

Lemma 3.2.1. *The conditional Entropy can be factorized as:*

$$\mathcal{H}(X|Z) = \mathcal{H}(Z|X) + \mathcal{H}(X) - \mathcal{H}(Z). \tag{3.5}$$

Proof. For two random variables that have a joint Entropy $\mathcal{H}(X, Z)$, we know that conditioning on Z yields $\mathcal{H}(X, Z) = \mathcal{H}(X|Z) + \mathcal{H}(Z)$. Similarly, conditioning on X yields $\mathcal{H}(X, Z) = \mathcal{H}(Z|X) + \mathcal{H}(X)$. Combining both equations and rearranging terms we obtain the desired equality. □

There are three quantities we need to evaluate in (3.5); the Entropy of the likelihood of a given measurement which is done via the measurement model, the prior Entropy and finally, the Entropy of the measurement given the history. The prior Entropy is common to all actions and can be calculate once, the rest of the terms involve future measurement and we shall apply selection to these terms.

Lemma 3.2.2. *Given two sets of expected measurements, the conditional Entropy can be factorized as:*

$$\mathcal{H}(X|Z) = \mathcal{H}(Z^s|X) + \mathcal{H}(Z^{\bar{s}}|X) - \mathcal{H}(Z^s, Z^{\bar{s}}) + \mathcal{H}(X). \tag{3.6}$$

Proof. Using (3.5), we can rewrite the conditional Entropy as $\mathcal{H}\left(X|Z\right) = \mathcal{H}(Z^s, Z^{\bar{s}}|X) - \mathcal{H}(Z^s, Z^{\bar{s}}) + \mathcal{H}(X)$. The measurements are independent given the state, such that $\mathbb{P}\left(Z^s, Z^{\bar{s}}|X\right) = \mathbb{P}\left(Z^s|X\right)\mathbb{P}\left(Z^{\bar{s}}|X\right)$. The Entropy of two independent random variables is just the sum of individual Entropies such that $\mathcal{H}\left(Z^s, Z^{\bar{s}}|X\right) = \mathcal{H}\left(Z^s|X\right) + \mathcal{H}\left(Z^{\bar{s}}|X\right)$.□

Having established what measurement partitioning looks like in (3.2), we can use it to bound the expected reward. The following two Theorems present our main result.

Theorem 3.1. *The conditional Entropy can be bounded from above by:*

$$\mathcal{H}(X|Z) \leq \mathcal{UB} \triangleq \mathcal{H}\left(Z^s|X\right) + \mathcal{H}\left(X\right) - \mathcal{H}\left(Z^s\right). \tag{3.7}$$

Proof. It is not difficult to show that $\mathcal{H}\left(X|Z^s\right) - \mathcal{H}\left(X|Z\right) = \mathcal{I}(X|Z^s; Z \setminus Z^s)$. Recalling that the mutual information between two random variables is always non-negative, we get:

$$\mathcal{H}\left(X|Z\right) \leq \mathcal{H}\left(X|Z^s\right). \tag{3.8}$$

We denote this the conditioning argument, i.e. conditioning on a random variable always reduces Entropy, and refer to it later. Using lemma 3.2.1 we get $\mathcal{H}\left(X|Z^s\right) = \mathcal{H}\left(Z^s|X\right) + \mathcal{H}\left(X\right) - \mathcal{H}\left(Z^s\right)$. □

Theorem 3.2. *The conditional Entropy can be bounded from bellow by:*

$$\mathcal{H}(X|Z) \geq \mathcal{LB} \triangleq \mathcal{H}(Z^s \mid X) + \mathcal{H}(Z^{\bar{s}}|X) - \mathcal{H}(Z^s) - \mathcal{H}(Z^{\bar{s}}) + \mathcal{H}(X). \tag{3.9}$$

Proof. The joint Entropy of the measurements can be written as $\mathcal{H}\left(Z^s, Z^{\bar{s}}\right) = \mathcal{H}\left(Z^s|Z^{\bar{s}}\right) + \mathcal{H}\left(Z^{\bar{s}}\right)$. From the conditioning argument in (3.8), we get $\mathcal{H}\left(Z^s|Z^{\bar{s}}\right) \leq \mathcal{H}\left(Z^s\right)$, such that $\mathcal{H}\left(Z^s, Z^{\bar{s}}\right) \leq \mathcal{H}\left(Z^{\bar{s}}\right) + \mathcal{H}\left(Z^s\right)$. Rearranging the last inequality concludes the proof. □

We can think of this lower bound from the perspective of mutual information: the difference between the original quantity $\mathcal{H}\left(Z^s, Z^{\bar{s}}\right)$ and the quantities $\mathcal{H}\left(Z^s\right)$ and $\mathcal{H}\left(Z^{\bar{s}}\right)$ is exactly $\mathcal{I}\left(Z^s; Z^{\bar{s}}\right)$, such that the lower bound double counts the mutual information between the measurement sets, see Fig. 3.2.

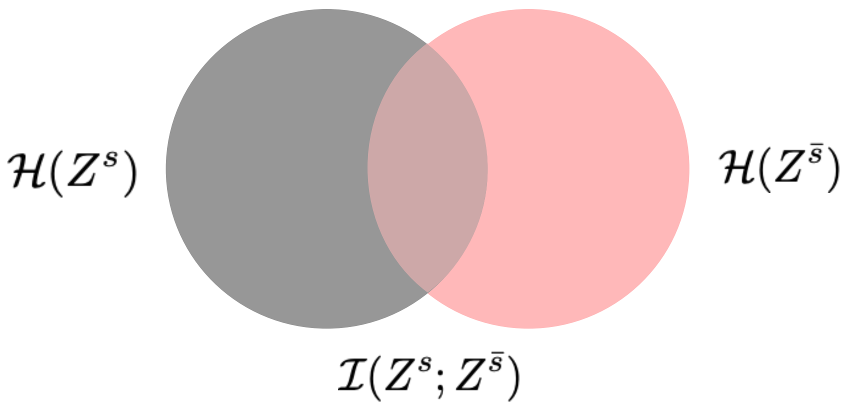

Figure 3.2: Visualizing the marginal Entropies of two measurement variables, the lower bound double counts the overlapping region which is the mutual information between those variables.

The bounds in (3.9) and (3.7) have a beneficial form when the belief is non-parametric. In such cases we do not have access to the underlying belief distribution and we resort to approximation, usually by sampling. In such cases the only closed-form expressions we have are for the motion and measurement models, so we use them for sampling.

On the other hand, for parametric beliefs, we have a closed form expression for $\mathbb{P}(X|Z)$. In such cases it makes sense to rearrange the bounds as follows:

Corollary 3.3. *The conditional Entropy can be bounded by:*

$$\mathcal{LB} = \mathcal{H}(X|Z^s) + \mathcal{H}(X|Z^{\bar{s}}) - \mathcal{H}(X), \tag{3.10}$$

$$\mathcal{UB} = \mathcal{H}(X|Z^s). \tag{3.11}$$

This can be obtained directly from (3.7) and (3.9) using Lemma 3.2.1.

3.3 Bounds with Hierarchical Partitioning

In the previous section, we formed bounds based on partitioning of measurement sets, by double counting the mutual information between said sets. This intuition applies to the hierarchical partitioning from Section 3.1 as well, allowing us to create a hierarchical notion of the bounds, starting with the lower bound. To formulate this idea we define a new operator:

$$g(Z^s, Z^{\bar{s}}) = \mathcal{H}(X|Z^s) + \mathcal{H}(X|Z^{\bar{s}}) - \mathcal{H}(X), \tag{3.12}$$

where $g(Z, \emptyset) = g(\emptyset, Z) \triangleq \mathcal{H}(X|Z) - \mathcal{H}(X)$, and $g(\emptyset, \emptyset) \triangleq -\mathcal{H}(X)$.
Using this operator, (3.10) can be expressed as $\mathcal{LB} = g(Z^s, Z^{\bar{s}})$. Given that measurements sets can be hierarchically partitioned further we can formulate their bounds.

Theorem 3.4. *For the sets $Z^s, Z^{\bar{s}}$, and their respective children $Z^{s_1}, Z^{s_2}, Z^{\bar{s}_1}, Z^{\bar{s}_2}$ the following holds:*

$$
\begin{aligned}
&g(Z^s, Z^{\bar{s}}) \geq \\
&g(Z^{s_1}, Z^{s_2}) + g(\emptyset, Z^{\bar{s}}) \geq \\
&g(Z^{s_1}, Z^{s_2}) + g(Z^{\bar{s}_1}, Z^{\bar{s}_2}) + g(\emptyset, \emptyset)
\end{aligned}
$$

Proof. From (3.10) we know that

$$\mathcal{H}(X|Z) \geq \mathcal{H}(X|Z^s) + \mathcal{H}(X|Z^{\bar{s}}) - \mathcal{H}(X),$$

substituting $\mathcal{H}(X|Z^s)$ instead of $\mathcal{H}(X|Z)$ yields

$$\mathcal{H}(X|Z^s) \geq \mathcal{H}(X|Z^{s_1}) + \mathcal{H}(X|Z^{s_2}) - \mathcal{H}(X),$$

which proves the first inequality. Doing the same for $\mathcal{H}(X|Z^{\bar{s}})$ proves the second inequality. □

Each of the quantities in Theorem 3.4 is a lower bound on the expected reward by itself. As for the upper bound, the Entropy of each of the child sets of a given set, is an upper bound on the Entropy of that given parent set.

Theorem 3.5. *For the set Z^s, and its children Z^{s_1}, Z^{s_2} the following holds:*

$$\mathcal{H}(X|Z^s) \leq \mathcal{H}(X|Z^{s_1}) \wedge \mathcal{H}(X|Z^s) \leq \mathcal{H}(X|Z^{s_2}) \tag{3.13}$$

Proof. This follows directly from the conditioning argument. □

We can use the above theorems to perform further partitioning of the measurement sets. Theorem 3.4 shows that we can mix different partition depths for the lower bound, while Theorem 3.5 shows that any node in the partition tree is an upper bound, see partition tree Fig. 3.3 for example.

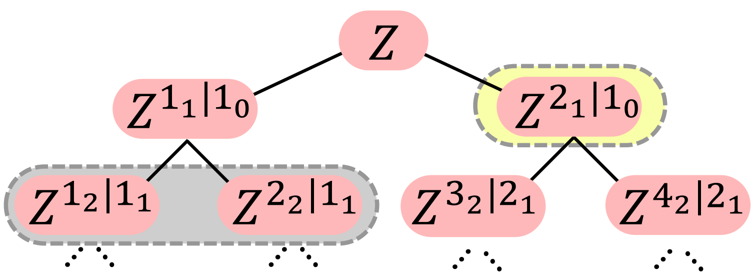

Figure 3.3: Any combination of children nodes, that their union equals to the parent node, can make up a lower bound on that parent node. Any child node at any depth can make up an upper bound on a parent node. For instance,$g(Z^{\bar{s}_1}, \emptyset)$ and $g(Z^{s_1}, Z^{s_2}) + g(\emptyset, Z^{\bar{s}})$ are upper and lower bound bound on $g(Z, \emptyset)$, respectively.

We also note that we have the ability to adaptively change those bounds, by moving between partitioning levels, as well as by moving measurements from one set to its compliment. In the next section we show the convergence of the bounds, so by adaptively changing the bounds we also control how tight they are.

3.4 Analysis of the Bounds

In this section, we analyze the properties of the derived bounds. We look at the convergence and monotonicity of the bounds, both as a function of the sets, for a given partition depth, and as a function of the partition depth. We then examine the computational complexity of obtaining the bounds.

3.4.1 Convergence

For a given partition depth of $d \in [1, \log_2 m]$, we show that the bounds converge to the bounds of the parent depth, $d-1$. In particular, when the partition depth is 1 the bounds converge to the original expected reward. For $Z^s \cup Z^{\bar{s}} \subseteq Z$, the upper bound converges when we add variables to the set Z^s, while the lower bound converges when we remove variables from the set $Z^{\bar{s}}$ and add them to the set Z^s. We show the proof for the first partition depth, but it is valid for any arbitrary depth.

Theorem 3.6. *If $Z^s \to Z$ and $Z^{\bar{s}} \to \emptyset$, then $g(Z^s, Z^{\bar{s}}) - \mathcal{H}(X) \to \mathcal{H}(X|Z)$ and $g(Z^s) \to \mathcal{H}(X|Z)$*

Proof. WLOG, assuming $Z^s \in \mathbb{R}^n$ and $Z^{\bar{s}} \in \mathbb{R}^{m-n}$ as in (3.2), we start with the upper bound, $\mathcal{UB} = \mathcal{H}(X|Z^s)$. Using the conditioning argument, $\mathcal{H}(X|Z^s \cup \{Z^{n+1}\}) \leq \mathcal{H}(X|Z^s)$ where $Z^{n+1} \subseteq Z^{\bar{s}}$ but $Z^{n+1} \not\subset Z^s$. We continue adding variables to the set Z^s in this way, until $Z^s \cup \{Z^{n+1}\} \cup \ldots \cup \{Z^m\} = Z^s$ and $\mathcal{H}(X|Z^s \cup \{Z^{n+1}\} \cup \ldots \cup \{Z^m\}) = \mathcal{H}(X|Z)$.
As for the lower bound: $\mathcal{LB} = \mathcal{H}(X|Z^s) + \mathcal{H}(X|Z^{\bar{s}}) - \mathcal{H}(X)$. We use the same argument, while working in the opposite direction with $Z^{\bar{s}}$: $\mathcal{H}(X|Z^{\bar{s}}) \leq \mathcal{H}(X|Z^{\bar{s}} \setminus \{Z^{n+1}\})$, where $Z^{n+1} \subseteq Z^{\bar{s}}$. We continue removing variables from the set $Z^{\bar{s}}$ in this way until $Z^{\bar{s}} \setminus \{Z^{n+1}\} \setminus \ldots \setminus \{Z^m\} = \emptyset$ and $\mathcal{H}(X|\emptyset) = \mathcal{H}(X)$. Plugging into the expression for $\mathcal{LB}$ we get $\mathcal{H}(X|Z) + \mathcal{H}(X|\emptyset) - \mathcal{H}(X) = \mathcal{H}(X|Z)$ □

3.4.2 Monotonicity

Note that from the conditioning argument, we can see that $\mathcal{UB}$ monotonically converges to the expected reward, or in different words, the Entropy is monotone in the size of measurement set. This applies for both between different partitioning depth, and for a given one. However, we cannot say the same about $\mathcal{LB}$ since it depends on the change of both $\mathcal{H}(X|Z^s)$ and $\mathcal{H}(X|Z^{\bar{s}})$. We always move measurements from one set to the other, such that these two quantities change in opposition to one another. We cannot say a-priori what change is greater and thus cannot deduce monotonicity. In fact, it can be proven that this bound is non-monotone using the sub-modularity of the Entropy in the measurements, but it is beyond the scope of this work.

3.4.3 Computational Complexity

It is only rational to use the bounds instead of the full calculation of the expected reward function, if computing the bounds is cheaper computationally. We are now going to compare these two calculations.

For general distributions, we can compare the calculation as a function of the observation space size. The baseline calculation involves evaluating the following quantities (omitting history and actions, see lemma 3.2.2):

$$\mathcal{H}(Z^s|X), \mathcal{H}(Z^{\bar{s}}|X), \mathcal{H}(X), \mathcal{H}(Z^s, Z^{\bar{s}}). \tag{3.14}$$

Using the bounds we need to evaluate (see theorems 3.2 and 3.1):

$$\mathcal{H}(Z^s|X), \mathcal{H}(Z^{\bar{s}}|X), \mathcal{H}(Z^{\bar{s}}), \mathcal{H}(Z^s), \mathcal{H}(X). \tag{3.15}$$

Since that the prior Entropy is not a function of the action or expected measurements, we can calculate it once for each planning session. Overall, the difference between the the expected reward and the bounds boils down to the difference between the joint versus the marginal Entropy of the measurements, i.e. $\mathcal{H}(Z^s, Z^{\bar{s}})$ versus $\mathcal{H}(Z^{\bar{s}})$ and $\mathcal{H}(Z^s)$. The baseline calculation is:

$$\mathcal{H}(Z) = -\int_{Z^s}\int_{Z^{\bar{s}}} \mathbb{P}(Z^s, Z^{\bar{s}}) \log \mathbb{P}(Z^s, Z^{\bar{s}}) dZ^s dZ^{\bar{s}}. \tag{3.16}$$

Assuming that the observation space is finite and countable, the cost of evaluating the integral terms is a function of the random variables. Evaluating (3.16) is of the order of $O(|\mathcal{Z}^s||\mathcal{Z}^{\bar{s}}|)$, while evaluating the simplified terms $\mathcal{H}(Z^s), \mathcal{H}(Z^{\bar{s}})$, is of the order of $O(|\mathcal{Z}^s| + |\mathcal{Z}^{\bar{s}}|)$. $\mathcal{Z}^s$ and $\mathcal{Z}^{\bar{s}}$ are entirely defined by the measurement model, see example on section 3.1

Note that, in practice we do not have access to the joint or marginal distribution over the measurements, only the measurement model. Marginalization over the state yields:

$$\begin{aligned}
\mathcal{H}(Z) &= -\int_Z\int_X \mathbb{P}(Z, X) \log \int_{X'} \mathbb{P}(Z, X')dX'dZdX && (3.17)\\
&= -\int_Z\int_X \mathbb{P}(Z|X)\mathbb{P}(X) \log \int_{X'} \mathbb{P}(Z|X')\mathbb{P}(X')dX'dZdX && (3.18)
\end{aligned}$$

The simplification is of the same factor, only now it is magnified by the state vector size to the second power; $O(|\mathcal{Z}^s||\mathcal{Z}^{\bar{s}}||\mathcal{X}^2|)$ for the baseline, and $O((|\mathcal{Z}^s| + |\mathcal{Z}^{\bar{s}}|)|\mathcal{X}^2|)$ for the bounds.

3.5 High Dimensional State

In this section we discuss what the measurement simplification looks like when the state space is high dimensional. Specifically, we consider an active SLAM formulation as in Section 2.2. Utilizing data association information from (2.9) and (2.11), we can further simplify the bounds (3.7) and (3.9):

$$\mathcal{LB} \triangleq \mathcal{H}\left(Z^s|X^{\mathrm{inv}_s}\right) + \mathcal{H}\left(Z^{\bar{s}}|X^{\mathrm{inv}_{\bar{s}}}\right) - \mathcal{H}\left(Z^s\right) - \mathcal{H}\left(Z^{\bar{s}}\right) + \mathcal{H}\left(X\right), \tag{3.19}$$

$$\mathcal{UB} \triangleq \mathcal{H}\left(Z^s|X^{\mathrm{inv}_s}\right) + \mathcal{H}\left(X\right) - \mathcal{H}\left(Z^s\right), \tag{3.20}$$

where X^{inv_s} and $X^{\mathrm{inv}_{\bar{s}}}$, are the states involved in the random measurements Z^s and $Z^{\bar{s}}$, respectively, as determined by β.

3.6 Gaussian Belief

The bounds shown in the previous section, hold for general belief distributions. In this section we develop a specific form of the bounds, considering Gaussian distributions and a high-dimensional state in the context of active SLAM.

In the case of Gaussian distributions, we can replace the Entropy term in (2.5) for its closed form expression, combining (2.12) and (2.14). For a lookahead step $i \in [k+1, k+\ell]$ the expected Entropy can be expressed as:

$$\mathop{\mathbb{E}}_{Z_{k+1:i}}[\mathcal{H}(X_i|h_i)] = C - \frac{1}{2}\mathop{\mathbb{E}}_{Z_{k+1:i}}[\ln\left|\Lambda_k^{\mathrm{Aug}} + A_i(z)^T A_i(z)\right|], \tag{3.21}$$

where $A_i(z) = \tilde{A}_i(z_{k+1:i})$, is the collective Jacobian for the entire planning horizon as a function of the measurements, and $C \triangleq N\ln(2\pi e)$.

For convenience we define: $f(\Lambda, A) \triangleq |\Lambda + A^T A|$. Since we are interested in the Jacobian of the measurement but not the motion factors, we choose to present the expected Entropy in the following form:

$$\mathop{\mathbb{E}}_{Z_{k+1:i}}[\mathcal{H}(X_i|h_i)] = C - \frac{1}{2}\mathop{\mathbb{E}}_{Z_{k+1:i}}[\ln f\left(\Lambda_k^{\mathrm{Aug-}}, A_i(z)\right)], \tag{3.22}$$

where $\Lambda_k^{\mathrm{Aug-}}$ is the propagated belief, i.e. without taking into account measurements until time step i, augmented with zeros and the measurements vector defined in (3.1) becomes $Z_{k+1:i}$.

Utilizing (3.10), (3.11) and (3.21), we are ready to present the expected reward bounds for the Gaussian case. The bounds apply in expectation, and are more efficient to calculate than SOTA methods per sample of measurements under the expectation operator.

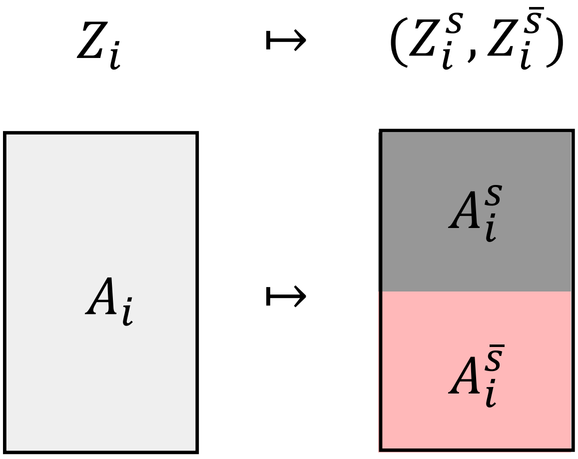

Figure 3.4: Illustration of a possible partitioning of a measurement collective Jacobian, given some partitioning of Z_i.

Theorem 3.7. *For the ith look ahead step, the expected Entropy of a Gaussian belief can be bounded by:*

$$\mathcal{LB} = C - \frac{1}{2} \underset{Z_{k+1:i}}{\mathbb{E}} [\ln \frac{f\left(\Lambda_k^{\text{AUG}-}, A_i^s\right) \cdot f\left(\Lambda_k^{\text{AUG}-}, A_i^{\bar{s}}\right)}{|\Lambda_k^{\text{AUG}-}|}], \tag{3.23}$$

$$\mathcal{UB} = C - \frac{1}{2} \underset{Z_{k+1:i}}{\mathbb{E}} [\ln f\left(\Lambda_k^{\text{AUG}-}, A_i^s\right)], \tag{3.24}$$

where A_i^s and $A_i^{\bar{s}}$ are the Jacobian rows associated with the measurements Z_i^s and $Z_i^{\bar{s}}$, respectively, see Fig. 3.4.

Proof. We have obtained a closed form expression (3.21) for the expected Entropy. If we plug in these expressions into (3.10) we get:

$$\begin{aligned} \mathcal{H}(X_i|Z_i) \geq C &- \frac{1}{2} \underset{Z_{k+1:i}}{\mathbb{E}} [\ln f\left(\Lambda_k^{\text{AUG}-}, A_i^s\right)] \\ &+ C - \frac{1}{2} \underset{Z_{k+1:i}}{\mathbb{E}} [\ln f\left(\Lambda_k^{\text{AUG}-}, A_i^{\bar{s}}\right)] \\ &- C + \frac{1}{2} \underset{Z_{k+1:i}}{\mathbb{E}} [\ln \left|\Lambda_k^{\text{AUG}-}\right|]. \end{aligned}$$

In a similar way, using (3.11) we obtain the upper bound. □

3.6.1 Estimation of Expected Entropy

An empirical expectation is generally calculated by sampling the observation model; for each such sampled observation a posterior distribution of the belief needs to be obtained. The latter usually involves solving a non-linear optimization problem via some iterative linearization method. For each posterior belief, one can calculate the corresponding reward bounds using (3.23) and (3.24), where the expectation operator is approximated by samples. In this case, the performance guarantees are asymptotic, since one assumption that was used to obtain the bounds is only asymptotically valid, namely, the non-negativity of the Mutual Information between measurements. We note that it is possible to formulate non-asymptotic guarantees in this case, but it is outside of the scope of this work and we leave it for future work.

A different approach, would be to make a common assumption for the optimization process within planning, by taking a single iteration step. In the context of planning, this is usually a good approximation for the linearization point when the state prior is informative, and is accurate when the measurement model uses a linear function or for Maximum Likelihood estimation for the measurements [7, 19]. Under said assumption, the Jacobian is only a function of β and not of any particular measurement realization. When the Jacobian is independent of the actual measurement values, we can drop the expectation operator from (3.21) and still represent the expected Entropy:

$$\mathcal{H}\left(X_i | Z_{k+1:i}\right) = C - \frac{1}{2}(\ln \left|\Lambda_k^{\text{Aug}} + (A_i)^T \cdot A_i\right|). \tag{3.25}$$

In this case, the bounds in (3.23) and (3.24), as well as other SOTA calculations should be computed only once per action, i.e., without the expectation operator.

3.6.2 Methods for Determinant Calculation

The computational cost of the bounds in the Gaussian case, is the cost of evaluating the appropriate determinants, $|\Lambda_k^{\text{AUG}-} + (A_i^s)^T A_i^s|$ and $|\Lambda_k^{\text{AUG}-} + (A_i^{\bar{s}})^T A_i^{\bar{s}}|$. At a first glance, it is not clear that evaluating the bounds is indeed more efficient than evaluating the expected reward, we will now show one method which is, in fact, efficient.

There are three main methods to calculating the Entropy of the posterior belief: calculating the determinant of the posterior information matrix (denoted to as baseline), calculating the determinant of the posterior information matrix in its square root form (denoted as R), and using the Augmented Matrix Determinant Lemma (rAMDL) [14,15]. In essence, rAMDL utilizes the well known Matrix Determinant Lemma, combined with some clever calculation re-use, to efficiently evaluate the determinant of the posterior information matrix. It requires a one time calculation of some specific covariance entries, and its general form is as follows:

$$|\Lambda + A^T A| = |\Lambda| \cdot |\Delta| \cdot \left|(A_{\text{new}})^T \cdot \Delta^{-1} \cdot A_{\text{new}}\right|, \tag{3.26}$$

where $\Delta = I_m + A_{\text{old}} \cdot \Sigma \cdot (A_{\text{old}})^T$, Σ is the prior covariance matrix, and A_{old} and A_{new} are the blocks of the Jacobian matrix A, with respect to states at planning time (old) and states added by future actions (new), i.e., $A = [A_{old}, A_{new}]$

Applying measurement partitioning to the baseline method is not efficient: the baseline cost is $O(n^3)$, where n is the dimension of Λ. Since n remains the same given a partition, the overall calculation would be worse than $O(n^3)$.

We choose to apply measurement partitioning to rAMDL and not R for two reasons: Firstly, it has been shown that rAMDL is faster than R within planning [14,15]. Secondly, applying measurement partitioning in R does not make sense - although calculating the determinants in R is $O(n)$, most of the computational burden lies in the update of the posterior R matrix. This update involves a QR decomposition and by partitioning we split it into two separate QR decompositions that on average, are less efficient than the original one. We are then left with the most efficient method, which is rAMDL.

3.6.3 Computational Complexity

One main insight, is that the cost of calculating the determinant using rAMDL (not including the one-time calculation which is common across all actions), is $O(m^3)$, where m is the dimension of A. In a typical high-dimensional scenario, $m \ll n$, and by partitioning the measurements we reduce the dimension of A even further so it becomes even more efficient.

Applying (3.26) to the determinants required for (3.23) and (3.24) results in:

$$|\Lambda_k^{\text{AUG}-} + (A_i^s)^T A_i^s| = |\Lambda_k| \cdot |\Delta_k^s| \cdot \left|(A_{\text{new}}^s)^T (\Delta_k^s)^{-1} A_{\text{new}}^s\right|,$$

where $\Delta_k^s = I_m + A_{\text{old}}^s \cdot \Sigma_k \cdot (A_{\text{old}}^s)^T$, and A_{old}^s and A_{new}^s are the old and new blocks of the matrix A_s, i.e., $A^s = [A_{old}^s, A_{new}^s]$. In a similar way we apply rAMDL to $|\Lambda_k^{\text{AUG}-} + (A_i^{\bar{s}})^T A_i^{\bar{s}}|$.

The most efficient partitioning, for the first depth, is when $s = \bar{s}$, so the resulting matrices are of rank $\frac{m}{2}$. The cost of evaluating each of the determinants for the partitioned Jacobians is $O(\frac{m^3}{8})$. Compared to rAMDL, the cost is reduced from $O(m^3)$ to $O(\frac{m^3}{4})$ by using the bounds.

Chapter 4

RESULTS

In this section, we start off by demonstrating the properties of the bounds as presented in 3.4, for a typical active SLAM scenario. We then show that compared to the other SOTA method our approach is faster while achieving similar reward, both in a simulated scenario and in real-world experiment.

4.1 Setup

The setting considered is a high-dimensional SLAM, where the belief is normally distributed and the reward function is differential Entropy. The state includes past poses, current poses, and landmarks. For simplicity, we generate observations assuming Maximum-Likelihood. All scenarios assume a prior belief containing poses and landmarks is available at the beginning of the planning session, represented as factor graph shown in figure 4.1.

GTSAM 4.1.0 and Python 3.9.7 were used for the simulated scenarios, running on Ubuntu 18.04 and AMD Ryzen 7 3700X 8-Core Processor. For the real-world experiment GTSAM 4.1.0 and Python 3.8 were used, running on Ubuntu 20.04 and Intel(R) Xeon(R) CPU E5-1620.

4.2 Bounds Analysis

4.2.1 Inter Depth Properties

We show the properties of each of the bounds, as individual measurements are moved from on partition to the other, specifically, as $Z^s \rightarrow Z$ and $Z^{\bar{s}} \rightarrow \emptyset$. We can see that both bounds converge to the actual expected Entropy, the upper bounds does so monotonically while the lower bounds does not, as expected. As proposed in Section 3.4, the lower bound does not converge monotonically because it is a function of the mutual information between the measurement partitions. Since the assignment to partitions is random, so is the value of the mutual information between them.

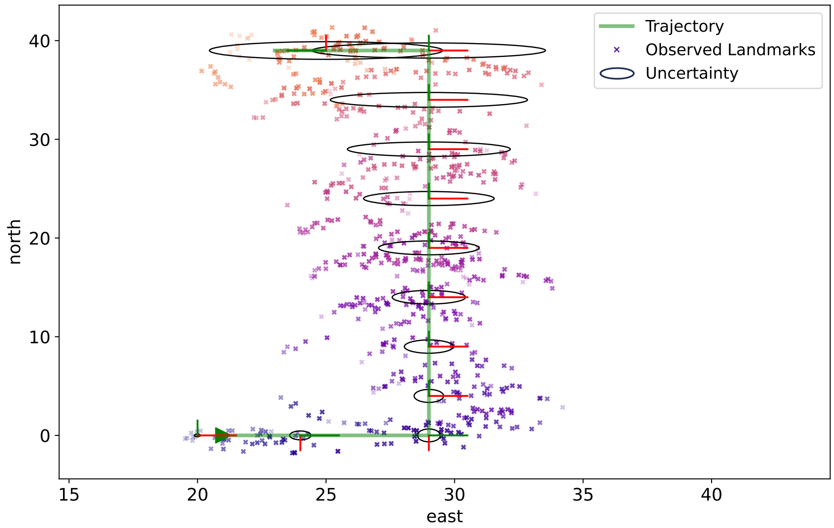

Figure 4.1: Illustration of the prior belief showing a subset of the state landmarks. The uncertainty associated with the joint covariance matrix is shown for every fifth pose.

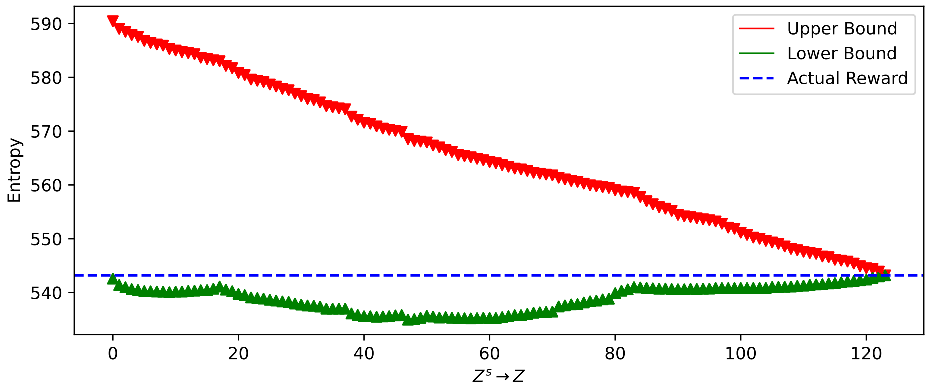

Figure 4.2: Empirical demonstration of the bounds' behavior, for $Z^s \to Z$.

4.2.2 Intra Depth Properties

We show the properties of the bounds, as we go deeper in the partition tree. Figure 4.3 shows, as expected, that the lower we go down the partition tree, the efficiency of the bounds comes at the expanse of their tightness. We can see that both bounds monotonically converge to the real expected reward as we go up the partition tree.

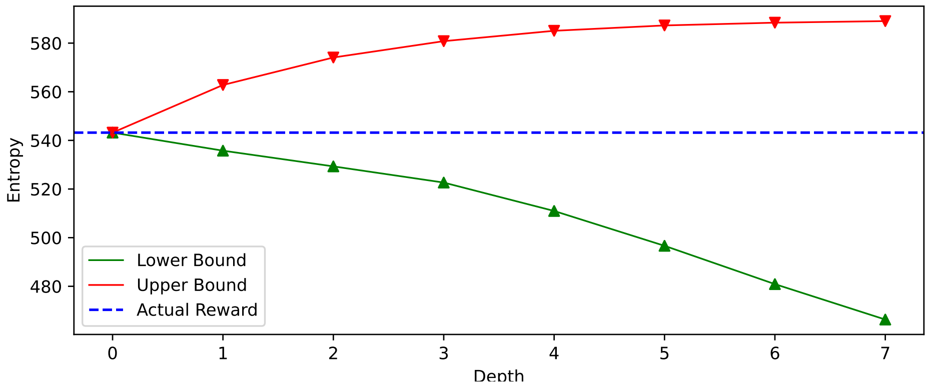

Figure 4.3: Empirical demonstration of the bounds' behavior, as $d \to \log_2 m$.

4.2.3 Correlation and Tightness

Here we demonstrate the usefulness of the bounds in the general case, by presenting the expected reward bounds, for a set of myopic random actions. Looking at figure 4.4, we can see that although some bounds overlap, we can safely prune around 60 percent of the actions which are sub-optimal.

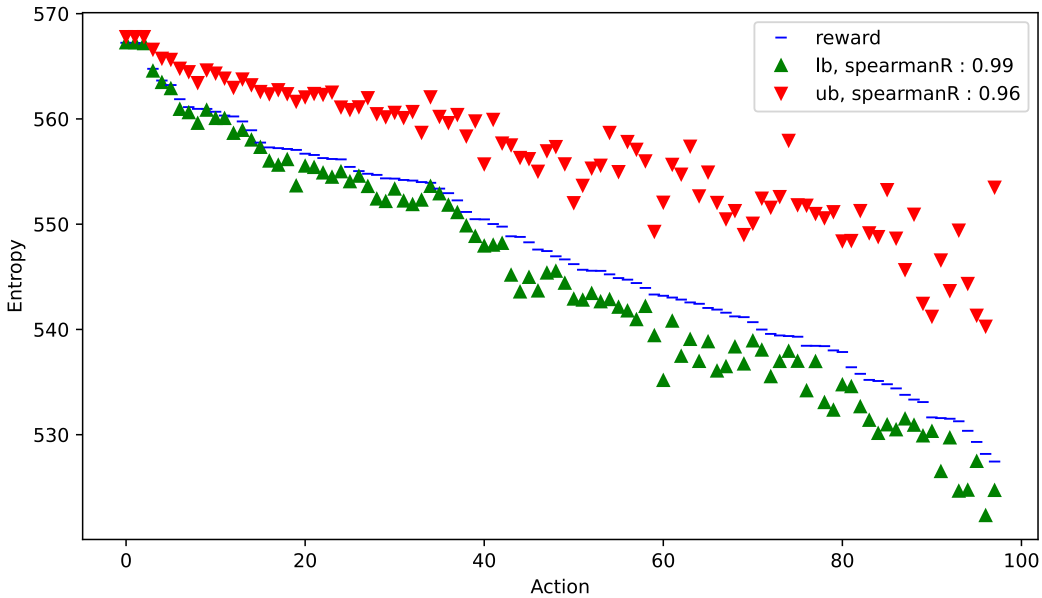

Figure 4.4: Expected Entropy bounds for each sampled action.

We can observe that the lower bound is very strongly correlated to the actual reward while the upper bound shows slightly weaker correlation to the reward. Again, a possible explanation for this behavior is the random assignment of the individual observations to their corresponding set. Since the lower bounds accounts for both sets, it is only a function of the their mutual information, while the lower bound is a function of this random assignment.

Next we compare two specific actions, taken from the previous random action set, by showing their respective hierarchical bounds. Figure 4.5 shows, that for actions that lead to rewards which are distinct, it is possible to select the optimal action based on a very efficient calculation of the bounds. Specifically, we compare an action which explores a new part of the map, to another which re-observes previously seen landmarks.

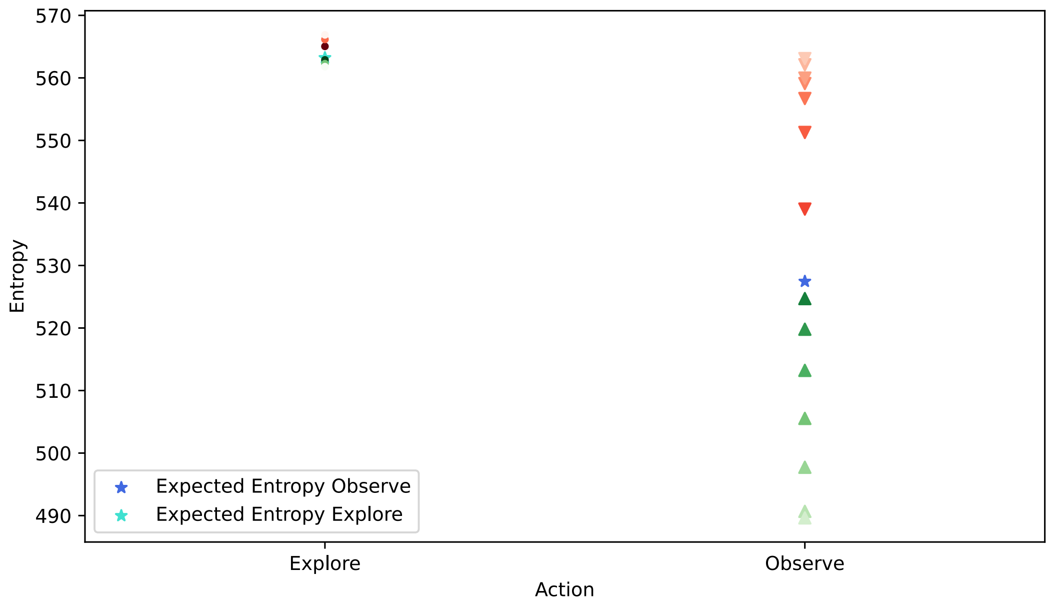

Figure 4.5: Hierarchical bounds for two distinct actions, the lighter the color the more efficient the bounds are.

4.3 Planning Using Bounds

4.3.1 Simulation

We now demonstrate the use of the bounds in a simulated SLAM scenario using the GTSAM library and a *Probabilistic Roadmap* (PRM) generator [11], the goal is to demonstrate a typical use-case in active SLAM setting. We start the planning session with a prior belief as in figure 4.1. We then use the PRM to randomly generate a set of candidate paths, shown in figure 4.6. As seen in figure 4.7, the paths start at the green dot and the goal shown in red, each path consisting of around 12 actions. For each path from a set of the shortest ones, we evaluate the expected cumulative reward and its bounds, and choose the optimal one.

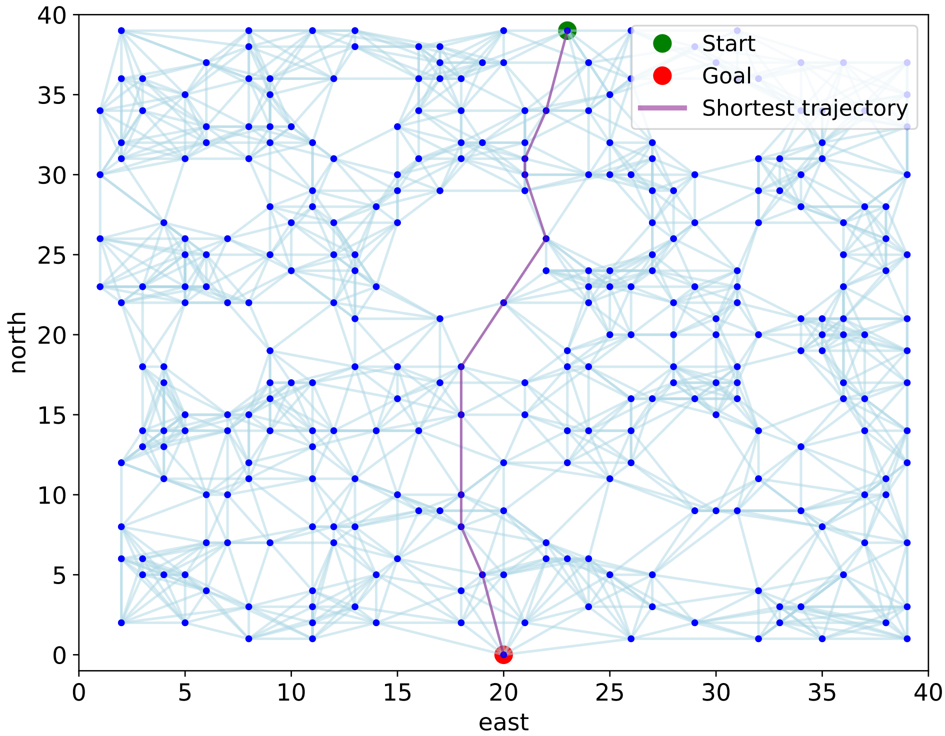

Figure 4.6: PRM generator is used to construct a viable path from the agent's pose at the start of the planning session to the desired goal. We then select a number of the shortest ones.

We start by comparing our proposed method using Measurement Partitioning (MP) to rAMDL. Since our method uses rAMDL to compute the determinants required for the bounds, many of the calculations are common and are not accounted for in the comparison. When compared to iSAM2, these will be accounted for. We compared the performance for three different scenarios, while two of them utilize re-planning. In a re-planning scenario, the agent chooses an optimal path, takes the first action, re-draws paths to the goal from the new pose. The number of re-planning steps is 5, since the

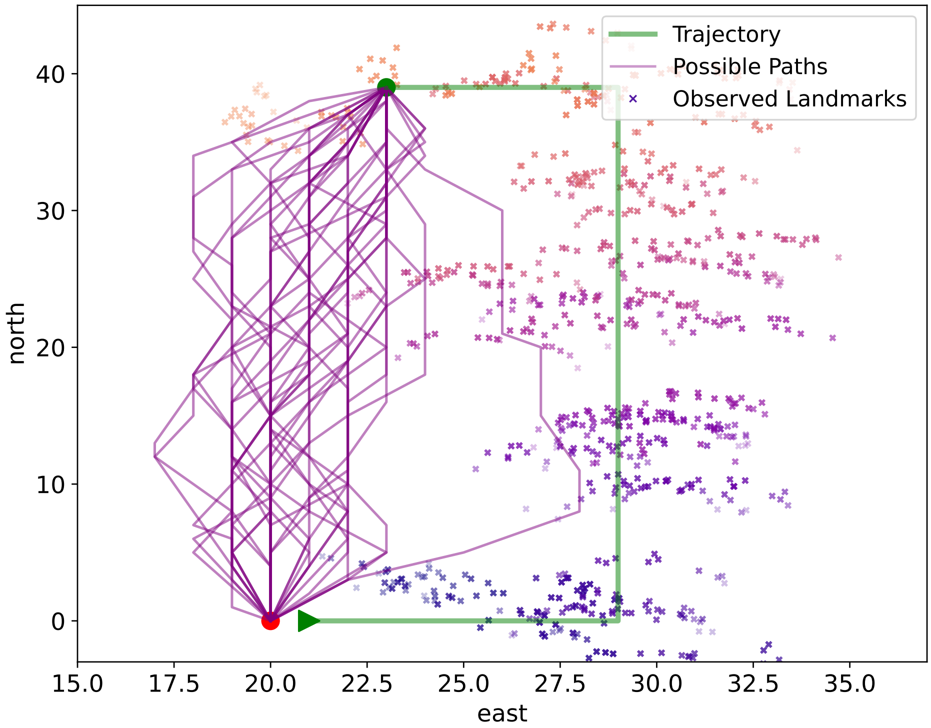

Figure 4.7: Illustration of the simulated scenario, showing a subset of the landmarks and paths.

goal is fixed, each consecutive trajectory is shorter than its predecessor, see Fig 4.8 for details.

From Table [4.1] we can see that the speed-up increases with the number of factors. That is expected since the speed-up is correlated with the number of Jacobian rows. Since most of the prior factors relate to landmarks, a higher number of factors results in a higher number of observation. We can also see that the speed-up is approaching the theoretical value of $O(\frac{m}{4})$, shown in sub-section 3.6.3. Similar to all of the following scenarios, we use the bound to prune sub-optimal paths; when the bounds overlap we choose the path that has the lowest lower bound and bound the possible loss we might incur. For example, for the third scenario in table [4.1], the bound on the expected loss was 109.578 and the optimal cumulative reward was 5215.999. The ratio of loss divided by the optimal cumulative reward was 0.021, such that in the worst case, the trajectory chosen would set us off from the optimal one by two percent. Although unknowable in planning, in this specific case as in many others, the lowest lower bound was indeed associated with the optimal cumulative rewards, and the loss in practice was equal to zero.

Next we compare our method to iSAM2 and rAMDL, including all one-time calculations. We use iSAM2 [9] to incrementally update the square root of the information

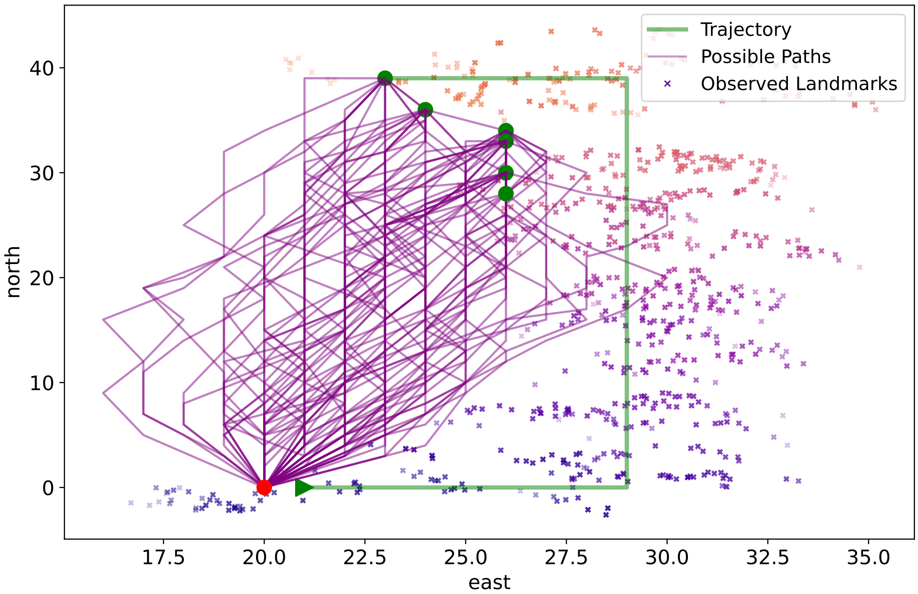

Figure 4.8: Re-planning scenario, each green dot represents the new pose for planning.

# Paths	# Factors	RP	rAMDL	MP (ours)
100	2956	No	11.521 ± 0.537	$\mathbf{6.888 \pm 0.155}$
100	2956	Yes	24.636 ± 1.381	$\mathbf{11.758 \pm 0.372}$
100	5904	Yes	84.376 ± 14.458	$\mathbf{32.069 \pm 4.913}$

Table 4.1: Total planning time in seconds (lower is better), not including all common one-time calculations.

matrix, then use the updated R matrix to calculate the reward as in sub-section 3.6.2. The number of factors in the prior graph was 12185 and number of paths evaluated was 2500. Figure 4.9 shows the bounds for this simulated scenario, that evaluated paths without re-planning, where we can see that about 85 percent of the sub-optimal paths can be safely pruned using the bounds. Table [4.2] shows the total planning time for each method. Table [4.3] shows separately the time it takes to recover the required entries from the prior covariance matrix: *worst-case* is the time it takes to recover the full matrix, while the *actual* is the time it takes to recover only the entries required for the states involved in the measurements for the set of all trajectories. Generally, the earlier in the robot's past trajectory the involved states are, the more of the covariance matrix entries we would need to recover, which is time consuming. However, in a scenario such as that, the time it would take to update the posterior R matrix needed for iSAM2 would be significant as well. Including the covariance recovery, we can observe that our method is about 45 percent faster than rAMDL, and about 65 percent faster than iSAM2.

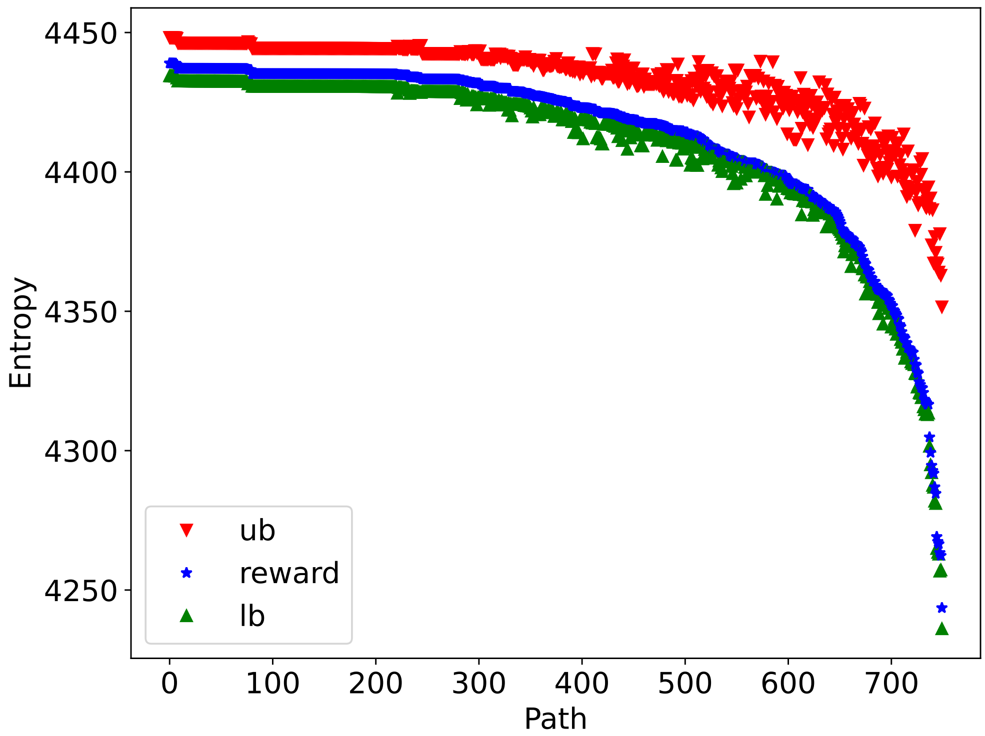

Figure 4.9: Bounds for simulated scenario, showing a subset of the lowest Entropy paths.

Method	time [sec]
MP (ours)	$\mathbf{323.445 \pm 0.175}$
rAMDL	590.584 ± 0.463
iSAM2	728.854 ± 0.348

Table 4.2: Total planning time

worst-case [sec]	actual [sec]
21.05	6.055

Table 4.3: Covariance recovery time

4.3.2 Experiment

Next, we demonstrate the use of the bounds in a real-world experiment using a DJI Robomaster S1, equipped with a stereo ZED camera. GTSAM is used for the backend while the frontend is handled by SuperGlue [22]. The planning session starts after a partial mapping of the room which is stored as a prior factor-graph. From its current pose, the robot is given a goal, it then uses the PRM to generate possible trajectories to its goal. For each method 1000 paths were evaluated, each consists of 20 actions.

Figure 4.10 shows the hardware setup for the experiment, figure 4.11 shows how feature matching is done. Visual odometry is used as motion model factors, and projection factors are used as observation factors.

Table [4.4] summarizes the planning time for each of the methods. We see again that our method allows for a meaningful speed-up. Additionally, we can observe that MP

Figure 4.10: Robomaster S1 robot, equipped with a ZED camera

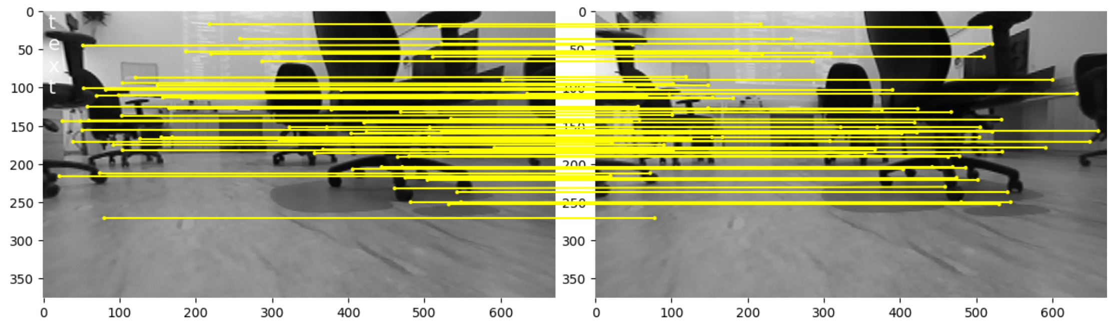

Figure 4.11: SuperGlue feature matching.

and rAMDL are substantially faster than iSAM2 compared to the simulated experiment. The run-time of iSAM2 depends on the degree of connectivity of the factor graph, more connections are a results of the same landmarks being observed from multiple poses, which in turn, results in a denser R matrix. Generally, a denser prior R matrix would require more QR operations to update a posterior R matrix, and thus, resulting in a longer time for iSAM2 to perform the R update before obtaining the R determinant, which is relatively fast (as it is linear in the dimension of the state). Since rAMDL and MP are not affected by the density of the prior R matrix, we conjecture that this difference in running time can be explain by the difference in R sparsity between the simulated and real experiments. The sparsity difference occurs since the real experiment relies on visual measurements, in this setting, the same features tend to be observed at many consecutive poses, which results in higher degree of factor graph connectivity.

Method	time [sec]
MP (ours)	$\mathbf{523.507 \pm 27.153}$
rAMDL	740.545 ± 25.651
iSAM2	2111.835 ± 26.521

Table 4.4: Total planning time in seconds (lower is better), number of factors in the prior graph is 8220, number of paths evaluated is 1000.

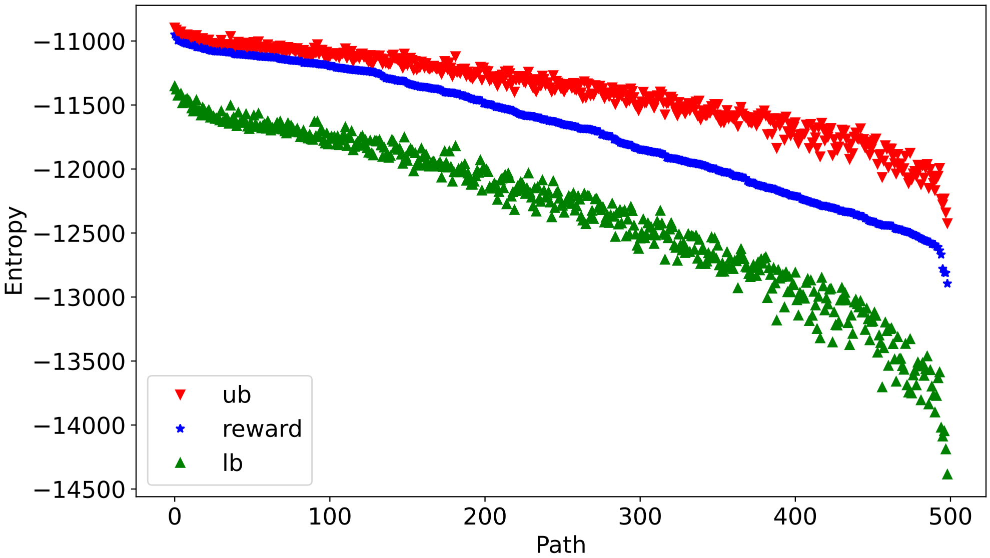

Figure 4.12: Robotmaster experiment.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

This work has presented the novel concept of observation space partitioning for BSP and POMDP planning problems. The concept is general and applies to all belief distributions and the underlying POMDP spaces. The partitioning of the underlying observation space allows for an inherent speed up in planning. The partitioned space allows for a more efficient way of identifying the optimal expected reward by using a simplification paradigm and forming analytical bounds on the expected reward. We have extended the idea of partitioning and introduced a partition tree which encodes a hierarchical partitioning of observations, where each level of partitioning We have demonstrated one possible use-case of this concept by studying a typical SLAM scenario with Gaussian beliefs. We have shown that both for simulated and real-world experiments, our method has a faster running time when compared to other SOTA methods, while reaching the same optimal reward.

5.2 Future work

There are multiple different direction to extending this work, which include; implementations for non-parametric belief distributions which would include an estimator for the expected reward function, an extension to policies over action-sequences, and the possible partitioning of other POMDP spaces such as the action space.

Bibliography

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

[2] M. Barenboim and V. Indelman. Adaptive information belief space planning. In *the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI)*, July 2022.

[3] Nicholas Carlevaris-Bianco, Michael Kaess, and Ryan M Eustice. Generic node removal for factor-graph SLAM. *IEEE Trans. Robotics*, 30(6):1371–1385, 2014.

[4] A.J. Davison. Active search for real-time vision. In *Intl. Conf. on Computer Vision (ICCV)*, pages 66–73, Oct 2005.

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

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

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

[8] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[9] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31(2):217–236, Feb 2012.

[10] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, Dec 2008.

[11] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, 1996.

[12] Kasra Khosoussi, Gaurav S. Sukhatme, Shoudong Huang, and Gamini Dissanayake. *Designing Sparse Reliable Pose-Graph SLAM: A Graph-Theoretic Approach*, pages 17–32. Springer International Publishing, 2020.

[13] A. Kim and R. M. Eustice. Active visual SLAM for robotic area coverage: Theory and experiment. *Intl. J. of Robotics Research*, 34(4-5):457–475, 2014.

[14] D. Kopitkov and V. Indelman. No belief propagation required: Belief space planning in high-dimensional state spaces via factor graphs, matrix determinant lemma and re-use of calculation. *Intl. J. of Robotics Research*, 36(10):1088–1130, August 2017.

[15] Dmitry Kopitkov and Vadim Indelman. General purpose incremental covariance update and efficient belief space planning via factor-graph propagation action tree. *Intl. J. of Robotics Research*, 38(14):1644–1673, 2019.

[16] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. of Machine Learning Research*, 9:235–284, 2008.

[17] H. Kretzschmar and C. Stachniss. Information-theoretic compression of pose graphs for laser-based SLAM. *Intl. J. of Robotics Research*, 31(11):1219–1230, 2012.

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

[19] R. Platt, R. Tedrake, L.P. Kaelbling, and T. Lozano-Pérez. Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems (RSS)*, pages 587–593, Zaragoza, Spain, 2010.

[20] Marija Popovic, Gregory Hitz, Juan Nieto, Inkyu Sa, Roland Siegwart, and Enric Galceran. Online informative path planning for active classification using uavs. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017.

[21] Nicholas Roy, Geoffrey J Gordon, and Sebastian Thrun. Finding approximate pomdp solutions through belief compression. *J. Artif. Intell. Res.(JAIR)*, 23:1–40, 2005.

[22] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4938–4947, 2020.

[23] T. Smith and R. Simmons. Heuristic search value iteration for pomdps. In *Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 520–527, 2004.

[24] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. In *NIPS*, volume 13, pages 1772–1780, 2013.

[25] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Robotics: Science and Systems (RSS)*, pages 65–72, 2005.

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

[27] J. Van Den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *Intl. J. of Robotics Research*, 31(11):1263–1278, 2012.

[28] Guangcong Zhang and Patricio A. Vela. Good features to track for visual slam. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1373–1382, 2015.

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

תקציר

סוכנים אוטונומיים פועלים בהינתן מידע לא מושלם, בין אם על סביבתם, על יחסי הגומלין הדינמיים שלהם עם הסביבה או על המדידות המתקבלות. כדי לתכנן אל העתיד במצב כזה, על הסוכן לפתור בעיה בסיסית הידועה בשם תהליך החלטה מרקובי ניתן לצפייה חלקית (POMDP). מכיוון שהמצב האמיתי של הסוכן אינו נגיש לו, וכחלק מהמודל של הבעיה, על הסוכן לתחזק התפלגות על המצב, הידועה בשם אמונה (Belief). התפלגות זו מתארת את סיכויי הסוכן לאכלס מצב מסוים במרחב המצבים האפשריים. על ידי שימוש בפילוג במקום התחייבות למצב אחד יחיד, יכול הסוכן לכמת את אי-הוודאות עבור תרחיש ספציפי. לסוכן יש גישה לשני מודלים שבהם הוא משתמש כדי לעדכן אמונה זאת בצורה בייסאנית, בהינתן מידע חדש. מודל אחד הוא מודל התנועה שמתאר את ההסתברות לעבור ממצב אחד לאחר, ומודל מדידה שמתאר את ההסתברות לקבל מדידות בהינתן מצב נתון. בבעיית התכנון, על הסוכן למצוא את סט פעולות אופטימלי שיביא לכך שפונקציית גמול מסוימת מקבלת ערך מקסימלי. הגידול המעריכי המהיר של אמונות עתידית, הנובע ממספר גדול של אפשרויות לפעולות ותצפיות עתידיות, משייך את בעיית התכנון למחלקת הבעיות מסוג NP-קשה.

מכיוון שרוב בעיות התכנון בעולם האמיתי אינן פתירות, עלינו לפתור בעיה מקורבת, למשל על ידי פישוט הבעיה. באופן כללי, פישוט בעיית התכנון מנסה לפשט חלקים מסוימים בבעיה, כך שאותו פתרון אופטימלי יתקבל, אך עבור זמני חישוב קצרים יותר. בשיטת פישוט כללית, ננסה למצוא חסמים אנליטיים על תוחלת פונקציית הגמול. בהנחה שהחסמים הינם יותר זולים לחישוב, נוכל להאיץ את תהליך התכנון על ידי שימוש בחסמים מבלי לחשב את תוחלת פונקציית הגמול באופן מפורש. השימוש בחסמים נעשה על ידי אלימינציה של מסלולים עתידיים שהינם תת-אופטימליים. אם ניקח לדוגמה 2 מסלולים, עבורם נחשב חסם עליון ותחתון עבור תוחלת פונקציית הגמול של כל מסלול, נוכל להגיד בוודאות כי מסלול אחד הינו אופטימלי ביחס למסלול השני, אם החסם התחתון של מסלול הראשון הינו גבוה יותר מהחסם העליון של המסלול השני.

במחקר זה, אנו מציגים גישה חדשנית לפישוט של בעיית התכנון, ספציפית על ידי פישוט מרחב המדידות מעליו מוגדר מודל המדידה. בגישה זו, מרחב המדידות עובר חלוקה לסטים נפרדים של מדידות אפשריות. על ידי שימוש בסטים הללו, אנו מציגים חסמים אנליטיים על תוחלת פונקציית הגמול מסוג אנטרופיה, שהיא מדד לאי-הוודאות המוכל בתוך האמונה של הסוכן על המצב. החסמים הללו הינם כלליים ומתאימים לכל פילוג של האמונה, בין אם הוא פרמטרי או לא. בנוסף, הם ניתנים לשינויי כך שיהיו יותר, או פחות, הדוקים ביחס לתוחלת פונקציית הגמול המקורית וכן הם מתכנסים לערך המקורי של תוחלת זו. כמו כן, אנחנו מציגים כיצד ניתן לעשות חלוקה הירכית של המדידות, כך שכל סט מחולק לתתי-סטים. אנו מציגים את הקידוד עבור ההיררכיה הזו בעץ של סטים, ועושים בה שימוש על ידי חסמים מתאימים על תוחלת פונקציית הגמול הצפויה. אנו מציגים איך ככל שנרד לעומק עץ זה, נוכל לקבל חסמים שהינם יעילים יותר.

לבסוף, אנו מדגימים כיצד ניתן לפתור בעיית תכנון עבור המקרה בו פילוג האמונה הינו נורמלי ומרחב המצב הינו רב-ממדי. אנו מדגימים את יעילות החישוב עבור תרחיש מדומה כאשר אנחנו משווים את השיטה שלנו לשיטות החדישות ביותר (SOTA). לאחר מכן אנחנו מדגימים את השיטה שלנו בניסוי אמיתי, על רובוט שמשתמש במצלמה על מנת לקבל מדידות מהסביבה. בכל המקרים אנחנו מראים שהשיטה שלנו מהירה משמעותית מכל שאר השיטות החדישות.

המחקר בוצע בהנחייתו של פרופסור חבר ואדים אינדלמן, בפקולטה למתמטיקה שימושית.

חלק מן התוצאות בחיבור זה פורסמו או הוגשו כמאמרים מאת המחבר ושותפיו למחקר בכנסים ובכתבי-עת במהלך תקופת מחקר המאסטר של המחבר, אשר גרסאותיהם העדכניות ביותר הינן:

T. Yotam and V. Indelman. Measurement Simplification in BSP with Performance Guarantees. In *Israel Conference Robotics (ICR)*, 2023. Accepted.

T. Yotam and V. Indelman. Measurement Simplification in ρ-POMDP with Performance Guarantees. *In IEEE Transactions on Robotics (T-RO),* 2023. To be submitted.

מחבר/ת חיבור זה מצהיר/ה כי המחקר, כולל איסוף הנתונים, עיבודם והצגתם, התייחסות והשוואה למחקרים קודמים וכו׳, נעשה כולו בצורה ישרה, כמצופה ממחקר מדעי המבוצע לפי אמות המידה האתיות של העולם האקדמי. כמו כן, הדיווח על המחקר ותוצאותיו בחיבור זה נעשה בצורה ישרה ומלאה, לפי אותן אמות מידה.

אני מודה לטכניון על התמיכה הנדיבה במשך השתלמותי.

פישוט מדידות בתהליך החלטה מרקובי ניתן לצפייה חלקית

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים במתמטיקה שימושית

תום יותם

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
תמוז התשפ״ג חיפה יוני 2023

פישוט מדידות בתהליך
החלטה מרקובי ניתן
לצפייה חלקית

תום יותם