Efficient Belief Space Planning in High-dimensional State Spaces by Exploiting Sparsity and Calculation Re-use

Dmitry Kopitkov

Efficient Belief Space Planning in High-dimensional State Spaces by Exploiting Sparsity and Calculation Re-use

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Technion Autonomous
Systems Program (TASP)

Dmitry Kopitkov

Submitted to the Senate
of the Technion — Israel Institute of Technology
Adar 5777 Haifa March 2017

This research was carried out under the supervision of Assistant Prof. Vadim Indelman, in the Faculty of Aerospace.

Acknowledgements

I would like to hugely thank my advisor Prof. Vadim Indelman for all his guidance, help, support and patience through the years of my Master's studies. My research skills got elevated significantly all thanks to his excellent schooling and high-standard demands.

I would also like to thank the people from Technion Autonomous Systems Program, especially so Sigalit Preger, for helping me getting over the bureaucracy of the university and for being supportive through sometimes very uneasy moments during the last years.

Finally, I thank all my friends for having my back and being there for me this entire time. Your help allowed me to keep standing on my legs and to continue and accomplish this research.

The generous financial help of the Technion is gratefully acknowledged.

Contents

List of Figures

List of Tables

Abstract **1**

Abbreviations and Notations **3**

1 Introduction **5**
- 1.1 Related Work . . . 6
- 1.2 Contributions . . . 7
- 1.3 Organization . . . 8

2 Notations and Problem Formulation **11**

3 Approach **17**
- 3.1 BSP as Factor Graph . . . 17
- 3.2 BSP via Matrix Determinant Lemma . . . 21
  - 3.2.1 Unfocused Case . . . 21
  - 3.2.2 Focused Case . . . 23
- 3.3 Augmented BSP via AMDL . . . 25
  - 3.3.1 Augmented Matrix Determinant Lemma (AMDL) . . . 25
  - 3.3.2 Unfocused Augmented BSP through IG . . . 27
  - 3.3.3 Focused Augmented BSP . . . 29
- 3.4 Re-use Calculations Technique . . . 33
- 3.5 Connection to Mutual Information Approach and Theoretical Meaning of IG . . . 35
- 3.6 Mutual Information - Fast Calculation via Information Matrix . . . 38

4 Application to Different Problem Domains **39**
- 4.1 Sensor Deployment . . . 39
- 4.2 Augmented BSP in Unknown Environments . . . 43
- 4.3 Graph Reduction . . . 49

5 Alternative Approaches **51**

6 Results **53**
6.1 Sensor Deployment (*focused* and *unfocused*) 53
6.2 Measurement Selection in SLAM 57
6.3 Autonomous Navigation in Unknown Environment 58

7 Conclusions and Future Work **65**
7.1 Future Work 65

8 Appendix - Related Publications **67**

9 Appendix - Proof of Lemmas **69**
9.1 Proof of Lemma 3.3.1 69
9.2 Proof of Lemma 3.3.2 70
9.3 Proof of Lemma 3.3.3 71
9.4 Proof of Lemma 3.3.4 72
9.5 Proof of Lemma 3.3.5 73
9.6 Proof of Lemma 3.3.6 74
9.7 SLAM Solution - focus on last pose $X^F_{k+L} \equiv x_{k+L}$ 75
9.8 SLAM Solution - focus on mapped landmarks $X^F_{k+L} \equiv L_k$ 77
9.9 Proof of Lemmas 3.6.1 and 3.6.2 80

Hebrew Abstract **i**

List of Figures

2.1 Illustrution of Λ_{k+L}'s construction for a given candidate action in Augmented BSP case. First, Λ_{k+L}^{Aug} is created by adding n' zero rows and columns. Then, the new information of belief is added through $\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + A^T A$. 13

3.1 Illustration of belief propagation in factor graph representation - not-augmented case. Two actions a_i and a_j are considered, introducing into graph two new factor sets $\mathcal{F}(a_i)$ and $\mathcal{F}(a_j)$ respectively (colored in green). 18

3.2 Illustration of belief propagation in factor graph representation - augmented case. Two actions a_i and a_j are considered, introducing their own factor graphs $G(a_i)$ and $G(a_j)$ (colored in pink) that are *connected* to prior G_k through factor sets $\mathcal{F}^{conn}(a_i)$ and $\mathcal{F}^{conn}(a_j)$ (colored in green) respectively. 19

3.3 Concept illustration of A's structure. Each column represents some variable from state vector. Each row represents some factor from Eq. (2.6). Here, A represents set of factors $\mathcal{F} = \{f_1(x_{i-1}, x_i), f_2(x_i, l_j)\}$ where factor f_1 of motion model that involves two poses x_i and x_{i-1} will have non-zero values only at columns of x_i and x_{i-1}. Factor f_2 of observation model that involves together variables x_i and l_j will have non-zero values only at columns of x_i and l_j. 21

3.4 Partitions of Jacobians and state vector X_{k+L} in Augmented BSP case, `unfocused` scenario. Note: the shown variable ordering is only for illustration, while the developed approach supports *any* arbitrary variable ordering. Also note that all white blocks consist of only zeros. Top: Jacobian A of factor set $\mathcal{F}(a) = \{\mathcal{F}^{conn}(a), \mathcal{F}^{new}(a)\}$. Bottom: Jacobians B and D of factor sets $\mathcal{F}^{conn}(a)$ and $\mathcal{F}^{new}(a)$ respectively. 25

3.5 Partitions of Jacobians and state vector X_{k+L} in Augmented BSP case, `Focused` ($X_{k+L}^F \subseteq X_{new}$) scenario. Note: the shown variable ordering is only for illustration, while the developed approach supports *any* arbitrary variable ordering. Also note that all white blocks consist of only zeros. Top: Jacobian A of factor set $\mathcal{F}(a) = \{\mathcal{F}^{conn}(a), \mathcal{F}^{new}(a)\}$. Bottom: Jacobians B and D of factor sets $\mathcal{F}^{conn}(a)$ and $\mathcal{F}^{new}(a)$ respectively. 30

3.6 Partitions of Jacobians and state vector X_{k+L} in Augmented BSP case, `Focused` ($X^F_{k+L} \subseteq X_{old}$) scenario. Note: the shown variable ordering is only for illustration, while the developed approach supports *any* arbitrary variable ordering. Also note that all white blocks consist of only zeros. Top: Jacobian A of factor set $\mathcal{F}(a) = \{\mathcal{F}^{conn}(a), \mathcal{F}^{new}(a)\}$. Bottom: Jacobians B and D of factor sets $\mathcal{F}^{conn}(a)$ and $\mathcal{F}^{new}(a)$ respectively. 32

4.1 Illustration of belief propagation in factor graph representation - Sensor Deployment scenario. The space is discretized through grid of locations $L \doteq \{l_1, \ldots, l_9\}$. Factor f_0 within prior factor graph G_k represents our prior belief about state vector, $\mathbb{P}_0(X)$. Factors $f_1 - f_3$ represent measurements taken from sensors deployed at locations l_1, l_2 and l_4. Two actions $a_i = \{l^3, l^6\}$ and $a_j = \{l^7, l^8\}$ are considered, introducing into graph two new factor sets $\mathcal{F}(a_i)$ and $\mathcal{F}(a_j)$ respectively (colored in green). In this example value of c' is 2. 40

4.2 Illustration of belief propagation in factor graph representation - SLAM scenario. Nodes x_i represent robot poses, while nodes l_i - landmarks. Factor f_1 is prior on robot's initial position x_1; factors between robot poses represent motion model (Eq. (4.8)); factors between pose and landmark represent observation model (Eq. (4.9)). Two actions a_i and a_j are considered, performing loop-closure to re-observe landmark l_1 and l_2 respectively. Both actions introduce their own factor graphs $G(a_i)$ and $G(a_j)$ (colored in pink) that are *connected* to prior G_k through factor sets $\mathcal{F}^{conn}(a_i)$ and $\mathcal{F}^{conn}(a_j)$ (colored in green) respectively. . . . 43

6.1 `Unfocused` sensor deployment scenario. Running time for calculating impact of a single action as a function of state dimension n (a) and as a function of Jacobian A's height m (b). In (a), $m = 2$, while in (b) $n = 625$. *rAMDL* `Unfocused Objective` represents only calculation time of candidates' impacts (IG objective for all actions), without one-time calculation of prior covariance; *Covariance Inverse* represents the time it took to calculate covariance matrix Σ_k from dense information matrix Λ_k, $\Sigma_k = \Lambda_k^{-1}$. (c) Running time for sequential decision making, i.e. evaluating impact of all candidate actions, each representing candidate locations of 2 sensors. (d) prior and final uncertainty of the field, with red dots marking selected locations. (e) number of action candidates per decision. (f) running time for sequential decision making, with number of candidates limited to 100. 55

6.2 `Focused` sensor deployment scenario, (a) overall time it took to make decision with different approaches; *rAMDL* `Focused Objective` represents only calculation of candidates' impacts (IG objective for all actions) while *rAMDL* `Focused` - both one-time calculation of prior covariance Σ_k and candidates' evaluation. (b) Final uncertainty of the field, with red dots marking selected locations. (c) `Focused` set of variables (green circles) and locations selected by algorithm (red dots). (d) Overall system entropy (above) and entropy of `focused` set (bottom) after each decision, with blue line representing `unfocused` algorithm, and red line - `focused` algorithm. Note - all `unfocused` methods make exactly the same decisions, with difference only in their runtime complexity. Same is also true for all `focused` methods. 56

6.3 Measurement selection scenario, (a) simulated trajectory of robot; black dots are the landmarks, blue marks and surrounding ellipses are the estimated trajectory along with the uncertainty covariance for each time instant, red mark is robot's initial pose; (b) number of measurement candidates per decision; (c) state's dimension n per decision; (d) overall time it took to evaluate impacts of pose's all measurements, with different approaches; *rAMDL* `Unfocused Objective` represents only calculation of candidates' impacts (IG objective for all actions) while *rAMDL* `Unfocused` - both one-time calculation of marginal covariance $\Sigma_k^{M,X_{All}}$ and candidates' evaluation. 57

6.4 `Focused` BSP scenario with `focused` robot's last pose. (a) Dimensions of the BSP problem (state dimension, average number of new factor terms, average number of new variables, average number of old involved variables) at each time; (b) Number of action candidates at each time; (c) Final robot trajectory. Blue dots are mapped landmarks, red line with small ellipses is estimated trajectory with pose covariances, blue line is the real trajectory, red pluses with numbers beside them are robot's goals. Green mark is robot's start position; (d) Zoom-in of robot's trajectory near goal 12. 59

6.5 `Focused` BSP scenario with `focused` robot's last pose. (a) Running time of planning, i.e. evaluating impact of all candidate actions, each representing possible trajectory; Results are shown both for `focused` and `unfocused` cases; (b) Zoom of fastest approaches from (a); (c) `Focused` approaches from (b). Note that *iSAM* `Focused` is not depicted because as seen in (a) it is much slower comparing to other `focused` techniques; (d) `Unfocused` approaches from (b). The lowest line, labeled *Marginal Cov*, represents time it took to calculate prior marginal covariance $\Sigma_k^{M,X_{All}}$ in *rAMDL* approach (see Section 3.4). As can be seen, while *rAMDL* technique (`Unfocused` and `Focused`) is faster than *From-Scratch* and *iSAM*, the *rAMDL-Extended* gives even better performance. Further, it is interesting to note that performance of `Unfocused` and `Focused` *rAMDL* is almost the same, as also performance of `Unfocused` and `Focused` *rAMDL-Extended* 60

6.6 `Focused` BSP scenario with `focused` robot's last pose. Running times from Figure 6.5 normalized by number of candidates. 61
6.7 `Focused` BSP scenario with `focused` landmarks. (a) Number of action candidates at each time; (b) Final robot trajectory; (c) Running time of planning, i.e. evaluating impact of all candidate actions, each representing possible trajectory; (d) Running time from (c) normalized by number of candidates; (e) Zoom of fastest approaches from (c); (f) Zoom of fastest approaches from (d). The lowest line, labeled *Marginal Cov*, represents time it took to calculate prior marginal covariance $\Sigma_k^{M,X_{All}}$ in *rAMDL* approach (see Section 3.4). 62
6.8 `Focused` BSP scenario with `focused` robot's last pose, using Victoria Park dataset. (a) Number of action candidates at each time; (b) Final robot trajectory; (c) Running time of planning, i.e. evaluating impact of all candidate actions, each representing possible trajectory; (d) Running time from (c) normalized by number of candidates; (e) Zoom of fastest approaches from (c); (f) Zoom of fastest approaches from (d). The lowest line, labeled *Marginal Cov*, represents time it took to calculate prior marginal covariance $\Sigma_k^{M,X_{All}}$ in *rAMDL* approach (see Section 3.4). 63

List of Tables

3.1 Different partitions of state variables in BSP 21
3.2 Different partitions of state variables in Augmented BSP 27
3.3 Different problems and required entries of prior covariance. BSP - short for non-augmented belief space planning, Augmented BSP - augmented belief space planning. 33

6.1 Considered approaches in different problems from Chapter 6, along with their appropriate equations. 54

Abstract

Belief space planning (BSP) is a fundamental problem in robotics and artificial intelligence, with applications including autonomous navigation and active SLAM. The information-theoretic BSP is its sub-problem where objective is to find action which will minimize the posterior uncertainty of the state. In order to solve this problem, the state of the art approaches typically propagate the belief state, for each candidate action, through calculation of the posterior information (or covariance) matrix and subsequently compute its determinant (required for entropy). The per-candidate time-complexity of such approaches is $O(n^3)$ where n is the state dimension and typically is very large, making such approaches to be very computationally expensive.

In this research we develop a computationally efficient approach for evaluating the information theoretic term within belief space planning (BSP), where during belief propagation the state vector can be constant or augmented. We consider both unfocused and focused problem settings, whereas uncertainty reduction of the entire system or only of chosen variables is of interest, respectively. Our approach reduces run-time complexity by avoiding posterior belief propagation and determinant calculation of large matrices. We formulate the problem in terms of factor graphs and show that belief propagation is not needed, requiring instead a one-time calculation that depends on (the increasing with time) state dimensionality, and per-candidate calculations that are *independent* of the latter. To that end, we develop an augmented version of the matrix determinant lemma, and show computations can be re-used when evaluating impact of different candidate actions. These two key ingredients and the factor graph representation of the problem result in a computationally efficient (augmented) BSP approach that accounts for different sources of uncertainty and can be used with various sensing modalities. We examine the unfocused and focused instances of our approach, and compare it to the state of the art, in simulation and using real-world data, considering problems such as autonomous navigation in unknown environments, measurement selection and sensor deployment, carried out at the Autonomous Navigation and Perception Lab at the Technion. We show that our approach significantly reduces running time without any compromise in performance.

Abbreviations and Notations

BSP	:	Belief Space Planning
POMDP	:	Partially Observable Markov Decision Process
SLAM	:	Simultaneous Localisation and Mapping
IG	:	Information Gain
MAP	:	Maximum A Posteriori (estimation)
AMDL	:	Augmented Matrix Determinant Lemma
rAMDL	:	Re-use of calculations and Augmented Matrix Determinant Lemma
iSAM	:	Incremental Smoothing and Mapping
GT-SAM	:	Georgia Tech-Smoothing and Mapping
X_k	:	state vector at time k
X_{k+L}	:	state vector at future time $k + L$
${}^{I}X$	:	subset of X_{k+L} with variables *involved* in new terms in Eq. (2.6)
${}^{\neg I}X$	:	subset of X_{k+L} with variables *not involved* in new terms in Eq. (2.6)
X^F	:	subset of X_{k+L} with `focused` variables
X^U	:	subset of X_{k+L} with `unfocused` variables
${}^{I}X^U$	:	subset of X^U with variables *involved* in new terms in Eq. (2.6)
${}^{\neg I}X^U$	:	subset of X^U with variables *not involved* in new terms in Eq. (2.6)
$b[X]$	:	belief of state vector X, its probability density function
Σ_k	:	covariance matrix of state vector X_k
Λ_k	:	information matrix of state vector X_k, which is inverse of matrix Σ_k
$\widetilde{A}$	:	Jacobian matrix of new factors, introduced by candidate action
A	:	noise-weighted Jacobian matrix of new factors, introduced by candidate action
$\mathcal{H}(\cdot)$	:	differential entropy

Chapter 1

Introduction

Decision making under uncertainty and belief space planning are fundamental problems in robotics and artificial intelligence, with applications including autonomous driving, surveillance, sensor deployment, object manipulation and active SLAM. The goal is to autonomously determine best actions according to a specified objective function, given the current belief about random variables of interest that could represent, for example, robot poses, tracked target or mapped environment, while accounting for different sources of uncertainty.

Since the true state of interest is typically unknown and only partially observable through acquired measurements, it can be only represented through a probability distribution conditioned on available data. Belief space planning (BSP) and decision making approaches reason how this distribution (the *belief*) evolves as a result of candidate actions and future expected observations. Such a problem is an instantiation of partially observable Markov decision process (POMDP), while calculating an optimal solution of POMDP was proven to be computationally intractable [22] for all but the smallest problems due to curse of history and curse of dimensionality. Recent research has therefore focused on the development of sub-optimal approaches that trade-off optimality and runtime complexity.

Decision making under uncertainty, also sometimes referred to as active inference, and BSP can be formulated as selecting optimal action from a set of candidates, based on some cost function. In information-based decision making the cost function typically contains terms that evaluate the expected posterior uncertainty upon action execution, with commonly used costs including (conditional) entropy and mutual information. Thus, for Gaussian distributions the corresponding calculations typically involve calculating a determinant of a posteriori covariance (information) matrices, and moreover, these calculations are to be performed for *each* candidate action.

Decision making and BSP become an even more challenging problems when considering *high* dimensional state spaces. Such a setup is common in robotics, for example in the context of belief space planning in uncertain environments, active SLAM, sensor deployment, graph reduction and graph sparsification. In particular, calculating a determinant of information (covariance) matrix for an n-dimensional state is in general $O(n^3)$, and is smaller for sparse matrices as in SLAM problems [3].

Moreover, state of the art approaches typically perform such time-consuming calculations from scratch for *each* candidate action. In contrast, in this work we provide a novel way to perform information-theoretic BSP, which is fast, simple and general; yet, it does not require calculation of a posterior belief and does not need determinant computation of large matrices. Additionally, we have succeeded to reuse calculations between different candidate actions, eventually providing decision making solver which is significantly faster compared to standard approaches.

1.1 Related Work

As mentioned above, the optimal solution to POMDP is computationally intractable and many approximation approaches exist to solve it in sub-optimal way. These approaches can be classified into those that discretize the action, state and measurement spaces, and those that operate over continuous spaces.

Approaches from the former class include point-based value iteration methods [36], simulation based [39] and sampling based approaches [1, 38]. On the other hand, approaches that avoid discretization are often termed direct trajectory optimization methods (e.g. [21, 34, 37, 41, 43]); these approaches typically calculate from a given nominal solution a locally-optimal one.

To solve the BSP problem, standard methods usually perform expensive calculations for each candidate action from scratch. For example, in the context of active SLAM, state of the art BSP approaches first calculate the posterior belief within the planning horizon, and then use that belief to evaluate the objective function, which typically includes an information-theoretic term [15, 21, 26, 40]. These approaches then determine the best action by performing the mentioned calculations for each action from a given set of candidate actions, or by local search using dynamic programming or gradient descent (for continuous setting).

Sensor deployment is another example of decision making in high dimensional state spaces. The basic formulation of the problem is to determine locations to deploy the sensors such that some metric can be measured most accurately through the entire area (e.g. temperature). The problem can also be viewed as selecting optimal action from the set of candidate actions (available locations) and the objective function usually contains a term of uncertainty, like the entropy of a posterior system [27]. Also here, state of the art approaches evaluate a determinant over large posterior covariance (information) matrices for each candidate action, and do so from scratch [44, 45].

A similar situation also arises in measurement selection [5,9] and graph pruning [4,14,31,42] in the context of long-term autonomy in SLAM. In the former case, the main idea is to determine the most informative measurements (e.g. image features) given measurements provided by robot sensors, thereby discarding uninformative and redundant information. Such a process typically involves reasoning about mutual information, see e.g. [7, 9], for each candidate selection. Similarly, graph pruning and sparsification can be considered as instances of decision making in high dimensional state spaces [4, 14], with decision corresponding to determining what nodes to marginalize out [16, 28], and avoiding the resulting fill-in in information matrix by resorting

to sparse approximations of the latter [4, 14, 31, 42]. Also here, existing approaches typically involve calculation of determinant of large matrices for each candidate action.

Although many particular domains can be specified as decision making and BSP problems, they all can be classified into two main categories, one where state vector is fixed during belief propagation and another where the state vector is augmented with new variables. Sensor deployment is an example of the first case, while active SLAM, where future robot poses are introduced into the state, is an example for the second case. Conceptually the first category is a particular case of the second, but as we will see both will require different solutions. Therefore, in order to differentiate between these two categories, in this research we will consider the first category (fixed-state) as *BSP* problem, and the second category (augmented-state) as *Augmented BSP* problem.

Moreover, we show the proposed concept is applicable also to active `focused` inference. Unlike the `unfocused` case discussed thus far, active `focused` inference approaches aim to reduce the uncertainty over only a predefined set of the variables. The two problems can have significantly different optimal actions, with an optimal solution for the `unfocused` case potentially performing badly for the `focused` setup, and vice versa (see e.g. [30]). While the set of `focused` variables can be small, exact state of the art approaches calculate the marginal posterior covariance (information) matrix, for each action, which involves a computationally expensive Schur complement operation. For example, Mu et al. [32] calculate posterior covariance matrix per each measurement and then use the selection matrix in order to get marginal of `focused` set. Levine et al. [30] develop an approach that determines mutual information between `focused` and `unfocused` variables through message passing algorithms on Gaussian graphs but their approach is limited to only graphs with unique paths between the relevant variables.

Finally, there is also a relation to the recently introduced concept of decision making in a conservative sparse information space [19, 20]. In particular, considering unary observation models (involving only one variable) and greedy decision making, it was shown that appropriately dropping all correlation terms and remaining only with a diagonal covariance (information) matrix does not sacrifice performance while significantly reducing computational complexity. While the approach presented herein confirms this concept for the case of unary observation models, our approach addresses a general non-myopic decision making problem, with arbitrary observation and motion models.

1.2 Contributions

In this thesis we develop a computationally efficient and exact approach for decision making and BSP in high-dimensional state spaces that addresses the aforementioned challenges. The *key idea* is to use the (augmented) general matrix determinant lemma to calculate action impact with complexity *independent* of state dimensionality n, while *re-using* calculations between evaluating impact for different candidate actions. Our approach supports general observation and motion models, and non-myopic planning, and is thus applicable to a wide range of applications

such as those mentioned above, where fast decision making and BSP in high-dimensional state spaces is required.

For `focused` BSP scenarios we present a new way to calculate posterior entropy of `focused` variables, which is very computationally efficient, yet exact and does not require expensive calculation of a Schur complement and a posterior covariance matrix. In combination with our *re-use* algorithm, it provides `focused` decision making algorithm which is significantly faster compared to state of the art approaches.

Calculating the posterior information matrix in Augmented BSP problems involves augmenting an appropriate prior information matrix with zero rows and columns, i.e. zero padding, and then adding new information due to candidate action (see Figure 2.1). While the general matrix determinant lemma is an essential part of our approach, unfortunately it is not applicable to the mentioned augmented prior information matrix since the latter is singular (even though the posterior information matrix is full rank). In this thesis, we develop a new variant of the matrix determinant lemma, called the *augmented matrix determinant lemma* (AMDL), that addresses general augmentation of future state vector. Based on AMDL, we then develop a augmented belief space planning approach, considering both `unfocused` and `focused` cases.

To summarize, our main contributions in this research are as follows: (a) we formulate (augmented) belief space planning in terms of factor graphs which allow to see the problem in more intuitive and simple way; (b) we develop an augmented version of matrix determinant lemma (AMDL), where the subject matrix first is augmented by zero rows/columns and only then new information is introduced (c) we develop an approach for a nonmyopic `focused` and `unfocused` (augmented) belief space planning in high-dimensional state spaces that uses the (augmented) matrix determinant lemma to avoid calculating determinants of large matrices, with per-candidate complexity independent of state dimension; (d) we show how calculations can be *re-used* when evaluating impacts of different candidate actions; We integrate the calculations *re-use* concept and AMDL into a general and highly efficient BSP solver, that does not involve explicit calculation of posterior belief evolution for different candidate actions, naming this approach *rAMDL*; (e) we introduce an even more efficient *rAMDL* variant specifically addressing a sensor deployment problem.

1.3 Organization

This thesis is organized as follows.

1. Chapter 2 introduces the concepts of BSP, and gives a formal statement of the problem.
2. Chapter 3 describes our approach *rAMDL* for general formulation.
3. Chapter 4 tailors approach for specific domains, providing even more efficient solutions to number of them.
4. In Chapter 5 standard approaches are discussed as the main state-of-the-art alternatives to *rAMDL* .

5. Chapter 6 presents experimental results, evaluating the proposed approach and comparing it against mentioned state-of-the-art.

6. Conclusions are drawn in Chapter 7.

7. For purpose of simplicity, the proof of several lemmas is moved into Appendix 9.

Chapter 2

Notations and Problem Formulation

In this thesis we are developing computationally efficient approaches for belief space planning. As evaluating action impact involves inference over an appropriate posterior, we first formulate the corresponding inference problem.

Consider a high-dimensional problem-specific state vector $X_k \in \mathbb{R}^n$ at time t_k. In different applications the state X_k can represent robot configuration and poses (optionally for whole history), environment-related variables or any other variables to be estimated. Additionally, consider factors $F_i = \{f_i^1(X_i^1), \ldots, f_i^{n_i}(X_i^{n_i})\}$ that were added at time $0 \leq t_i \leq t_k$, where each factor $f_i^j(X_i^j)$ represents a specific measurement model, motion model or prior, and as such involves appropriate state variables $X_i^j \subseteq X_i$.

The joint pdf can be then written as

$$\mathbb{P}(X_k|\mathbf{H_k}) \propto \prod_{i=0}^{k} \prod_{j=1}^{n_i} f_i^j(X_i^j), \tag{2.1}$$

where $\mathbf{H_k}$ is history that contains all the information gathered till time t_k (measurements, controls, etc.).

As common in many inference problems, we will assume that all factors have a Gaussian form:

$$f_i^j(X_i^j) \propto \exp(-\frac{1}{2}\|h_i^j(X_i^j) - r_i^j\|^2_{\Sigma_i^j}), \tag{2.2}$$

with appropriate model

$$r_i^j = h_i^j(X_i^j) + \upsilon_i^j, \quad \upsilon_i^j \sim \mathcal{N}(0, \Sigma_i^j) \tag{2.3}$$

where h_i^j is a known nonlinear function, υ_i^j is zero-mean Gaussian noise and r_i^j is the expected value of h_i^j ($r_i^j = \mathbb{E}[h_i^j(X_i^j)]$). Such a factor representation is a general way to express information about the state. In particular, it can represent a measurement model, in which case, h_i^j is the observation model, and r_i^j and υ_i^j are, respectively, the actual measurement z and measurement noise. Similarly, it can also represent a motion model (see Section 4.2). A maximum a posteriori (MAP) inference can be efficiently calculated (see e.g. [24]) such that

$$\mathbb{P}(X_k|\mathbf{H_k}) = \mathcal{N}(X_k^*, \Sigma_k), \tag{2.4}$$

where X_k^* and Σ_k are the mean vector and covariance matrix, respectively.

We shall refer to the posterior $\mathbb{P}(X_k|\mathbf{H_k})$ as the belief and write

$$b[X_k] \doteq \mathbb{P}(X_k|\mathbf{H_k}). \tag{2.5}$$

In the context of BSP, we typically reason about the evolution of future beliefs $b[X_{k+l}]$ at different look-ahead steps l as a result of different candidate actions. Particular candidate action can provide unique information (future observations and controls) and can be more and less beneficial for specific tasks such as reducing future uncertainty. For example, in SLAM application choosing trajectory that is close to the mapped landmarks will reduce uncertainty because of loop-closures. Furthermore, conceptually each candidate action can introduce different additional state variables into the future state vector, like in case of smoothing SLAM formulation where state is augmented by (various) number of future robot poses.

Therefore, in order to reason about the belief $b[X_{k+l}]$, first it needs to be carefully modeled. More specifically, let us focus on a non-myopic candidate action $a \doteq \{\bar{a}_1, \ldots, \bar{a}_L\}$ which is a sequence of myopic actions with planning horizon L. Each action $\bar{a}_l$ can be represented by new factors $F_{k+l} = \{f_{k+l}^1(X_{k+l}^1), \ldots, f_{k+l}^{n_{k+l}}(X_{k+l}^{n_{k+l}})\}$ and, possibly, new state variables X_{new}^{k+l} $(1 \leq l \leq L)$ that are acquired/added while applying $\bar{a}_l$. Similar to Eq. (2.1), the future *belief* $b[X_{k+L}]$ can be explicitly written as

$$b[X_{k+L}] \propto b[X_k] \prod_{l=k+1}^{k+L} \prod_{j=1}^{n_l} f_l^j(X_l^j), \tag{2.6}$$

where $X_{k+L} \doteq \{X_k \cup X_{new}^{k+1} \cup \ldots \cup X_{new}^{k+L}\}$ contains old and new state variables. Similar expressions can be also written for any other look ahead step l. Observe in the above *belief* that the future factors depend on future observations, whose actual values are unknown.

It is important to note that, according to our definition from Chapter 1, new variables are added only in the augmented setting of the BSP problem, e.g. in the active SLAM context. On the other hand, in a non-augmented BSP setting, the states X_{k+L} and X_k are identical, while the beliefs $b[X_{k+L}]$ and $b[X_k]$ are still conditioned on different data. For example, in sensor deployment and measurement selection problems the candidate actions are all possible subsets of sensor locations and of acquired observations, respectively. Here, when applying a candidate action, new information about X_k is brought in, but the state vector itself is unaltered.

In contrast, in Augmented BSP problem new variables are always introduced. In particular, in both smoothing and filtering formulation of SLAM, candidate actions (trajectories) will introduce both new information (future measurements), and also new variables (future robot poses). While in *filtering* formulation old pose variables are marginalized out, the *smoothing* formulation instead keeps past and current robot poses and newly mapped landmarks in the state vector which is beneficial for better estimation accuracy and sparsity. As such, smoothing formulation is an excellent example for Augmented BSP problem, where as filtering formulation can be considered as `focused` BSP scenario which described below.

As such the non-augmented BSP setting can be seen as a special case of Augmented BSP. In order to use similar notations for both problems, however, in this thesis we will consider X_{new}^{k+l}

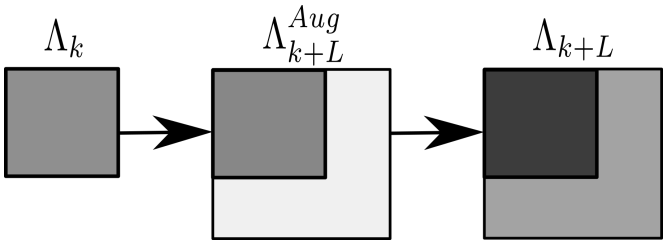

Figure 2.1: Illustrution of Λ_{k+L}'s construction for a given candidate action in Augmented BSP case. First, Λ_{k+L}^{Aug} is created by adding n' zero rows and columns. Then, the new information of belief is added through $\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + A^T A$.

to be an empty set for the former case and non-empty for Augmented BSP.

It is not difficult to show (see e.g. [21]) that in case of non-augmented BSP the posterior information matrix of the *belief* $b[X_{k+L}]$ is given by:

$$\Lambda_{k+L} = \Lambda_k + \sum_{l=k+1}^{k+L} \sum_{j=1}^{n_l} (H_l^j)^T \cdot (\Sigma_l^j)^{-1} \cdot H_l^j \tag{2.7}$$

where Λ_k is prior information matrix and $H_l^j \doteq \nabla_x h_l^j$ are the Jacobian matrices of h_l^j functions (see Eq. (2.2)) for all the new factor terms in Eq. (2.6).

As was already mentioned, in case of Augmented BSP the joint state X_{k+L} includes also new variables (with respect to the current state X_k). Considering $X_k \in \mathbb{R}^n$, first, new n' variables are introduced into future state vector $X_{k+L} \in \mathbb{R}^N$ with $N \doteq n + n'$, and then new factors involving appropriate variables from X_{k+L} are added to form a posterior *belief* $b[X_{k+L}]$, as shown in Eq. (2.6).

Consequently, in Augmented BSP scenario the posterior information matrix of *belief* $b[X_{k+L}]$, i.e. Λ_{k+L}, can be constructed by first augmenting the current information matrix Λ_k with n' zero rows and columns to get $\Lambda_{k+L}^{Aug} \in \mathbb{R}^{N \times N}$, and thereafter adding to it new information, as illustrated in Figure 2.1 (see e.g. [21]):

$$\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + \sum_{l=k+1}^{k+L} \sum_{j=1}^{n_l} (H_l^j)^T \cdot (\Sigma_l^j)^{-1} \cdot H_l^j \tag{2.8}$$

where $H_l \doteq \nabla_x h_l^j$ are augmented Jacobian matrices of all new factors in Eq. (2.6), linearized about the current estimate of X_k and about initial values of newly introduced variables.

After stacking all new Jacobians in Eqs. (2.7) and (2.8) into a single matrix $\widetilde{A}$, and combining all noise matrices into block-diagonal Ψ, we get respectively

$$\Lambda_{k+L} = \Lambda_k + \widetilde{A}^T \cdot \Psi^{-1} \cdot \widetilde{A} = \Lambda_k + A^T \cdot A \tag{2.9}$$

$$\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + \widetilde{A}^T \cdot \Psi^{-1} \cdot \widetilde{A} = \Lambda_{k+L}^{Aug} + A^T \cdot A \tag{2.10}$$

where

$$A \doteq \Psi^{-\frac{1}{2}} \cdot \widetilde{A} \tag{2.11}$$

is an $m \times N$ matrix that represents both Jacobians and noise covariances of all new factor terms in Eq. (2.6). The above equations can be considered as a single iteration of Gauss-Newton optimization and, similar to prior work [21, 26, 41], we take maximum-likelihood assumption by assuming they sufficiently capture the impact of candidate action. Under this assumption, the posterior information matrix Λ_{k+L} is independent of (unknown) future observations [21]. One can further incorporate reasoning if a future measurement will indeed be acquired [6, 21, 43]; however, this is outside the scope of this research.

Each block row of matrix A represents a single factor from new terms in Eq. (2.6) and has a *sparse* structure. Only a limited number of its sub-blocks is non-zero, i.e. sub-blocks that correspond to the *involved* variables X_l^j in the relevant factor $f_l^j(X_l^j)$.

For notational convenience, we define the set of non-myopic candidate actions by $\mathcal{A} = \{a_1, a_2, ...\}$ with appropriate Jacobian matrices $\Phi_A = \{A_1, A_2, ...\}$. While the planning horizon is not explicitly shown, each $a \in \mathcal{A}$ can represent a future *belief* $b[X_{k+L}]$ for different number of look ahead steps L.

A general objective function in decision making/BSP can be written as [21]:

$$J(a) \doteq \underset{Z_{k+1:k+L}}{\mathbb{E}} \left\{ \sum_{l=0}^{L-1} c_l(b[X_{k+l}], u_{k+l}) + c_L(b[X_{k+L}]) \right\}, \tag{2.12}$$

with L immediate cost functions c_l, for each look-ahead step, and one cost function for terminal future *belief* c_L. Each such cost function can include a number of different terms related to aspects such as information measure of future *belief*, distance to goal and energy spent on control. Arguably, evaluating the information terms involves the heaviest calculations of J.

Thus, in this research we will focus only on the information-theoretic term of terminal *belief* $b[X_{k+L}]$, and consider differential entropy $\mathcal{H}$ (further referred to just as entropy) and information gain (IG) as the cost functions. Both can measure amount of information of future *belief* $b[X_{k+L}]$, and will lead to the same optimal action. Yet, calculation of one is sometimes more efficient than the other, as will be shown in Chapter 3. Therefore, we consider two objective functions:

$$J_{\mathcal{H}}(a) \doteq \mathcal{H}\left(b[X_{k+L}]\right) \tag{2.13}$$

$$J_{IG}(a) \doteq \mathcal{H}(b[X_k]) - \mathcal{H}(b[X_{k+L}]), \tag{2.14}$$

where the information matrix Λ_{k+L}, that corresponds to the *belief* $b[X_{k+L}]$, is a function of candidate a's Jacobian matrix A, see Eq. (2.9) and Eq. (2.10). The optimal candidate a^*, which produces the most certain future *belief*, is then given by $a^* = \arg\min_{a \in \mathcal{A}} J_{\mathcal{H}}(a)$, or by $a^* = \arg\max_{a \in \mathcal{A}} J_{IG}(a)$ with both being mathematically identical.

In particular, for Gaussian distributions, entropy is a function of the determinant of a posterior information (covariance) matrix, i.e. $\mathcal{H}\left(b[X_{k+L}]\right) \equiv \mathcal{H}\left(\Lambda_{k+L}\right)$ and the objective functions can

be expressed as

$$J_{\mathcal{H}}(a) = \frac{n \cdot \gamma}{2} - \frac{1}{2} \ln \left|\Lambda_{k+L}\right|, \quad J_{IG}(a) = \frac{1}{2} \ln \frac{\left|\Lambda_{k+L}\right|}{\left|\Lambda_{k}\right|} \tag{2.15}$$

for BSP, and

$$J_{\mathcal{H}}(a) = \frac{N \cdot \gamma}{2} - \frac{1}{2} \ln \left|\Lambda_{k+L}\right|, \quad J_{IG}(a) = \frac{n' \cdot \gamma}{2} + \frac{1}{2} \ln \frac{\left|\Lambda_{k+L}\right|}{\left|\Lambda_{k}\right|} \tag{2.16}$$

for Augmented BSP, where $\gamma \doteq 1 + \ln(2\pi)$, and Λ_{k+L} can be calculated according to Eq. (2.9) and Eq. (2.10). Thus, evaluating J requires determinant calculation of an $n \times n$ (or $N \times N$) matrix, which is in general $O(n^3)$, per candidate action $a \in \mathcal{A}$. In many robotics applications state dimensionality can be huge and even increasing with time (e.g. SLAM), and straight forward calculation of the above equations makes real-time planning hardly possible.

So far, the exposition referred to `unfocused` BSP problems, where the action impact is calculated by considering *all* the random variables in the system, i.e. the entire state vector. However, as will be shown in the sequel, our approach is applicable also to `focused` BSP problems.

`Focused` BSP, in both augmented and non-augmented cases, is another important problem, where in contrast to the former case, only a subset of variables is of interest (see, e.g., [27, 30, 32]). For example one can look for action that reduces uncertainty of robot's final pose. The complexity of such a problem is much higher and proposed techniques succeeded to solve it in $O(kn^3)$ [27, 30] with k being size of candidate actions set, and in $O(\tilde{n}^4)$ [32] with $\tilde{n}$ being size of involved clique within Markov random field representing the system.

Considering posterior entropy over the `focused` variables $X^F_{k+L} \subseteq X_{k+L}$ we can write:

$$J^F_{\mathcal{H}}(a) = \mathcal{H}(X^F_{k+L}) = \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln \left|\Sigma^{M,F}_{k+L}\right|, \tag{2.17}$$

where n_F is the dimensionality of the state X^F_{k+L}, and $\Sigma^{M,F}_{k+L}$ is the posterior marginal covariance of X^F_{k+L} (suffix M for marginal), calculated by simply retrieving appropriate parts of posterior covariance matrix $\Sigma_{k+L} = \Lambda^{-1}_{k+L}$.

Solving the above problem in a straightforward manner involves $O(N^3)$ operations for each candidate action, where $N = n + n'$ is dimension of posterior system. In the following sections we develop a computationally more efficient approach that addresses both `unfocused` and `focused` (augmented) BSP problems. As will be seen, this approach naturally supports non-myopic actions and arbitrary factor models h^j_i, and it is in particular attractive to belief space planning in high-dimensional state spaces.

Chapter 3

Approach

Our approach, *rAMDL* , utilizes the well-known matrix determinant lemma [13] and re-use of calculations to significantly reduce computation of candidate action's impact, as defined in Chapter 2, for both augmented and non-augmented cases of BSP problem. In Section 3.1 we reformulate these problems in terms of factor graphs which will allow us to see another, more simplified, picture of the BSP problem. In Section 3.2.1 we develop a novel way to calculate the information-theoretic term for `unfocused` non-augmented BSP, and then extend it in Section 3.2.2 to the `focused` case. Additionally, in order to significantly reduce computational complexity of the Augmented BSP problem, as defined in Chapter 2, in Section 3.3.1 we extend the matrix determinant lemma for the matrix augmentation case. We then discuss in Sections 3.3.2-3.3.3 how this extension can be used within `unfocused` and `focused` Augmented BSP. Further, in Section 3.4 we discuss another key component of *rAMDL* - the re-use of calculations, which exploits the fact that many calculations can be shared among different candidate actions. Finally, in Section 3.5 we describe connection between our technique and mutual information approach from [9, 23], and discuss an interesting conceptual meaning of IG metric.

3.1 BSP as Factor Graph

The inference problem can be naturally represented by a factor graph [29], which is a bipartite graph $G = (\mathcal{F}, \Theta, \mathcal{E})$ with two node types: variables nodes $\theta_i \in \Theta$ and factor nodes $f_i \in \mathcal{F}$ (e.g. see Figures 3.1 and 3.2). Variable nodes represent state variables that need to be estimated, while factor nodes express different constraints between different variables. Each factor node is connected by edges $e_{ij} \in \mathcal{E}$ to variable nodes that are involved in the corresponding constraint. Such a formulation is general and can be used to represent numerous inference problems (e.g. SLAM), while exploiting sparsity. Furthermore, computationally efficient approaches, based on such formulation and exploiting its natural sparsity, have been recently developed [24, 25].

Below we will show that inference within BSP can also be formulated in terms of a factor graph.

The *belief* at time t_k, $b[X_k]$ can be represented by a factor graph $G_k = (\mathcal{F}_k, X_k, \mathcal{E}_k)$, where

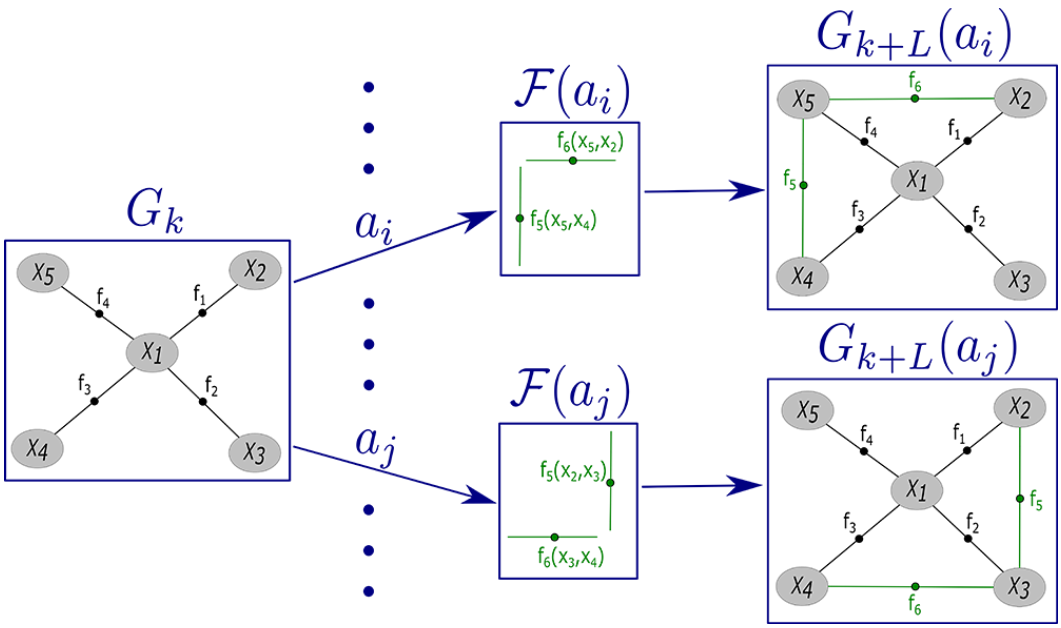

Figure 3.1: Illustration of belief propagation in factor graph representation - not-augmented case. Two actions a_i and a_j are considered, introducing into graph two new factor sets $\mathcal{F}(a_i)$ and $\mathcal{F}(a_j)$ respectively (colored in green).

with little abuse of notations, we use X_k to denote the estimated variables, $\mathcal{F}_k$ is the set of all factors acquired till time t_k, and where $\mathcal{E}_k$ encodes connectivity according to the variables X_i^j involved in each factor f_i^j, as defined in Eq. (2.2). The future *belief* $b[X_{k+L}]$ is constructed by introducing new variables and by adding new factors to the belief $b[X_k]$, as was shown in Chapter 2. Therefore, it can be represented by a factor graph which is an augmentation of the factor graph G_k, as will be shown below.

More specifically, in case of non-augmented BSP, let $\mathcal{F}(a) = \{f^1, \dots, f^{n_a}\}$ denote all the new factors from Eq. (2.6) introduced by action a, with n_a being the number of such factors. This abstracts the explicit time notations of factors inside Eq. (2.6) which in their turn can be seen as unimportant for solution of BSP problem. Then the factor graph of $b[X_{k+L}]$ is the prior factor graph G_k with newly introduced factor nodes $\mathcal{F}(a)$ connected to appropriate variable nodes (see Figure 3.1 for illustration). Thus, it can be denoted by $G_{k+L}(a)$:

$$G_{k+L}(a) = (\mathcal{F}_{k+L}, X_{k+L}, \mathcal{E}_{k+L}), \tag{3.1}$$

where $\mathcal{F}_{k+L} = \{\mathcal{F}_k, \mathcal{F}(a)\}$, $X_{k+L} \equiv X_k$ are unaltered state variables, and $\mathcal{E}_{k+L}$ represents connectivity between variables and factors according to definition of each factor (Eq. (2.2)).

For simplicity, we denote the augmentation of a factor graph G_k with a set of new factors through operator $\oplus$. Thus, for the non-augmented BSP setting, we have $G_{k+L}(a) \doteq G_k \oplus \mathcal{F}(a)$. Additionally, with a slight abuse of annotations we will use the same augmentation operator $\oplus$ to define combination of two factor graphs into one, which will be required in the context of augmented BSP.

In augmented BSP scenario, we denote all new state variables introduced by action a as

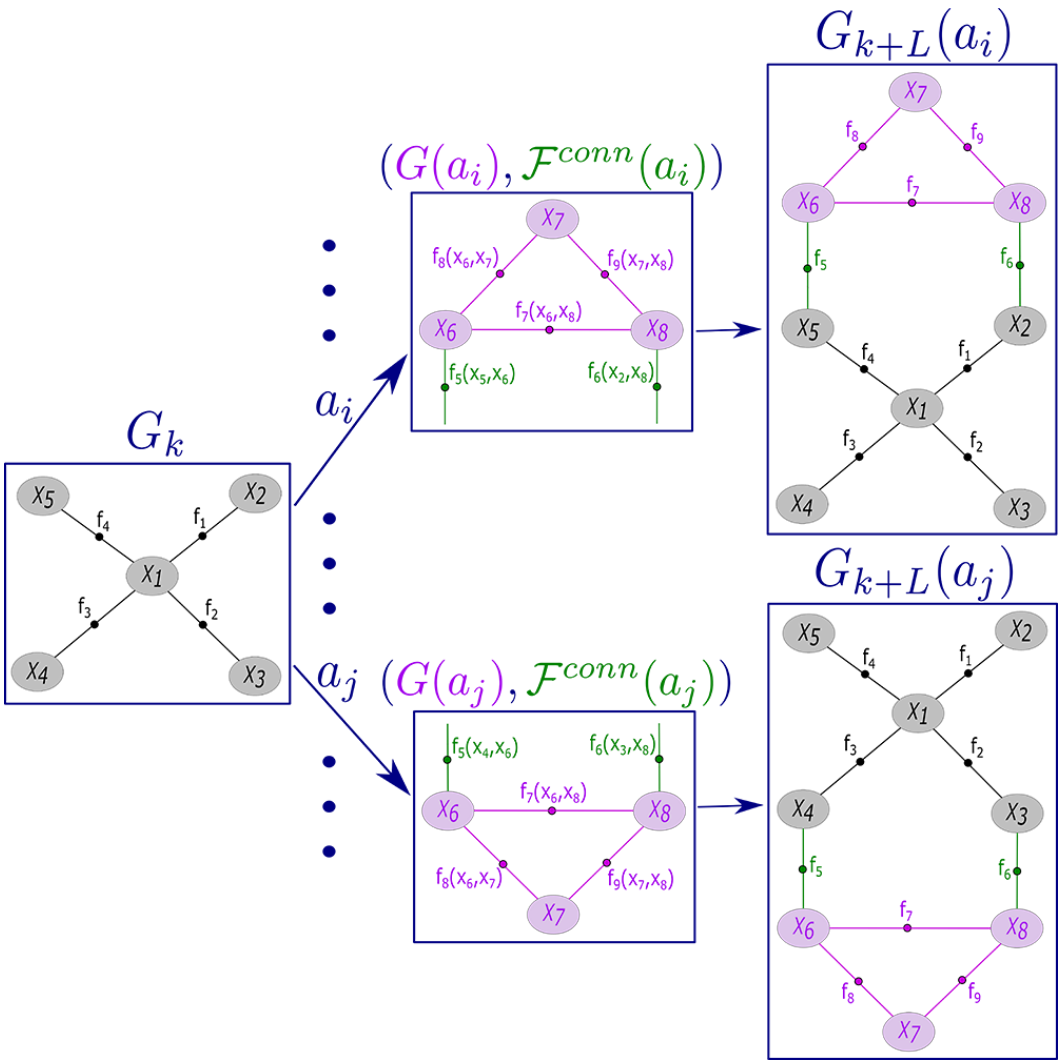

Figure 3.2: Illustration of belief propagation in factor graph representation - augmented case. Two actions a_i and a_j are considered, introducing their own factor graphs $G(a_i)$ and $G(a_j)$ (colored in pink) that are *connected* to prior G_k through factor sets $\mathcal{F}^{conn}(a_i)$ and $\mathcal{F}^{conn}(a_j)$ (colored in green) respectively.

X_{new}, and also separate all new factors $\mathcal{F}(a)$ from Eq. (2.6) into two groups:

$$\mathcal{F}(a) = \{\mathcal{F}^{new}(a), \mathcal{F}^{conn}(a)\}. \tag{3.2}$$

Factors connecting only new variables X_{new} are denoted by $\mathcal{F}^{new}(a)$:

$$\mathcal{F}^{new}(a) = \{f^1, \ldots, f^{n_{new}}\}, \tag{3.3}$$

while the rest of the factors are denoted by $\mathcal{F}^{conn}(a)$:

$$\mathcal{F}^{conn}(a) = \{f^1, \ldots, f^{n_{conn}}\}, \tag{3.4}$$

connecting between old and new variables.

Next, let us denote action's factor graph as $G(a) = (\mathcal{F}^{new}(a), X_{new}, \mathcal{E}_{new})$ with $\mathcal{E}_{new}$ repre-

senting connectivity according to *involved* variables in each factor in $\mathcal{F}^{new}(a)$. Then the factor graph that represents the future belief $b[X_{k+L}]$ is a combination of two factor graphs, the prior G_k and action's $G(a)$, *connected* by factors from $\mathcal{F}^{conn}(a)$ (see Figure 3.2 for illustration). Thus,

$$G_{k+L}(a) = G_k \oplus G(a) \oplus \mathcal{F}^{conn}(a) = (\mathcal{F}_{k+L}, X_{k+L}, \mathcal{E}_{k+L}), \tag{3.5}$$

where $\mathcal{F}_{k+L} = \{\mathcal{F}_k, \mathcal{F}^{new}(a), \mathcal{F}^{conn}(a)\}$, $X_{k+L} \equiv \{X_k, X_{new}\}$ is an augmented state vector, and $\mathcal{E}_{k+L}$ represents connectivity between variables and factors according to factors' definition. The separation of factors into two groups lets us to present future *belief* 's factor graph as simple graph augmentation, and will also be useful during derivation of our approach in Section 3.3. Moreover, the reason for $\mathcal{F}^{conn}(a)$ not to be defined as part of $G(a)$ is due to the fact that factors inside $\mathcal{F}^{conn}(a)$ involve state variables outside of $G(a)$.

Note that the new factors in $G_{k+L}(a)$ are not fully defined, as some of them involve future observations which are unknown at planning time. However, the taken maximum-likelihood assumption expects that mean vector of $b[X_{k+L}]$ will coincide with current estimate of X_k and with initial values of new variables X_{new} [21, 37, 41]. Knowing the mean vector it is possible to calculate Jacobians of old and new factors within $G_{k+L}(a)$. Since information matrix $\Lambda = A^T A$ is a product of Jacobian matrices, Λ_{k+L} of future *belief* $b[X_{k+L}]$ can also be calculated without knowing the future observations. Thus, we can reason about information (and covariance) matrix of $G_{k+L}(a)$, as was shown in Chapter 2.

Now we can reformulate the information-theoretic objective of the BSP problem. In order to evaluate information impact of action a in a non-augmented BSP setting (Eq. (2.15)), we need to measure the amount of information added to a factor graph after augmenting it with new factors $G_{k+L}(a) = G_k \oplus \mathcal{F}(a)$. In case of augmented BSP (Eq. (2.16)), in order to evaluate information impact of action a we need to measure the amount of information added to a factor graph after connecting it to another factor graph $G(a)$ through factors in $\mathcal{F}^{conn}(a)$, $G_{k+L}(a) = G_k \oplus G(a) \oplus \mathcal{F}^{conn}(a)$.

In equations (2.9) and (2.10) we expressed the posterior information matrix Λ_{k+L} of $G_{k+L}(a)$ through matrix A, which is the weighted Jacobian of new terms from Eq. (2.6). In non-augmented BSP, each block-row of A represents a specific factor from $\mathcal{F}(a)$, while in augmented BSP block-rows in A represent factors from $\mathcal{F}(a) = \{\mathcal{F}^{new}(a), \mathcal{F}^{conn}(a)\}$. Block-columns of A represent all estimated variables within X_{k+L}. As was mentioned, each factor's block-row is sparse, with only non-zero block entries under columns of variables connected to the factor within the factor graph. For example, the Jacobian matrix's block-row that corresponds to a motion model factor $p(x_{k+l}|x_{k+l-1}, u_{k+l-1})$ will involve only two non-zero block entries for the state variables x_{k+l} and x_{k+l-1}. Factors for many measurement models, such as projection and range model, will also have only two non-zero blocks (see Figure 3.3).

We define two properties for any set of factors $\mathcal{F}$ that will be used in the sequel to analyze complexity of the proposed approach. Denote by $\mathcal{M}(\mathcal{F})$ the sum of dimensions of all factors in $\mathcal{F}$, where dimension of each factor is the dimension of its expected value r_i^j from Eq. (2.3). Additionally, let $\mathcal{D}(\mathcal{F})$ denote the total dimension of all variables involved in at least one factor

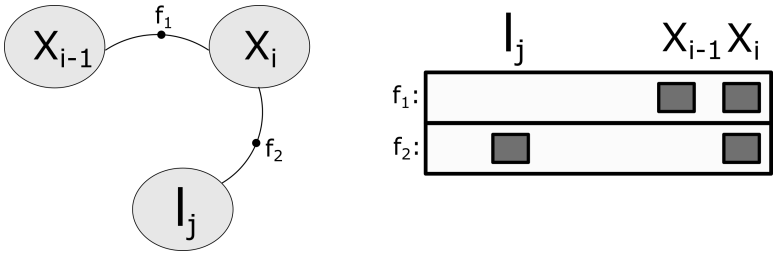

Figure 3.3: Concept illustration of A's structure. Each column represents some variable from state vector. Each row represents some factor from Eq. (2.6). Here, A represents set of factors $\mathcal{F} = \{f_1(x_{i-1}, x_i), f_2(x_i, l_j)\}$ where factor f_1 of motion model that involves two poses x_i and x_{i-1} will have non-zero values only at columns of x_i and x_{i-1}. Factor f_2 of observation model that involves together variables x_i and l_j will have non-zero values only at columns of x_i and l_j.

Notation	Description
$X_k = X_{k+L}$	state vector at times k and $k + L$
${}^{I}X$	subset of X_{k+L} with variables *involved* in new terms in Eq. (2.6)
${}^{\neg I}X$	subset of X_{k+L} with variables *not involved* in new terms in Eq. (2.6)
$X_k^F = X_{k+L}^F = X^F$	subset of X_{k+L} with `focused` variables
$X_k^U = X_{k+L}^U = X^U$	subset of X_{k+L} with `unfocused` variables
${}^{I}X^U$	subset of X^U with variables *involved* in new terms in Eq. (2.6)
${}^{\neg I}X^U$	subset of X^U with variables *not involved* in new terms in Eq. (2.6)

Table 3.1: Different partitions of state variables in BSP

from $\mathcal{F}$. It is not difficult to show that Jacobian matrix $A \in \mathbb{R}^{m \times n}$ of $\mathcal{F}$ has height $m = \mathcal{M}(\mathcal{F})$, and number of its columns that are not entirely equal to zero is $\mathcal{D}(\mathcal{F})$. The letter $\mathcal{D}$ is used here because *density* of information matrix is affected directly by value of $\mathcal{D}(\mathcal{F})$. It is important to note that, for any candidate action a, the total dimension of new factors $\mathcal{M}(\mathcal{F}(a))$ and dimension of involved variables $\mathcal{D}(\mathcal{F}(a))$ are *independent* of n, which is dimension the belief at planning time $b[X_k]$. Instead, both properties are only functions of the planning horizon L.

In the following sections we describe our BSP approach, using the above notions of factor graphs.

3.2 BSP via Matrix Determinant Lemma

3.2.1 Unfocused Case

Information theoretic BSP involves evaluating the costs from Eq. (2.15), operations that require calculating the determinant of a large $n \times n$ matrix (posterior information matrix), with n being the dimensionality of the state X_{k+L}. State of the art approaches typically perform these calculations from scratch for each candidate action.

In contrast, our approach contains a one-time calculation that depends on state dimension and will be re-used afterwards to calculate impact of each candidate action (see Section 3.4). As will be seen below, the latter depends only on $\mathcal{M}(\mathcal{F}(a))$ and $\mathcal{D}(\mathcal{F}(a))$, while being independent

of state dimension.

Recalling notations from the previous section, we would like to measure the amount of information gained after graph augmentation $G_{k+L}(a) = G_k \oplus \mathcal{F}(a)$. We can measure it through the IG as the utility function. It is not difficult to show that IG from Eq. (2.15) can be written as $J_{IG}(a) = \frac{1}{2} \ln \frac{\left|\Lambda_k + A^T A\right|}{\left|\Lambda_k\right|}$, where $A \in \mathbb{R}^{m \times n}$ is Jacobian of factors in $\mathcal{F}(a)$ weighted by their noise, with $m = \mathcal{M}(\mathcal{F}(a))$. Using the generalized matrix determinant lemma [13], this equation can be written as

$$J_{IG}(a) = \frac{1}{2} \ln \left| I_m + A \cdot \Sigma_k \cdot A^T \right|, \quad \Sigma_k \equiv \Lambda_k^{-1} \tag{3.6}$$

as previously suggested in [16, 32] in the context of compact pose-SLAM and `focused` active inference.

Eq. (3.6) provides an exact and general solution for information-based decision making, where each action candidate can produce any number of new factors (non-myopic planning) and where factors themselves can be of any motion or measurement model (unary, pairwise, etc.).

In many problem domains, such as SLAM, inference is typically performed in the information space and as such, the joint covariance matrix Σ_k is not readily available and needs to be calculated upon demand, which is expensive in general. While in first sight, it might seem the entire joint covariance matrix needs to be recovered, in practice this is not the case due to *sparsity* of the Jacobian matrix A, as was mentioned above.

Consequently, only specific entries from the covariance matrix Σ_k are really required, and sparse matrix techniques exist to calculate them efficiently [11, 23]. More formally, denote by ${}^I X$ the set of all variables that are connected to factors in $\mathcal{F}(a)$ (see Table 3.1), i.e. these are the variables that are involved in at least one factor among the new factors generated due to the currently considered candidate action a, see Eq. (2.6). Clearly, the columns of A that correspond to the rest of the variables, ${}^{\neg I} X$, are entirely filled with zeros (see Figure 3.3). Thus, Eq. (3.6) can be re-written as

$$J_{IG}(a) = \frac{1}{2} \ln \left| I_m + {}^I A \cdot \Sigma_k^{M,{}^I X} \cdot ({}^I A)^T \right| \tag{3.7}$$

where ${}^I A$ is constructed from A by removing all zero columns, and $\Sigma_k^{M,{}^I X}$ is a prior joint marginal covariance of variables in ${}^I X$, which should be calculated from the (square root) information matrix Λ_k. Note that dimension of ${}^I X$ is $\mathcal{D}(\mathcal{F}(a))$.

Intuitively, the posterior uncertainty reduction that corresponds to action a is a function of only the prior marginal covariance over variables involved in $\mathcal{F}(a)$ (i.e. $\Sigma_k^{M,{}^I X}$) and the new information introduced by the $\mathcal{F}(a)$'s Jacobian A, with the latter also involving the same variables ${}^I X$. Moreover, from the above equation it can be seen that uncertainty reduction in the posterior will be significant for large entries in A and high prior uncertainty over the variables ${}^I X$.

In particular, in case of myopic decision making with unary observation models (that involve only a single state variable), calculation of $IG(a)$ for different candidate actions only requires recovering the diagonal entries of Σ_k, regardless of the actual correlations between the states, as was recently shown in [19, 20]. However, while in the mentioned papers the per-action

calculation takes $O(n)$, the $IG(a)$ calculation is not dependent on n at all, as will be shown in Section 3.4.

Given a prior marginal covariance $\Sigma_k^{M,{}^{I}X}$, whose dimension is $\mathcal{D}(\mathcal{F}(a)) \times \mathcal{D}(\mathcal{F}(a))$, the calculation in Eq. (3.7) is bounded by calculating determinant of an $\mathcal{M}(\mathcal{F}(a)) \times \mathcal{M}(\mathcal{F}(a))$ matrix which is in general $O(\mathcal{M}(\mathcal{F}(a))^3)$, where $\mathcal{M}(\mathcal{F}(a))$ is the number of constraints due to new factors (for a given candidate action a). This calculation should be performed for each candidate action in the set $\mathcal{A}$. Furthermore, in many problems it is logical to assume that $\mathcal{M}(\mathcal{F}(a)) << n$, as $\mathcal{M}(\mathcal{F}(a))$ depends mostly on the planning horizon L, which is typically defined and constant, while n (state dimensionality) can be huge and grow with time in real systems (e.g. SLAM). Consequently, given the prior covariance our complexity for selecting best action is $O(|\mathcal{A}|)$, i.e. *independent* of state dimensionality n.

To conclude this section, we showed that calculation of action impact for a single candidate action does not depend on n. While this result is interesting by itself in the context of active inference, in Section 3.4 we go a step further and present an approach to calculate covariance entries, required by all candidates, with one-time calculation which can be re-used afterwards.

3.2.2 Focused Case

In this section we present a novel approach to calculate change in entropy of a `focused` set of variables after factor graph augmentation $G_{k+L}(a) = G_k \oplus \mathcal{F}(a)$, combining it with the ideas from the previous sections (generalized matrix determinant lemma and IG cost function) and showing that impact of one candidate action can be calculated independently of state dimension n.

First we recall definitions from Chapter 2 and introduce additional notations (see also Table 3.1): $X_k^F \equiv X_{k+L}^F \in \mathbb{R}^{n_F}$ denotes the set of `focused` variables (equal to X_{k+L}^F to remind us that prior and posterior states are identical in non-augmented case), $X_k^U \doteq X_k / X_k^F \in \mathbb{R}^{n_U}$ is a set of the remaining, `unfocused` variables, with $n = n_F + n_U$. The $n_F \times n_F$ prior marginal covariance and information matrices of X_k^F are denoted, respectively, by $\Sigma_k^{M,F}$ (suffix M for marginal) and $\Lambda_k^{M,F} \equiv (\Sigma_k^{M,F})^{-1}$. Furthermore, we partition the joint information matrix Λ_k as

$$\Sigma_k = \begin{bmatrix} \Sigma_k^{M,U} & \Sigma_k^{M,UF} \\ (\Sigma_k^{M,UF})^T & \Sigma_k^{M,F} \end{bmatrix}, \; \Lambda_k = \begin{bmatrix} \Lambda_k^{U} & \Lambda_k^{U,F} \\ (\Lambda_k^{U,F})^T & \Lambda_k^{F} \end{bmatrix}, \tag{3.8}$$

where $\Lambda_k^F \in \mathbb{R}^{n_F \times n_F}$ is constructed by retrieving from Λ_k only the rows and the columns related to X_k^F (it is actually conditional information matrix of X_k^F, conditioned on rest of variables X_k^U), $\Lambda_k^U \in \mathbb{R}^{n_U \times n_U}$ is defined similarly for X_k^U, and $\Lambda_k^{U,F} \in \mathbb{R}^{n_U \times n_F}$ contains remaining blocks of Λ_k as shown in Eq. (3.8).

The marginal information matrix of X_k^F, i.e. $\Lambda_k^{M,F}$, can be calculated via Schur complement $\Lambda_k^{M,F} = \Lambda_k^F - (\Lambda_k^{UF})^T \cdot (\Lambda_k^U)^{-1} \cdot \Lambda_k^{UF}$. However, one of Schur complement's properties [33] is

$\left|\Lambda_k\right| = \left|\Lambda_k^{M,F}\right| \cdot \left|\Lambda_k^U\right|$, from which we can conclude that

$$\left|\Lambda_k^{M,F}\right| = \frac{1}{\left|\Sigma_k^{M,F}\right|} = \frac{\left|\Lambda_k\right|}{\left|\Lambda_k^U\right|}. \tag{3.9}$$

Therefore, the posterior entropy of X_{k+L}^F (see Eq. (2.17)) is a function of the posterior Λ_{k+L} and its partition Λ_{k+L}^U:

$$J_{\mathcal{H}}^F(a) = \mathcal{H}(X_{k+L}^F) = \frac{n_F \cdot \gamma}{2} - \frac{1}{2} \ln \frac{\left|\Lambda_{k+L}\right|}{\left|\Lambda_{k+L}^U\right|}. \tag{3.10}$$

From Eq. (2.9) one can observe that $\Lambda_{k+L}^U = \Lambda_k^U + (A^U)^T \cdot A^U$, where $A^U \in \mathbb{R}^{m \times n_U}$ is constructed from Jacobian A by taking only the columns that are related to variables in X_k^U.

The next step is to use IG instead of entropy, with the same motivation and benefits as in the `unfocused` case (Section 3.2.1). The optimal action $a^* = \arg\max_{a \in \mathcal{A}} J_{IG}^F(a)$ will maximize $J_{IG}^F(a) = \mathcal{H}(X_k^F) - \mathcal{H}(X_{k+L}^F)$, and by combining Eq. (3.10) with the generalized matrix determinant lemma we can write:

$$J_{IG}^F(a) = \frac{1}{2} \ln \left| I_m + A \cdot \Sigma_k \cdot A^T \right| - \frac{1}{2} \ln \left| I_m + A^U \cdot \Sigma_k^{U|F} \cdot (A^U)^T \right|, \tag{3.11}$$

where $\Sigma_k^{U|F} \in \mathbb{R}^{n_U \times n_U}$ is a prior covariance matrix of X_k^U conditioned on X_k^F, and it is actually the inverse of Λ_k^U.

Further, A^U can be partitioned into ${}^I A^U$ and ${}^{\neg I} A^U$, representing `unfocused` variables that are, respectively, *involved* (${}^I X^U$) or *not involved* (${}^{\neg I} X^U$) (see also Table 3.1). Note that ${}^{\neg I} A^U$ contains only zeros, and it can be concluded that:

$$\left| I_m + A^U \cdot \Sigma_k^{U|F} \cdot (A^U)^T \right| = \left| I_m + {}^I A^U \cdot \Sigma_k^{{}^I X^U|F} \cdot ({}^I A^U)^T \right|, \tag{3.12}$$

where $\Sigma_k^{{}^I X^U|F}$ is the prior covariance of ${}^I X^U$ conditioned on X_k^F.

Taking into account equations (3.7) and (3.12), $J_{IG}^F(a)$ can be calculated through

$$J_{IG}^F(a) = \frac{1}{2} \ln \left| I_m + {}^I A \cdot \Sigma_k^{M,{}^I X} \cdot ({}^I A)^T \right| - \frac{1}{2} \ln \left| I_m + {}^I A^U \cdot \Sigma_k^{{}^I X^U|F} \cdot ({}^I A^U)^T \right|. \tag{3.13}$$

We can see that the `focused` and `unfocused` information gains have a simple relation between them

$$J_{IG}^F(a) = J_{IG}(a) - \frac{1}{2} \ln \left| I_m + {}^I A^U \cdot \Sigma_k^{{}^I X^U|F} \cdot ({}^I A^U)^T \right|. \tag{3.14}$$

The second term in Eq. (3.14) is negative and plays a role of penalty, reducing the action's impact on posterior entropy of X_{k+L}^F. In Section 3.5 we will discuss the intuition behind this penalty term. Note that when all *involved* variables are `focused` , ${}^I X \subseteq X_{k+L}^F$, the variable set ${}^I X^U$ is empty and second term's matrix will be an identity matrix I_m. In such a case, the second term becomes zero and we have $J_{IG}^F(a) = J_{IG}(a)$.

Also here, given prior covariances $\Sigma_k^{M,{}^I X}$ and $\Sigma_k^{{}^I X^U|F}$, calculation of `focused` IG (Eq. (3.13))

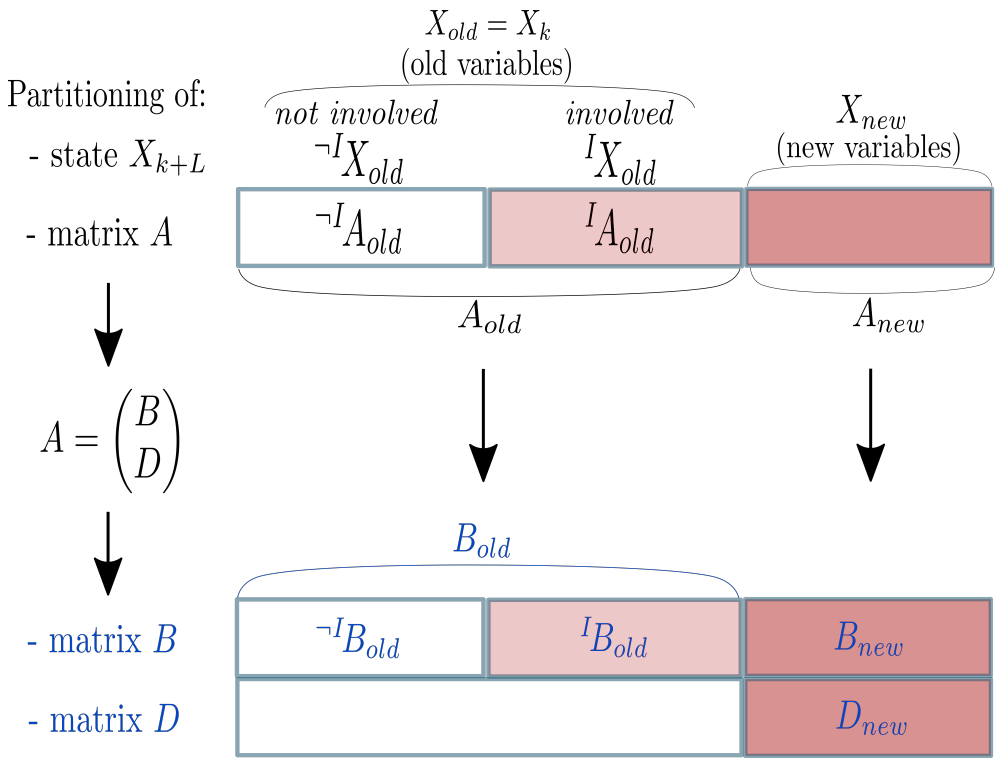

Figure 3.4: Partitions of Jacobians and state vector X_{k+L} in Augmented BSP case, `unfocused` scenario. Note: the shown variable ordering is only for illustration, while the developed approach supports *any* arbitrary variable ordering. Also note that all white blocks consist of only zeros. Top: Jacobian A of factor set $\mathcal{F}(a) = \{\mathcal{F}^{conn}(a), \mathcal{F}^{new}(a)\}$. Bottom: Jacobians B and D of factor sets $\mathcal{F}^{conn}(a)$ and $\mathcal{F}^{new}(a)$ respectively.

is *independent* of state dimensionality n, with complexity bounded by $O(\mathcal{M}(\mathcal{F}(a))^3)$. In the Section 3.4 we will show how the required covariances can be efficiently retrieved.

3.3 Augmented BSP via AMDL

3.3.1 Augmented Matrix Determinant Lemma (AMDL)

In order to simplify calculation of IG within Augmented BSP (Eq. (2.16)) one could resort, similar to previous sections, to the matrix determinant lemma. However, due to zero-padding, the information matrix Λ_{k+L}^{Aug} is singular and thus the matrix determinant lemma cannot be directly applied. In this section we develop a variant of the matrix determinant lemma for the considered augmented case (further referred to as AMDL).

Specifically, we want to solve the following problem: Recalling $\Lambda^+ = \Lambda^{Aug} + A^T \cdot A$ (see also Eq. (2.10)), and dropping the time indices to avoid clutter, our objective is to express the determinant of Λ^+ in terms of Λ and $\Sigma = \Lambda^{-1}$.

Lemma 3.3.1. *The ratio of determinants of Λ^+ and Λ can be calculated through:*

$$\frac{\left|\Lambda^+\right|}{\left|\Lambda\right|} = \left|\Delta\right| \cdot \left|A_{new}^T \cdot \Delta^{-1} \cdot A_{new}\right|, \tag{3.15}$$

with $\Delta \doteq I_m + A_{old} \cdot \Sigma \cdot A_{old}^T$*, where the matrices* $A_{old} \in \mathbb{R}^{m \times n}$ *and* $A_{new} \in \mathbb{R}^{m \times n'}$ *are constructed from A by retrieving columns of only old n variables (denoted as* X_{old}*) and only new n′ variables (denoted as* X_{new}*), respectively (see Figure 3.4 and Table 3.2).*

The proof of Lemma 3.3.1 is given in Appendix (9.1).

Remark 1: It is not difficult to show that AMDL for the matrix update of the form $\Lambda^+ = \Lambda^{Aug} + \widetilde{A}^T \cdot \Psi^{-1} \cdot \widetilde{A}$ (see Eq. (2.10)) assumes the form

$$\frac{\left|\Lambda^+\right|}{\left|\Lambda\right|} = \left|\Psi^{-1}\right| \cdot \left|\widetilde{\Delta}\right| \cdot \left|\widetilde{A}_{new}^T \cdot \widetilde{\Delta}^{-1} \cdot \widetilde{A}_{new}\right| \tag{3.16}$$

with $\widetilde{\Delta} \doteq \Psi + \widetilde{A}_{old} \cdot \Sigma \cdot \widetilde{A}_{old}^T$.

Additionally, we can extend the AMDL lemma for specific structure of matrix A. As was explained in Section 3.1, in case of augmented BSP the new factors can be separated into two sets $\mathcal{F}^{new}(a)$ and $\mathcal{F}^{conn}(a)$. It is not difficult to see that A's structure in such case will be:

$$A = \begin{pmatrix} A_{old} & A_{new} \end{pmatrix} = \begin{pmatrix} B_{old} & B_{new} \\ D_{old} & D_{new} \end{pmatrix} = \begin{pmatrix} B_{old} & B_{new} \\ 0 & D_{new} \end{pmatrix} \tag{3.17}$$

where B's rows represent factors from $\mathcal{F}^{conn}(a)$, and D's rows - factors from $\mathcal{F}^{new}(a)$ (see also Figure 3.4). Note that $D_{old} \equiv 0$.

Lemma 3.3.2. *The ratio of determinants of* Λ^+ *and* Λ *where A has structure from Eq. (3.17) can be calculated through:*

$$\frac{\left|\Lambda^+\right|}{\left|\Lambda\right|} = \left|\Delta_1\right| \cdot \left|B_{new}^T \cdot \Delta_1^{-1} \cdot B_{new} + D_{new}^T \cdot D_{new}\right| \tag{3.18}$$

with $\Delta_1 \doteq I_{m_{conn}} + B_{old} \cdot \Sigma \cdot B_{old}^T$ *and* $m_{conn} = \mathcal{M}(\mathcal{F}^{conn}(a))$*, where partitions of B and D are defined above in Eq. (3.17) and also can be seen in Figure 3.4.*

The proof of Lemma 3.3.2 is given in Appendix (9.2).

We note the above equations are general standalone solutions for any augmented positive definite symmetric matrix.

To summarize, we developed two augmented determinant lemmas, Eq. (3.15) and Eq. (3.18), with the latter exploiting additional knowledge about A's structure. Dimension of matrix Δ from Eq. (3.15) is $\mathcal{M}(\mathcal{F}(a)) \times \mathcal{M}(\mathcal{F}(a))$, whereas dimension of Δ_1 from Eq. (3.18) is $\mathcal{M}(\mathcal{F}^{conn}(a)) \times \mathcal{M}(\mathcal{F}^{conn}(a))$. Thus, complexity of calculation in Eq. (3.18) is lower than in Eq. (3.15) since $\mathcal{M}(\mathcal{F}^{conn}(a)) \leq \mathcal{M}(\mathcal{F}(a))$. In sections below we will use both of the lemmas in order to develop efficient solution to Augmented BSP problem.

Notation	Description
X_k	state vector at time k
X_{k+L}	state vector at time $k+L$
X^F_{k+L}	subset of X_{k+L} with `focused` variables
X_{old}	subset of X_{k+L} with old variables, i.e. X_k
X_{new}	subset of X_{k+L} with new variables
${}^{I}X_{old}$	subset of X_{old} with variables *involved* in new terms in Eq. (2.6)
${}^{\neg I}X_{old}$	subset of X_{old} with variables *not involved* in new terms in Eq. (2.6)
	`Focused` Augmented BSP ($X^F_{k+L} \subseteq X_{new}$), Section 3.3.3
X^F_{new}	subset of X_{new} with `focused` variables
X^U_{new}	subset of X_{new} with `unfocused` variables
	`Focused` Augmented BSP ($X^F_{k+L} \subseteq X_{old}$), Section 3.3.3
${}^{I}X^F_{old}$	subset of ${}^{I}X_{old}$ with `focused` variables
${}^{I}X^U_{old}$	subset of ${}^{I}X_{old}$ with `unfocused` variables
${}^{\neg I}X^F_{old}$	subset of ${}^{\neg I}X_{old}$ with `focused` variables
${}^{\neg I}X^U_{old}$	subset of ${}^{\neg I}X_{old}$ with `unfocused` variables

Table 3.2: Different partitions of state variables in Augmented BSP

3.3.2 Unfocused Augmented BSP through IG

Here we show how the augmented matrix determinant lemma from Section 3.3.1 can be used to efficiently calculate the `unfocused` IG as defined in Eq. (2.16), e.g. change in system's entropy after factor graph augmentation $G_{k+L}(a) = G_k \oplus G(a) \oplus \mathcal{F}^{conn}(a)$ (see Figure 3.2).

First we introduce different partitions of the joint state X_{k+L}, and the corresponding submatrices in the Jacobian matrix A from Eq. (2.10) (see Table 3.2 and Figure 3.4). Recall definitions of X_{new} and X_{old} (see Section 3.3.1) and let ${}^{I}X_{old}$ and ${}^{\neg I}X_{old}$ denote, respectively, the old involved and the old uninvolved state variables in the new terms in Eq. (2.6). We represent by ${}^{I}A_{old}$ and ${}^{\neg I}A_{old}$ the columns of matrix A that correspond to the state variables ${}^{I}X_{old}$ and ${}^{\neg I}X_{old}$, respectively (see Figure 3.4). Note, ${}^{\neg I}A_{old} \equiv 0$.

Next, using AMDL Lemma 3.3.1, the determinant ratio between posterior and prior information matrices is:

$$\frac{\left|\Lambda_{k+L}\right|}{\left|\Lambda_k\right|} = \left|C\right| \cdot \left|A^T_{new} \cdot C^{-1} \cdot A_{new}\right|, \tag{3.19}$$

where $C \doteq I_m + A_{old} \cdot \Sigma_k \cdot A^T_{old}$.

Consequently, the IG objective from Eq. (2.16) can be re-written as

$$J_{IG}(a) = \frac{n' \cdot \gamma}{2} + \frac{1}{2}\ln\left|C\right| + \frac{1}{2}\ln\left|A^T_{new} \cdot C^{-1} \cdot A_{new}\right|. \tag{3.20}$$

Moreover, considering the above partitioning of A_{old}, we conclude $A_{old} \cdot \Sigma_k \cdot A_{old}^T = {}^I A_{old} \cdot \Sigma_k^{M,{}^I X_{old}} \cdot ({}^I A_{old})^T$ where $\Sigma_k^{M,{}^I X_{old}}$ is the marginal prior covariance of ${}^I X_{old}$. Thus, matrix C can be rewritten as

$$C = I_m + {}^I A_{old} \cdot \Sigma_k^{M,{}^I X_{old}} \cdot ({}^I A_{old})^T. \tag{3.21}$$

Observe that, given $\Sigma_k^{M,{}^I X_{old}}$, all terms in Eq. (3.21) have relatively small dimensions and $\mathcal{M}(\mathcal{F}(a)) \times \mathcal{M}(\mathcal{F}(a))$ matrix C can be computed efficiently for each candidate action, with time complexity not depending anymore on state dimension n, similarly to the non-augmented BSP approach in Section 3.2.1. Calculation of the inverse C^{-1}, which is required in Eq. (3.20), is $O(\mathcal{M}(\mathcal{F}(a))^3)$ and will also not depend on n. The run-time of overall calculation in Eq. (3.20) will have complexity $O(\mathcal{M}(\mathcal{F}(a))^3 + n'^3)$ and will depend only on number of new factors $\mathcal{M}(\mathcal{F}(a))$ and number of new variables n'. Both are functions of the planning horizon L and can be considered as being considerably smaller than state dimension n. Moreover, higher ratios $n/\mathcal{M}(\mathcal{F}(a))$ lead to a bigger advantage of our approach vs the alternatives (see Chapter 6).

It is worthwhile to mention a specific case, where $\mathcal{M}(\mathcal{F}(a)) = n'$, which happens for example in SLAM application when candidate action a introduces only motion (or odometry) factors between the new variables. In such case it is not difficult to show that Eq. (3.20) will be reduced to $J_{IG}(a) = \frac{n' \cdot \gamma}{2} + \ln \left| A_{new} \right|$. In other words, the information gain in such case depends only on the partition A_{new} of A (see Figure 3.4), Jacobian entries related to new variables, , while the prior Λ_k is not involved in the calculations at all.

Remark 2: It is possible that posterior state dimension $N = n + n'$ will be different for different candidate actions (e.g. see Chapter 6). In such case, the entropy (or IG), being function of posterior eigenvalues' product, will be of different scale for each candidate and can not be directly compared. Thus, dimension normalization of Eq. (3.20) may be required. Even though the term $\frac{n' \cdot \gamma}{2}$ may already play a role of such a normalization, the detailed investigation of this aspect is outside the scope of this thesis.

We can further enhance the presented above approach by considering structure of A from Eq. (3.17) (see also Figure 3.4). This will allow us to slightly improve complexity of $J_{IG}(a)$'s calculation. By applying AMDL Lemma 3.3.2, we can show that information gained from connecting G_k (with covariance matrix Σ_k) and $G(a)$ (with information matrix $\Lambda_a = D_{new}^T \cdot D_{new}$) through factors $\mathcal{F}^{conn}(a)$ will be:

$$J_{IG}(a) = \frac{n' \cdot \gamma}{2} + \frac{1}{2} \ln \left| C_1 \right| + \frac{1}{2} \ln \left| B_{new}^T \cdot C_1^{-1} \cdot B_{new} + \Lambda_a \right|. \tag{3.22}$$

$$C_1 = I_{m_{conn}} + B_{old} \cdot \Sigma_k \cdot B_{old}^T \tag{3.23}$$

where matrix $B = \begin{pmatrix} B_{old} & B_{new} \end{pmatrix}$ is Jacobian of factors in $\mathcal{F}^{conn}(a)$.

Since B_{old} is sparse (see Figure 3.4), same as partition A_{old} in Eq. (3.19), C_1 also can be calculated in efficient way:

$$C_1 = I_{m_{conn}} + {}^I B_{old} \cdot \Sigma_k^{M,{}^I X_{old}} \cdot ({}^I B_{old})^T \tag{3.24}$$

It is interesting to note that the terms of the above-presented solution for the *unfocused* Augmented BSP problem (Eq. (3.22) and Eq. (3.24)) can be recognized as belonging to different operands in augmentation $G_k \oplus G(a) \oplus \mathcal{F}^{conn}(a)$: Prior covariance matrix $\Sigma_k^{M,{}^I X_{old}}$ represents information coming from prior factor graph G_k, information matrix Λ_a provides information of action's factor graph $G(a)$, and various partitions of matrix B introduce information coming from *connecting* factors $\mathcal{F}^{conn}(a)$.

Although the above solution (Eq. (3.22) and Eq. (3.24)) look somewhat more complicated, its matrix terms have slightly lower dimensions comparing with matrix terms in the general solution presented in Eq. (3.20) and Eq. (3.21), with complexity $O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n'^3)$, and therefore can be calculated faster, as will be shown in our simulations below. Moreover, the equations (3.22) and (3.24) have more independent terms which can be calculated in parallel, further improving time performance.

3.3.3 Focused Augmented BSP

The *focused* scenario in Augmented BSP setting, with the factor graph augmentation $G_{k+L}(a) = G_k \oplus G(a) \oplus \mathcal{F}^{conn}(a)$, can be separated to different cases. One such case is when the set of focused variables X_{k+L}^F contains only new variables added during BSP augmentation, as illustrated in Figure 3.5, i.e. $X_{k+L}^F \subseteq X_{new}$ are the variables coming from factor graph $G(a)$. Such a case happens, for example, when we are interested in reducing entropy of robot's last pose within the planning horizon. Another case is when the focused variables X_{k+L}^F contain only old variables, as shown in Figure 3.6, i.e. $X_{k+L}^F \subseteq X_{old} \equiv X_k$ are the variables coming from factor graph G_k. This, for example, could correspond to a scenario where reducing entropy of already-mapped landmarks is of interest (e.g. improve 3D reconstruction quality). The third option is for both new and old variables to be inside X_{k+L}^F. Below we develop a solution for the first two cases; the third case can be handled in a similar manner.

Remark 3: In most cases, actual variable ordering will be more sporadic than the one depicted in Figures 3.4, 3.5 and 3.6. For example, iSAM [24] determines variable ordering using COLAMD [8] to enhance sparsity of the square root information matrix. We note that our approach applies to any arbitrary variable ordering, with the equations derived herein remaining unchanged.

Focused Augmented BSP ($X_{k+L}^F \subseteq X_{new}$) - focused variables belong to $G(a)$

First we define additional partitions of Jacobian A (see Figure 3.5). The sub-matrices A_{old}, A_{new}, ${}^I A_{old}$ and ${}^{\neg I} A_{old}$ were already introduced in the sections above. We now further partition A_{new} into A_{new}^F and A_{new}^U, that correspond, respectively, to columns of new variables that are focused and unfocused. Denote the former set of variables as X_{new}^F and the latter as X_{new}^U (see also Table 3.2). Note, $X_{new}^F \equiv X_{k+L}^F$.

Lemma 3.3.3. *The posterior entropy of X_{new}^F (Eq. (2.17)) is given by*

$$J_{\mathcal{H}}^F(a) = \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln \left| (A_{new}^U)^T \cdot C^{-1} \cdot A_{new}^U \right| - \frac{1}{2} \ln \left| A_{new}^T \cdot C^{-1} \cdot A_{new} \right| \tag{3.25}$$

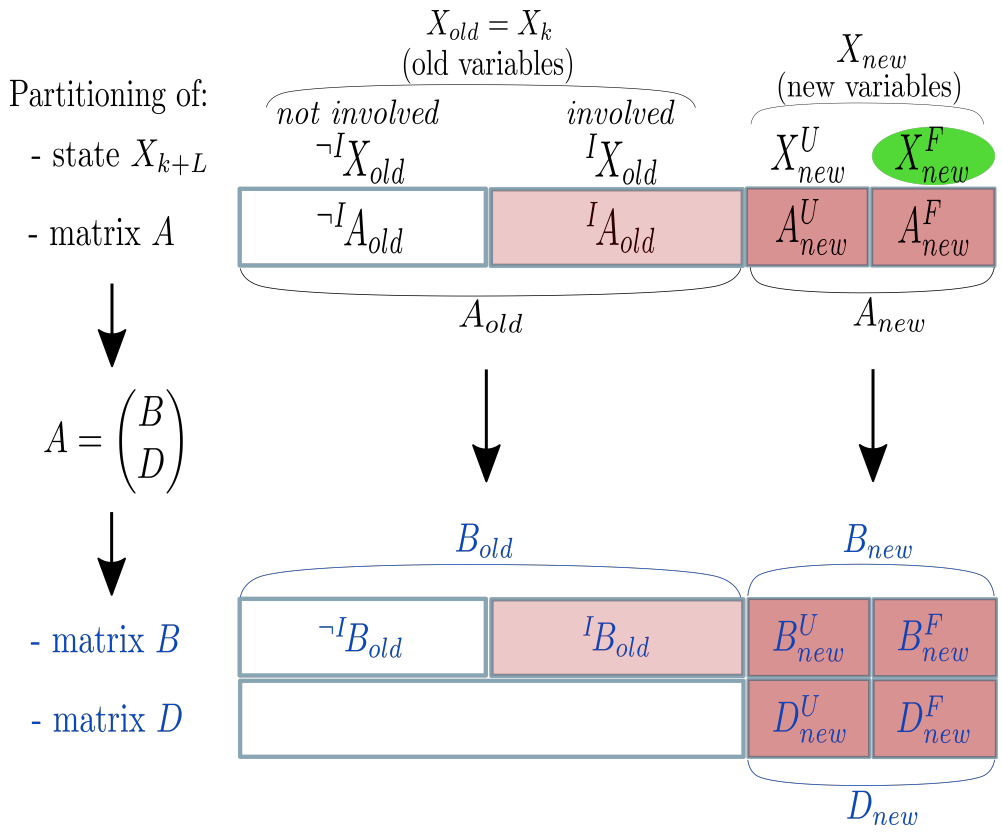

Figure 3.5: Partitions of Jacobians and state vector X_{k+L} in Augmented BSP case, `Focused` ($X^F_{k+L} \subseteq X_{new}$) scenario. Note: the shown variable ordering is only for illustration, while the developed approach supports *any* arbitrary variable ordering. Also note that all white blocks consist of only zeros. Top: Jacobian A of factor set $\mathcal{F}(a) = \{\mathcal{F}^{conn}(a), \mathcal{F}^{new}(a)\}$. Bottom: Jacobians B and D of factor sets $\mathcal{F}^{conn}(a)$ and $\mathcal{F}^{new}(a)$ respectively.

where C is defined in Eq. (3.21).

The proof of Lemma 3.3.3 is given in Appendix (9.3).

We got an exact solution for $J^F_{\mathcal{H}}(a)$ that, given $\Sigma^{M,{}^I X_{old}}_k$, can be calculated efficiently with complexity $O(\mathcal{M}(\mathcal{F}(a))^3 + n'^3)$, similarly to `unfocused` Augmented BSP in Section 3.3.2. In Section 3.4 we will explain how the prior marginal covariance term ($\Sigma^{M,{}^I X_{old}}_k$) can be efficiently retrieved, providing a fast solution for `focused` Augmented BSP.

Additionally, it is interesting to note that there is efficient way to calculate the term $\frac{1}{2}\ln\left|(A^U_{new})^T \cdot C^{-1} \cdot A^U_{new}\right| - \frac{1}{2}\ln\left|A^T_{new} \cdot C^{-1} \cdot A_{new}\right|$ from Eq. (3.25). First, we calculate matrix $V \doteq A^T_{new} \cdot C^{-1} \cdot A_{new}$. Note that each row/column of V represents one of the new variables X_{new}. Next, we reorder rows and columns of V to obtain matrix V^{UF} where first go rows and columns of X^U_{new}, followed by rows and columns of X^F_{new}. Now, we can perform Cholesky decomposition of $V^{UF} = L^T \cdot L$ and retrieve L's diagonal entries that belong to variables X^F_{new}, denoted by $r^F_{i,i}$. It is not difficult to show that:

$$\frac{1}{2}\ln\left|(A^U_{new})^T \cdot C^{-1} \cdot A^U_{new}\right| - \frac{1}{2}\ln\left|A^T_{new} \cdot C^{-1} \cdot A_{new}\right| = -\sum \log r^F_{ii} \tag{3.26}$$

Further, like in Section 3.3.2, we will additionally exploit the special structure of A from Eq. (3.17) (see also Figure 3.5). Similarly to `unfocused` Augmented BSP, this will allow us to

improve complexity of $J^F_{\mathcal{H}}(a)$'s calculation.

Lemma 3.3.4. *The posterior entropy of X^F_{new} (Eq. (2.17)), where A has structure from Eq. (3.17), is given by*

$$J^F_{\mathcal{H}}(a) = \frac{n_F \cdot \gamma}{2} + \frac{1}{2}\ln\left|(B^U_{new})^T \cdot C_1^{-1} \cdot B^U_{new} + \Lambda_a^{U|F}\right| - \frac{1}{2}\ln\left|B^T_{new} \cdot C_1^{-1} \cdot B_{new} + \Lambda_a\right|, \quad (3.27)$$

where C_1 is defined in Eq. (3.24), $\Lambda_a = D^T_{new} \cdot D_{new}$ is information matrix of action's factor graph $G(a)$, and $\Lambda_a^{U|F} = (D^U_{new})^T \cdot D^U_{new}$ is information matrix of variables X^U_{new} conditioned on X^F_{new} and calculated from distribution represented by $G(a)$.

The proof of Lemma 3.3.4 is given in Appendix (9.4).

Also here, the matrix terms from the above solution of `focused` Augmented BSP problem (Eq. (3.27) have lower dimensions comparing with the matrix terms from the general solution presented in Eq. (3.25). Given the prior marginal covariance $\Sigma_k^{M,{}^I X_{old}}$ its complexity is $O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n'^3)$. We demonstrate a run-time superiority of this solution in our simulations below.

It is important to mention that information-based planning problem for a system that is propagated through the (Extended) Kalman filter [41, 43], where the objective is to reduce uncertainty of only marginal future state of the system, is an instance of the `focused` Augmented BSP ($X^F_{k+L} \subseteq X_{new}$) problem. Thus, the solution provided in this section is applicable also for Kalman filter planning.

`Focused Augmented BSP` ($X^F_{k+L} \subseteq X_{old}$) - focused variables belong to G_k

Similarly to the previous section, we first introduce additional partitions of Jacobian A for the considered case (see Figure 3.6). From the top part of the figure we can see that ${}^{\neg I}A_{old}$ can further be partitioned into ${}^{\neg I}A^U_{old}$ and ${}^{\neg I}A^F_{old}$. In particular, ${}^{\neg I}A^U_{old}$ represents columns of old variables that are both not involved and unfocused, and ${}^{\neg I}A^F_{old}$ represents columns of old variables that are both not involved and focused. We denote the former group of variables by ${}^{\neg I}X^U_{old}$ and the latter by ${}^{\neg I}X^F_{old}$ (see Table 3.2). Likewise, ${}^I A_{old}$ can be partitioned into ${}^I A^U_{old}$ and ${}^I A^F_{old}$, representing old involved variables that are, respectively, unfocused (${}^I X^U_{old}$) or focused (${}^I X^F_{old}$). Note that in this case, the set of focused variables is $X^F_{k+L} = X^F_k = \{{}^{\neg I}X^F_{old} \cup {}^I X^F_{old}\}$ and is contained in factor graph G_k.

Lemma 3.3.5. *The* `focused` *IG of X^F_k is given by*

$$J^F_{IG}(a) = \frac{1}{2}(\ln\left|C\right| + \ln\left|A^T_{new} \cdot C^{-1} \cdot A_{new}\right| - \ln\left|S\right| - \ln\left|A^T_{new} \cdot S^{-1} \cdot A_{new}\right|), \quad (3.28)$$

where C is defined in Eq. (3.21), and

$$S \doteq I_m + {}^I A^U_{old} \cdot \Sigma_k^{{}^I X^U_{old}|F} \cdot ({}^I A^U_{old})^T, \quad (3.29)$$

and where $\Sigma_k^{{}^I X^U_{old}|F}$ is the prior covariance of ${}^I X^U_{old}$ conditioned on X^F_k.

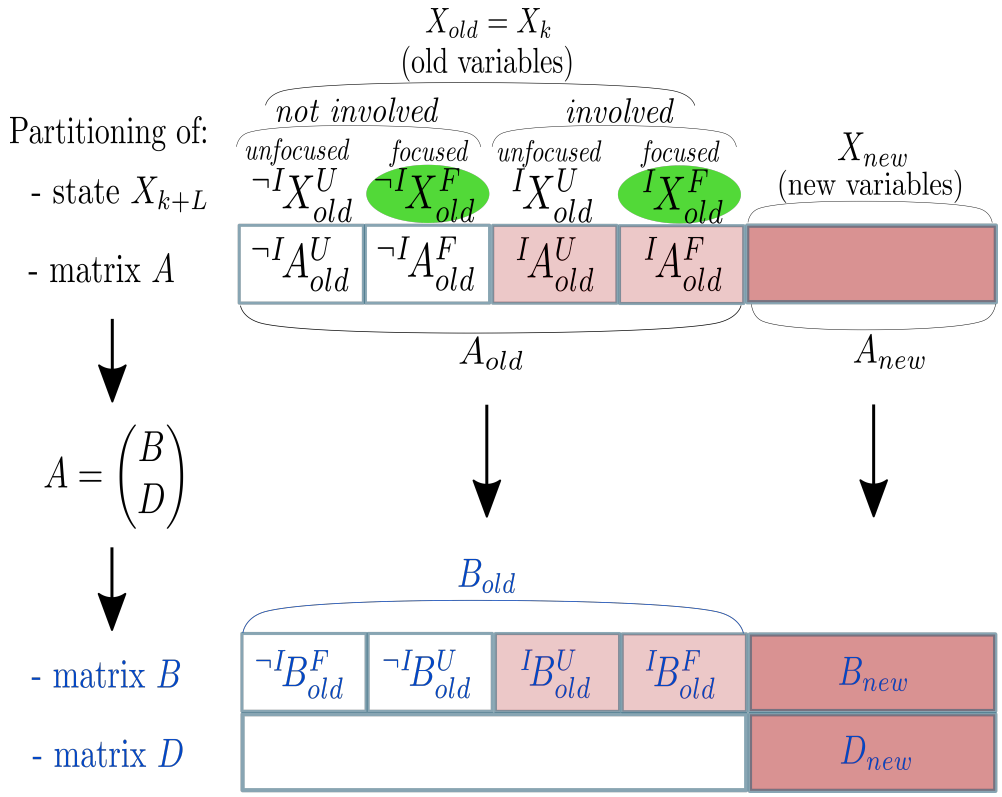

Figure 3.6: Partitions of Jacobians and state vector X_{k+L} in Augmented BSP case, `Focused` ($X^F_{k+L} \subseteq X_{old}$) scenario. Note: the shown variable ordering is only for illustration, while the developed approach supports *any* arbitrary variable ordering. Also note that all white blocks consist of only zeros. Top: Jacobian A of factor set $\mathcal{F}(a) = \{\mathcal{F}^{conn}(a), \mathcal{F}^{new}(a)\}$. Bottom: Jacobians B and D of factor sets $\mathcal{F}^{conn}(a)$ and $\mathcal{F}^{new}(a)$ respectively.

The proof of Lemma 3.3.5 is given in Appendix (9.5).

Similarly to the cases discussed above (Sections 3.3.2 and 3.3.3), given $\Sigma_k^{M,{}^I X_{old}}$ and $\Sigma_k^{{}^I X^U_{old}|F}$, calculation of $J^F_{IG}(a)$ per each action a can be performed efficiently with complexity $O(\mathcal{M}(\mathcal{F}(a))^3 + n'^3)$, independently of state dimension n.

It is interesting to note the specific case where $\mathcal{M}(\mathcal{F}(a)) = n'$. In other words, number of new measurements is equal to number of new state variables. In such case, it is not difficult to show that Eq. (3.28) will always return zero. We can conclude that for this specific case ($\mathcal{M}(\mathcal{F}(a)) = n'$) there is no new information about the old focused variables X^F_k.

Additionally, similar to previous sections, we will use the special structure of A from Eq. (3.17) (see also Figure 3.6) in order to improve complexity of $J^F_{IG}(a)$'s calculation.

Lemma 3.3.6. *The* `focused` *IG of* X^F_k*, where A has structure from Eq. (3.17), is given by*

$$J^F_{IG}(a) = \frac{1}{2}(\ln\left|C_1\right| + \ln\left|B^T_{new} \cdot C_1^{-1} \cdot B_{new} + \Lambda_a\right| - \ln\left|S_1\right| - \ln\left|B^T_{new} \cdot S_1^{-1} \cdot B_{new} + \Lambda_a\right|), \quad (3.30)$$

where C_1 *is defined in Eq. (3.24),* $\Lambda_a = D^T_{new} \cdot D_{new}$ *is information matrix of action's factor graph* $G(a)$*, and*

$$S_1 = I_{m_{conn}} + {}^I B^U_{old} \cdot \Sigma_k^{{}^I X^U_{old}|F} \cdot ({}^I B^U_{old})^T, \quad (3.31)$$

and where $\Sigma_k^{{}^I X^U_{old}|F}$ *is the prior covariance of* ${}^I X^U_{old}$ *conditioned on* X^F_k*.*

Problem	Required covariance entries
`Unfocused` BSP, Section 3.2.1	$\Sigma_k^{M,{}^I X}$ (prior marginal covariance of variables *involved* in new terms in Eq. (2.6))
`Focused` BSP, Section 3.2.2	$\Sigma_k^{M,{}^I X}$ and $\Sigma_k^{{}^I X^U \mid F}$ (prior covariance of `unfocused` and *involved* variables ${}^I X^U$ conditioned on `focused` variables X^F)
`Unfocused` Augmented BSP, Section 3.3.2	$\Sigma_k^{M,{}^I X_{old}}$ (prior marginal covariance of old variables *involved* in new terms in Eq. (2.6))
`Focused` Augmented BSP ($X_{k+L}^F \subseteq X_{new}$), Section 3.3.3	$\Sigma_k^{M,{}^I X_{old}}$
`Focused` Augmented BSP ($X_{k+L}^F \subseteq X_{old}$), Section 3.3.3	$\Sigma_k^{M,{}^I X_{old}}$ and $\Sigma_k^{{}^I X_{old}^U \mid F}$ (prior covariance of `unfocused` and *involved* old variables ${}^I X_{old}^U$ conditioned on `focused` variables X^F)

Table 3.3: Different problems and required entries of prior covariance. BSP - short for non-augmented belief space planning, Augmented BSP - augmented belief space planning.

The proof of Lemma 3.3.6 is given in Appendix (9.6).

The matrix terms from the above solution (Eq. (3.30) and Eq. (3.31)) have lower dimensions comparing with the matrix terms from the general solution presented in Eq. (3.28) and Eq. (3.29), with complexity $O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n'^3)$ given the prior marginal covariance matrices $\Sigma_k^{M,{}^I X_{old}}$ and $\Sigma_k^{{}^I X_{old}^U \mid F}$. The next section presents our approach to calculate the appropriate entries in the prior covariance only once and re-use the result whenever required.

3.4 Re-use Calculations Technique

As we have seen above, `unfocused` and `focused` (augmented) BSP problems require different prior covariance entries, in order to use the developed expressions. The required entries for each problem are summarized in Table 3.3. Note that $\Sigma_k^{M,{}^I X}$ and $\Sigma_k^{M,{}^I X_{old}}$ both represent exactly the same thing, prior marginal covariance of old variables *involved* in new terms in Eq. (2.6)), and have slightly different notations due to specifics of augmented and non-augmented settings of BSP. The same goes for $\Sigma_k^{{}^I X^U \mid F}$ and $\Sigma_k^{{}^I X_{old}^U \mid F}$, with both representing prior covariance of `unfocused` and *involved* old variables ${}^I X_{old}^U$ conditioned on `focused` variables X^F. In this section we will use notations of Augmented BSP ($\Sigma_k^{M,{}^I X_{old}}$ and $\Sigma_k^{{}^I X_{old}^U \mid F}$), considering the non-augmented BSP setting its special case.

From Table 3.3 it is seen that all approaches require prior marginal covariance of the involved old variables, i.e. ${}^I X_{old}$. In terms of factor graphs, in non-augmented BSP the ${}^I X_{old}$ represents variables connected to factors from set $\mathcal{F}(a)$ and has dimension $\mathcal{D}(\mathcal{F}(a))$, whereas in augmented BSP scenario the ${}^I X_{old}$ represents variables from prior factor graph G_k connected to factors in set $\mathcal{F}^{conn}(a)$ and has dimension $\mathcal{D}(\mathcal{F}^{conn}(a))$. Although each candidate action may induce a different set of involved variables, in practice these sets will often have many variables in

common as they are all related to the belief at the current time (e.g. about robot pose), in one way or another. With this in mind, we perform a one-time calculation of prior marginal covariance for *all* involved variables (due to at least one candidate action) and re-use it for efficiently calculating IG and entropy of different candidate actions.

More specifically, denote by $X_{All} \subseteq X_k$ the subset of variables that were involved in new terms in Eq. (2.6) for at least one candidate action. We can now perform a one-time calculation of the prior marginal covariance for this set, i.e. $\Sigma_k^{M,X_{All}}$. The complexity of such calculation may be different for different applications. For example, when using an information filter, the system is represented by information matrix Λ_k, and in general the inverse of Schur compliment of X_{All} variables should be calculated. However, there are techniques that exploit sparsity of the underlying matrices in SLAM problems, in order to efficiently recover marginal covariances [23], and more recently, to keep and update them incrementally [17]. In Chapter 6 we show that calculation time of $\Sigma_k^{M,X_{All}}$ while exploiting sparsity [11, 23] is relatively small comparing to total decision making time of alternative approaches. Still, the more detailed discussion about complexity of covariance retrieval can be found in publications [17, 23].

For `focused` BSP (Section 3.2.2) and for `focused` Augmented BSP ($X_{k+L}^F \subseteq X_{old}$) (Section 3.3.3) cases, we also need the term $\Sigma_k^{{}^I X_{old}^U | F}$ (see Eq. (3.13) and (3.29)). This term can be computed in two different ways as described below.

First way: we will calculate it through additional marginal covariance entries. First we will calculate the prior marginal covariance $\Sigma_k^{M,({}^I X_{old}^U, F)}$ for the set of variables $\{{}^I X_{old}^U, X_k^F\}$, and then compute the Schur complement over the relevant partitions in $\Sigma_k^{M,({}^I X_{old}^U, F)}$ (suffix M denotes marginal):

$$\Sigma_k^{{}^I X_{old}^U | F} = \Sigma_k^{M,{}^I X_{old}^U} - \Sigma_k^{M,{}^I X_{old}^U F} \cdot (\Sigma_k^{M,F})^{-1} \cdot \Sigma_k^{M,F {}^I X_{old}^U} \tag{3.32}$$

Consequently, we can use a one-time calculation also for the `focused` BSP and for `focused` Augmented BSP ($X_{k+L}^F \subseteq X_{old}$) cases as follows. Let us extend the set X_{All} to contain also all focused variables. Once $\Sigma_k^{M,X_{All}}$ is calculated, $\Sigma_k^{M,({}^I X_{old}^U, F)}$ will be just its partition and can be easily retrieved from it. As a result, the calculation of $\Sigma_k^{{}^I X_{old}^U | F}$ per candidate action becomes computationally cheap (through Eq. (3.32)). Furthermore, term $(\Sigma_k^{M,F})^{-1}$ can be calculated only once for all candidates.

Second way: we will compute the $\Sigma_k^{{}^I X_{old}^U | F}$ through information matrix partitioning. Note that matrix $\Lambda_k^{X_{old}^U}$, an partition of information matrix Λ_k that belong to old `unfocused` variables X_{old}^U, is information matrix of distribution of X_{old}^U variables conditioned on `focused` variables X_k^F. Thus, the $\Sigma_k^{{}^I X_{old}^U | F}$ is just a partition of $(\Lambda_k^{X_{old}^U})^{-1}$ that belong to old `unfocused` *involved* variables ${}^I X_{old}^U$. Therefore, we need to calculate specific entries of $\Lambda_k^{X_{old}^U}$'s inverse. In this case our one-time calculation will be as following. We denote by $X_{All}^U \subseteq X_k$ the subset of `unfocused` variables that were involved in new terms in Eq. (2.6) for at least one candidate action. Next, we will calculate $\Sigma_k^{X_{All}^U | F}$ - the entries of $\Lambda_k^{X_{old}^U}$'s inverse that belong to X_{All}^U. Now, the $\Sigma_k^{{}^I X_{old}^U | F}$ is just partition of $\Sigma_k^{X_{All}^U | F}$ and can be retrieved easily for each candidate action.

The first method is good option when dimension of X_k^F is relatively small. In such cases

the Eq. (3.32) can be calculated very fast. When this is not the case and number of focused variables is large, the second technique will become much faster alternative.

Remark 4: As we will see in Section 4.2, there are cases where ${}^{I}X^{U}_{old}$ is identical between all candidate actions. In such cases $\Sigma_k^{{}^{I}X^{U}_{old}|F}$ can be calculated only once and further reused by each candidate action.

To summarize this section, the presented technique performs time-consuming calculations in one computational effort; the results are then used for efficiently evaluating the impact of each candidate action. This concept thus preserves expensive CPU resources of any given autonomous system.

3.5 Connection to Mutual Information Approach and Theoretical Meaning of IG

Mutual information $I(a|b)$ is one additional metric from information theory that is used a lot in the field of information-based decision making. Basically it encodes the quantity of information about set of variables a that we would get in case the value of variables in other set b would be revealed to us. For example, this metric was used in [9, 23] to determine the most informative measurements in a measurement selection problem, and more recently in [2] for information-based active exploration, with both problems being very similar. Additionally, it was used in [4] to create a sparse approximation of the true marginalization using Chow-Liu tree.

In this section we will explore the connection between our BSP approach that uses IG (see Section 3.2) and mutual information approach that is applied in [9, 23]; we will show that objective functions of both are mathematically identical and calculate exactly the same metric, even though calculations in our approach are made in a much more efficient way. Moreover, we also will present theoretical meaning of IG that provides better intuition for equations (3.6) and (3.11).

In MI approach we would like to select the most informative measurements from the available set $\{z_1, z_2, \ldots\}$ and also to account for possible measurement correlation. Each candidate measurement has specific measurement model $z_i = h_i(X^i_k) + \upsilon_i$ with $\upsilon_i \sim \mathcal{N}(0, \Psi_i)$. The candidate measurements are a priori unknown and can be viewed as random variables whose statistic properties are fully defined by a random state vector X_k and random noises υ_i, due to measurement models. Combining candidate measurements with the state vector we have

$$W = (X_k, z_1, z_2, \ldots)^T, \tag{3.33}$$

and similarly to the mentioned papers, it can be shown that the covariance matrix of W is

$$\Sigma_W = \begin{pmatrix} \Sigma_k & \Sigma_k \cdot \widetilde{A_1}^T & \Sigma_k \cdot \widetilde{A_2}^T & \cdots \\ \widetilde{A_1} \cdot \Sigma_k & \widetilde{A_1} \cdot \Sigma_k \cdot \widetilde{A_1}^T + \Psi_1 & \widetilde{A_1} \cdot \Sigma_k \cdot \widetilde{A_2}^T & \cdots \\ \widetilde{A_2} \cdot \Sigma_k & \widetilde{A_2} \cdot \Sigma_k \cdot \widetilde{A_1}^T & \widetilde{A_2} \cdot \Sigma_k \cdot \widetilde{A_2}^T + \Psi_2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \tag{3.34}$$

where $\widetilde{A_i}$ is Jacobian of measurement model function $h_i(X_k^i)$ and where it wasn't combined yet with model noise Ψ_i, similarly to $\widetilde{A}$ defined in Eq. (2.9). The MI approach [9, 23] calculates $I(X_k|z_i)$ for each candidate z_i from Σ_W and selects candidates with highest mutual information.

Now we will show that objective $I(X_k|z_i)$ is mathematically identical to our $J_{IG}(z_i)$ from Section 3.2.1 (see also Eq. (3.6)). First, note that $\Sigma_W^{X_k|z_i} = (\Lambda_k + \widetilde{A_i}^T \cdot \Psi_i^{-1} \cdot \widetilde{A_i})^{-1}$ (easy to check by using Schur complement from left and Woodbury matrix identity from right). Further, MI for Gaussian distributions can be calculated through covariance matrices as

$$\begin{aligned} I(X_k|z_i) = \mathcal{H}(X_k) - \mathcal{H}(X_k|z_i) = \frac{1}{2} \ln \frac{\left|\Sigma_W^{M,X_k}\right|}{\left|\Sigma_W^{X_k|z_i}\right|} = \\ \frac{1}{2} \ln \frac{\left|\Sigma_W^{M,X_k}\right|}{\left|\Sigma_W^{M,X_k} - \Sigma_W^{M,X_k z_i} \cdot (\Sigma_W^{M,z_i})^{-1} \cdot \Sigma_W^{M,z_i X_k}\right|} = \\ \frac{1}{2} \ln \frac{\left|\Sigma_k\right|}{\left|\Sigma_k - \Sigma_k \cdot \widetilde{A_i}^T \cdot (\widetilde{A_i} \cdot \Sigma_k \cdot \widetilde{A_i}^T + \Psi_i)^{-1} \cdot \widetilde{A_i} \cdot \Sigma_k\right|} \end{aligned} \tag{3.35}$$

and further can be reduced to $I(X_k|z_i) = \frac{1}{2} \ln \frac{\left|\Lambda_k + \widetilde{A_i}^T \cdot \Psi_i^{-1} \cdot \widetilde{A_i}\right|}{\left|\Lambda_k\right|}$ which is exactly the `unfocused` IG from Eq. (3.6) for case when candidate action $a_i \equiv z_i$ introduces single factor into the factor graph.

While both approaches are obviously calculating the same metric, the computation complexity is not the same. In both [9] and [23], the objective is calculated through Eq. (3.35) and its complexity depends on X_k's dimension. In contrast, our approach *rAMDL* does so independently of state dimension through Eq. (3.7) as has been shown in sections above, making it more efficient comparing to the MI technique.

Additionally, in [23] Kaess et al. presented the approach to sequentially select informative measurements that accounts for measurements correlation and redundancy, but without the need to update state estimation during each decision. In Section 4.1 we present our algorithm *Sequential rAMDL* where we combine similar idea together with *rAMDL* technique in order to eliminate the need in marginal covariance calculation at each decision.

Most importantly, from the above equations we can see conceptually a very interesting meaning of the metric that is calculated (IG or MI). Without omitting noise matrix Ψ from our

formulation, we can show that `unfocused` IG of future measurement z is

$$J_{IG}(z) = \frac{1}{2} \ln \left| I_m + A \cdot \Sigma_k \cdot A^T \right| = \frac{1}{2} \ln \frac{\left| \Psi + \widetilde{A} \cdot \Sigma_k \cdot \widetilde{A}^T \right|}{\left| \Psi \right|}. \tag{3.36}$$

Further, from Eq. (3.34) we see that $\Sigma^z \doteq \Psi + \widetilde{A} \cdot \Sigma_k \cdot \widetilde{A}^T$ is covariance matrix of the random z. Thus, we can see that

$$J_{IG}(z) = \frac{1}{2} \ln \left| \Sigma^z \right| - \frac{1}{2} \ln \left| \Psi \right| = \mathcal{H}(z) - \mathcal{H}(\upsilon), \tag{3.37}$$

where υ is random noise from z's measurement model, with $\upsilon \sim \mathcal{N}(0, \Psi)$. From Eq. (3.37) we see that information gain is exactly the difference between entropies of future measurement and its noise. It can be explained in the following way - as was mentioned before, random variable z is fully defined by random variables X_k and υ through measurement model. When z's value is revealed it obviously provides information about both state and noise. The information about the state (the information gain) then will be the whole received information (the entropy of r.v. z) minus the information about the noise υ.

From the above we can see that in order for measurement z to be notably informative three conditions should apply. First, its noise should have small entropy $\mathcal{H}(\upsilon)$ which also comes from general knowledge about measurement estimation. Additionally, z should have big entropy $\mathcal{H}(z)$ from which we can conclude second and third conditions - the *involved* variables ${}^{I}X$ from the measurement model should have high prior uncertainty (high prior entropy), as also their $\widetilde{{}^{I}A}$ (Jacobian of measurement model at linearization point of ${}^{I}X$) should contain high absolute values (the sign does not matter because of quadratic term of $\widetilde{A}$ in Eq. (3.36)).

In same way we can review equation for `focused` IG (Eq. (3.11)). The first term $\frac{1}{2} \ln \left| I_m + A \cdot \Sigma_k \cdot A^T \right|$ measures amount of information about whole state X_k, while the second term

$$\frac{1}{2} \ln \left| I_m + A^U \cdot \Sigma_k^{U|F} \cdot (A^U)^T \right| = \frac{1}{2} \ln \frac{\left| \Psi + \widetilde{A^U} \cdot \Sigma_k^{U|F} \cdot (\widetilde{A^U})^T \right|}{\left| \Psi \right|} = \mathcal{H}(z|X_k^F) - \mathcal{H}(\upsilon) \tag{3.38}$$

measures the information given that X_k^F was provided, meaning information for only `unfocused` variables. The difference between total information and information of only `unfocused` variables will provide the information about the `focused` set X_k^F.

Such interpretation of IG's meaning through entropy of future measurement and of its noise can be considered not only for measurement selection problem but also for the more general formulation from Chapter 2, thus constituting a possible direction for future research.

3.6 Mutual Information - Fast Calculation via Information Matrix

While deriving our approach *rAMDL* , through similar equations we come by to the way to calculate mutual information from entries of only information matrix Λ_k, without necessity to retrieve covariance matrix Σ_k. We note that such calculation will be more efficient and beneficial in cases when only information matrix available (f.e. iSAM2 [24]). Thus, we present below the developed mathematical notations as additional contribution of this thesis.

Lemma 3.6.1. *When current state vector contains only variable sets a and b, $X_k = \{a, b\}$, then $I(a|b)$ can be calculated through*

$$I(a|b) = \frac{1}{2} \ln \frac{\left|\Lambda_k^a\right| \cdot \left|\Lambda_k^b\right|}{\left|\Lambda_k\right|} \tag{3.39}$$

where Λ_k^a and Λ_k^b are partitions of information matrix Λ_k with respect to variables from a and b respectively.

Lemma 3.6.2. *When current state vector contains additional variables r, $X_k = \{a, b, r\}$, then $I(a|b)$ can be calculated through*

$$I(a|b) = \frac{1}{2} \ln \frac{\left|\Lambda_k^{(a,r)}\right| \cdot \left|\Lambda_k^{(b,r)}\right|}{\left|\Lambda_k\right| \cdot \left|\Lambda_k^r\right|} \tag{3.40}$$

where $\Lambda_k^{(a,r)}$ is a partition of information matrix Λ_k with respect to variables from both a and r.

The proof of Lemmas 3.6.1 and 3.6.2 is given respectively in Appendix (9.9).

Practically we see that $I(a|b)$ is function of determinants of information matrix and its different partitions. Equations (3.39) and (3.40) allow quick calculation of mutual information between different subsets of X_k using information matrix Λ_k as representative of the system, and do not require calculation of covariance matrices first. And such formulation will be beneficial for cases when Information-Kalman filter or ISAM2 are used to estimate the system.

Chapter 4

Application to Different Problem Domains

In Chapter 3, we provided an efficient solution for a general BSP problem, considering both non-augmented and augmented cases. In this section we discuss various problem domains of (augmented) BSP and show how our approach can be applied for each case. More concretely, we focus on Sensor Deployment (Section 4.1), active SLAM (Section 4.2) and Graph Reduction (Section 4.3), as specific non-augmented and augmented BSP applications. For the former, we develop a more computationally efficient variant of our approach. For each case, we first briefly formulate the problem and then describe our solution.

4.1 Sensor Deployment

Sensor deployment is one of the most researched problems of decision making. The basic idea is to measure a specific metric in domain space such as, e.g., temperature within building's space. The goal is to find the best locations for available sensors in order to estimate the metric in entire domain in the most accurate way.

Typically discretization of domain space is made due to computation complexity considerations. Thus, we have n available locations in the space, $L \doteq \{l_1, \ldots, l_n\}$, where sensors can be deployed. The metric's values in these locations can be modeled as random variables and combined into state vector: $X = \{x_1, .., x_n\}$.

Putting sensor at location l_i will allow us to take measurement z_i at that location, which will provide information about the metric at place, x_i. Assume that measurement model of sensor is known and is:

$$z_i = h_i(x_i) + \upsilon_i, \quad \upsilon_i \sim \mathcal{N}(0, \Sigma_{\upsilon,i}). \tag{4.1}$$

Additionally, correlation between different locations may be known a priori. Such prior can be presented as X's joint distribution, $\mathbb{P}_0(X)$. Assuming that it is Gaussian, it may be represented as [20, 27, 44, 45]

$$X \sim \mathbb{P}_0(X) = \mathcal{N}(\mu, \Sigma_0) = \mathcal{N}^{-1}(\eta, \Lambda_0). \tag{4.2}$$

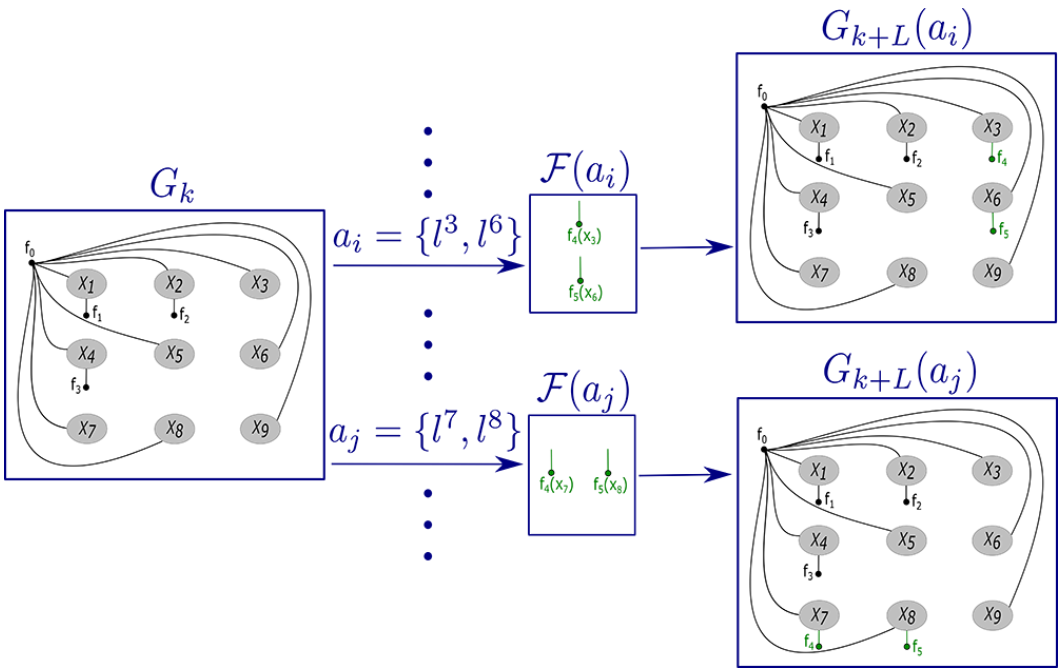

Figure 4.1: Illustration of belief propagation in factor graph representation - Sensor Deployment scenario. The space is discretized through grid of locations $L \doteq \{l_1, \ldots, l_9\}$. Factor f_0 within prior factor graph G_k represents our prior belief about state vector, $\mathbb{P}_0(X)$. Factors $f_1 - f_3$ represent measurements taken from sensors deployed at locations l_1, l_2 and l_4. Two actions $a_i = \{l^3, l^6\}$ and $a_j = \{l^7, l^8\}$ are considered, introducing into graph two new factor sets $\mathcal{F}(a_i)$ and $\mathcal{F}(a_j)$ respectively (colored in green). In this example value of c' is 2.

Note that in practice in typical sensor deployment problems Λ_0 is not actually available and Σ_0 is used instead. Nevertheless, in further formulation we assume that Λ_0 was calculated a priori (as Σ_0^{-1}) and therefore is available to us.

Finding best sensor locations in order estimate the metric in most accurate way is another instance of information-based not-augmented BSP and therefore can be viewed through a prism of factor graphs (see Figure 4.1) as we show below.

Conceptually, the space of candidate actions in Sensor Deployment setting contains all subsets of possible sensor locations $S \subseteq L$ with the usual constraint on cardinality of subset S, $|S| \leq c$, to represent that number of sensors is limited. But considering all subsets of size c is usually unrealistic as the number of all possible subsets $\binom{n}{c}$ is astronomical due to its combinatorial nature. Therefore, typically the problem is solved in greedy way.

We propose a sub-optimal approach where a sequence of decisions must be made instead of one decision. During each decision we are looking for subset S', $|S'| \doteq c'$, with c' locations chosen from locations that were not yet selected. The optimal S' is the one that maximizes X's estimation accuracy. The algorithm ends when overall set of locations $S = \{S'_1, S'_2, ..\}$ grows to cardinality of c. Note that number of locations in each subset, c', should be such that number of S' candidates, $\binom{n}{c'}$, is small enough to be evaluated in a realistic time period. Thus, c' is scenario-dependent and should be selected manually.

More specifically, we assume that till time t_k the disjoint subsets $\{S'_1, \ldots, S'_k\}$ of locations were selected, where each location subset $S'_j = \{l_j^1, \ldots, l_j^{c'}\}$ provided measurements $Z_j =$

$\{z_j^1, \ldots, z_j^{c'}\}$. Given these measurements, the joint pdf at time t_k is

$$\mathbb{P}(X|Z_{1:k}) \propto \mathbb{P}_0(X) \prod_{j=1}^{k} \prod_{i=1}^{c'} \mathbb{P}(z_j^i|x_j^i), \tag{4.3}$$

where observation model $\mathbb{P}(z_j^i|x_j^i)$ is defined in Eq. (4.1)

MAP estimation of X according to information in Eq. (4.3) will provide current state *belief* $b_k[X] \doteq \mathbb{P}(X|Z_{1:k}) = \mathcal{N}(X_k^*, \Sigma_k)$, and following Eq. (2.7) the information matrix of $b_k[X]$ is $\Lambda_k = \Sigma_k^{-1} = \Lambda_0 + \sum_{j=1}^{k} \sum_{i=1}^{c'} (H_j^i)^T \cdot (\Sigma_{v,j,i})^{-1} \cdot H_j^i$ where $H_j^i \doteq \nabla_x h_j^i$ are the Jacobian matrices of observation model (Eq. (4.1)) for all measurement terms in Eq. (4.3), linearized about the current estimate X_k^*. Note that the *belief* $b_k[X]$ can be naturally represented by a factor graph G_k as was explained in Section 3.1 (see also Figure 4.1).

The next decision requires to select next candidate action a - a location subset S'_{k+1} that will minimize posterior uncertainty. Therefore, candidate space contains all subsets of form $S' \subseteq L \setminus \{S'_1 \cup \ldots \cup S'_k\}$ and $|S'| = c'$. Each such candidate subset $a \equiv S' = \{l^1, \ldots, l^{c'}\}$ will provide future measurements $Z' = \{z^1, \ldots, z^{c'}\}$ and thus future *belief* $b_{k+1}[X]$ and its information matrix will be

$$b_{k+1}[X] = \mathbb{P}(X|Z_{1:k}, Z') \propto b_k[X] \prod_{i=1}^{c'} \mathbb{P}(z^i|x^i), \quad \Lambda_{k+1} = \Lambda_k + \sum_{i=1}^{c'} (H^i)^T \cdot (\Sigma_{v,i})^{-1} \cdot H^i. \tag{4.4}$$

Thus, the candidate S' introduces to G_k the factor set $\mathcal{F}(a)$, which contains exactly c' factors. Each one of the factors is connected to one variable - the x_i that represents location of factor's sensor (see Figure 4.1).

Similarly to the general formulation in Chapter 2, stacking all new Jacobians in the above equation together into a single matrix and combining all noise matrices into block-diagonal one, will lead to Eq. (2.9). Hence, the optimal candidate subset S' will be the one that maximizes IG from Eq. (2.15).

Note that the block-columns of Jacobian matrix $A \in \mathbb{R}^{m \times n}$ from Eq. (2.9) represent all possible sensor locations and block-rows represent new c' measurement factors from Eq. (4.4). As was mentioned before, only *involved* variables will have non-zero values in their block-columns. It is not difficult to show that in the Sensor Deployment problem, the *involved* variables are x_i that belong to locations in subset S'. Block-columns of all other variables in A will be zeros.

The rest part of the problem definition (objective functions, `unfocused` and `focused` settings) for the Sensor Deployment problem is exactly identical to the general formulation. In particular, in `unfocused` setting the optimal S'_{k+1} will be found through

$$S'_{k+1} = \underset{S' \subseteq X \setminus \{S'_1, \ldots, S'_k\}, |S'| = c'}{\arg\max} J_{IG}(S') = \frac{1}{2} \ln \left| I_m + A_{S'} \cdot \Sigma_k \cdot A_{S'}^T \right| \tag{4.5}$$

where $A_{S'}$ is Jacobian matrix of candidate S'.

Solution - *Sequential rAMDL*

The above problem can be straightforwardly solved by *rAMDL* approach, through Eq. (3.7) and (3.13). However, for *each* sequential decision the marginal covariance should be calculated for set of variables involved in any of the candidate actions, and it is not difficult to show that this set will contain all yet unoccupied locations. In scenarios with high number possible sensor locations, this can negatively affect overall time performance.

Here we present an enhanced approach, *Sequential rAMDL* , that performs the same sub-optimal sequence of decisions as described above, but uses *only* the prior covariance matrix Σ_0, *without* recalculating covariance entries after each decision. Such an approach gives an approximated solution (compared to the sub-optimal sequence of decisions described above) but without paying computation resources for expensive manipulation of high-dimensional matrices.

The first decision will be performed in exactly the same way - we will look for the best subset S'_1 of size c' that maximizes IG (Eq. (4.5)), for `unfocused` case. However, upon finding such a subset, the estimation solution of the system will not be updated due to measurements from new sensors. Instead, in each next decision we will look for a subset S'_{k+1} that maximizes the following objective

$$S'_{k+1} = \underset{S' \subseteq X \setminus \{S'_1,..,S'_k\}, |S'|=c'}{\arg\max} J_{IG}(\widetilde{S}) = \frac{1}{2} \ln \left| I_{\widetilde{m}} + A_{\widetilde{S}} \cdot \Sigma_0 \cdot A_{\widetilde{S}}^T \right|, \quad A_{\widetilde{S}} = \begin{pmatrix} A_{S'_1} \\ \vdots \\ A_{S'_k} \\ A_{S'} \end{pmatrix} \tag{4.6}$$

where $\widetilde{S} \doteq \{S'_1, .., S'_k, S'\}$, and $A_{\widetilde{S}}$ is a matrix with all appropriate Jacobians combined together.

Note that the sequential decision making through Eq. (4.6) will yield an exact solution, compared to sequential decision making through Eq. (4.5), if Jacobian matrices H^i (Eq. (4.4)) do not change after acquiring measurements from newly deployed sensors. This is the case, for instance, when linearization point X_k^* stayed the same or when measurement model (Eq. (4.1)) is linear with respect to x_i (i.e. $z_i = x_i + \nu_i$). Otherwise, Eq. (4.6) will merely be the approximation of the above approach.

After looking into Eq. (4.6) one can see that matrix inside is actually:

$$I_{\widetilde{m}} + A_{\widetilde{S}} \cdot \Sigma_0 \cdot A_{\widetilde{S}}^T = \begin{pmatrix} V_{S'_1} & Y_{S'_1,S'_2} & \cdots & Y_{S'_1,S'} \\ Y_{S'_2,S'_1} & V_{S'_2} & \cdots & Y_{S'_2,S'} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{S',S'_1} & Y_{S',S'_2} & \cdots & V_{S'} \end{pmatrix}, V_{S'} \doteq I_m + A_{S'} \cdot \Sigma_0 \cdot (A_{S'})^T, \quad Y_{S'_i,S'_j} \doteq A_{S'_i} \cdot \Sigma_0 \cdot A_{S'_j}^T \tag{4.7}$$

where $V_{S'}$ and $Y_{S'_i,S'_j}$ can be efficiently calculated (independently of state dimension) due to the sparsity of Jacobians. Moreover, after Σ_0 is calculated (or given) at the beginning of the algorithm, all its required entries are freely accessible through all the run-time of algorithm.

It can be seen that all diagonal matrices $V_{S'}$ were already calculated during the first decision and can be kept and re-used. Also all the correlation matrices $Y_{S'_i,S'_j}$ (except for $Y_{S'_k,S'}$) were

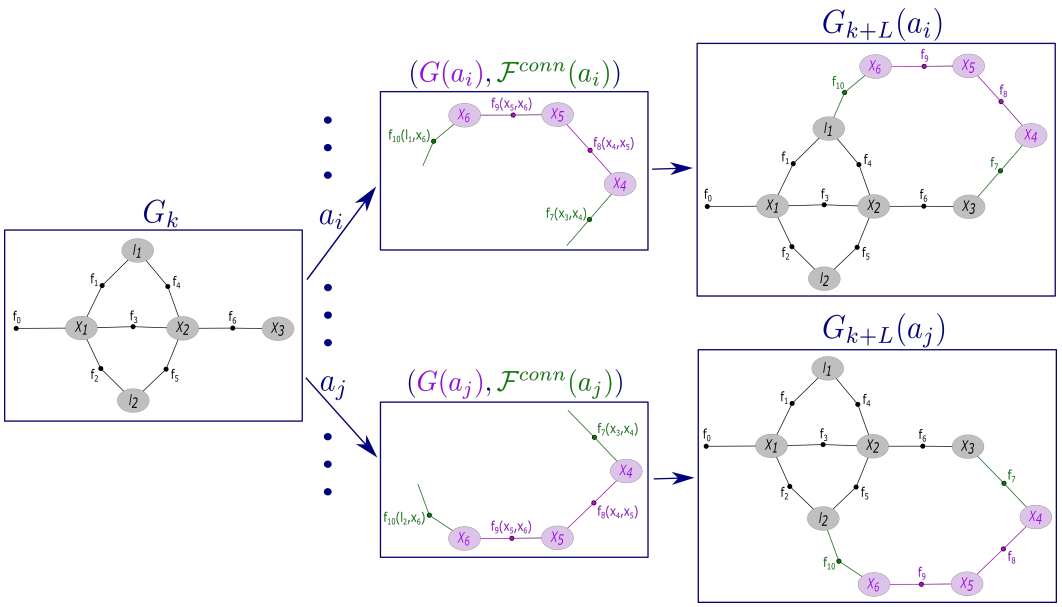

Figure 4.2: Illustration of belief propagation in factor graph representation - SLAM scenario. Nodes x_i represent robot poses, while nodes l_i - landmarks. Factor f_1 is prior on robot's initial position x_1; factors between robot poses represent motion model (Eq. (4.8)); factors between pose and landmark represent observation model (Eq. (4.9)). Two actions a_i and a_j are considered, performing loop-closure to re-observe landmark l_1 and l_2 respectively. Both actions introduce their own factor graphs $G(a_i)$ and $G(a_j)$ (colored in pink) that are *connected* to prior G_k through factor sets $\mathcal{F}^{conn}(a_i)$ and $\mathcal{F}^{conn}(a_j)$ (colored in green) respectively.

calculated in previous decisions. The only required calculation in every decision for each candidate S' is the matrix $Y_{S'_k,S'}$ and determinant of the combined matrix.

Our `unfocused` *Sequential rAMDL* approach can be seen as providing a little increase in the per-candidate calculation in order to escape the necessity of prior covariance calculation for each decision, similarly to method of sequential informative measurements selection presented in [23]. This approach can be a good alternative to *rAMDL* technique when one-time calculation part of *rAMDL* (Section 3.4) is more time-consuming than the part of candidates evaluation, as will be shown in our simulations. The `focused` *Sequential rAMDL* approach is also possible, by following similar derivations. Moreover, the same idea is applicable to other sequential domains like for example measurement selection problem.

4.2 Augmented BSP in Unknown Environments

In this section we discuss a specific case of the Augmented BSP problem from Chapter 2, considering a SLAM setting. Such a specification provides the reader with an illustrated example of the Augmented BSP problem for better intuition.

Let us refine the definition. In *smoothing* formulation of visual SLAM the state vector X_k represents robot poses per each time step, $\{x_0, \ldots, x_k\}$, and landmarks mapped till now, $L_k \doteq \{l_1, \ldots, l_{n_k}\}$. Further, we model robot motion dynamics and sensor observations through:

$$x_{i+1} = f(x_i, u_i) + \omega_i \quad , \quad \omega_i \sim \mathcal{N}(0, \Sigma_{\omega,i}) \tag{4.8}$$

$$z_{i,j} = h(x_i, l_j) + \upsilon_{i,j} \quad , \quad \upsilon_{i,j} \sim \mathcal{N}(0, \Sigma_{\upsilon,i,j}) \tag{4.9}$$

where u_i is control at time t_i, $z_{i,j}$ represents observation of landmark l_j by robot from position x_i at time t_i, and where ω_i and $\upsilon_{i,j}$ are the motion and measurement noises, respectively. Note that the motion model can be easily presented in the form of a general factor model $r_i^j = h_i^j(X_i^j) + \upsilon_i^j$ from Eq. (2.3) by moving the left side to the right:

$$0 = f(x_i, u_i) - x_{i+1} + \omega_i = \bar{f}(x_i, x_{i+1}) + \omega_i \quad , \quad \omega_i \sim \mathcal{N}(0, \Sigma_{\omega,i}). \tag{4.10}$$

The joint pdf for the SLAM problem at time t_k (or current *belief*) is then

$$b[X_k] = \mathbb{P}(X_k | Z_{0:k}, u_{0:k-1}) \propto \mathbb{P}(x_0) \prod_{i=1}^{k} \Big\{ \mathbb{P}(x_i | x_{i-1}, u_{i-1}) \prod_{j=1}^{n_i} \mathbb{P}(z_{i,j} | x_i, l_j) \Big\}, \tag{4.11}$$

where $\mathbb{P}(x_0)$ is a prior on robot's first pose, $Z_i = \{z_{i,1}, ..., z_{i,n_i}\}$ represents all observations at time t_i and n_i being the number of such observations. The motion and observation models $\mathbb{P}(x_i | x_{i-1}, u_{i-1})$ and $\mathbb{P}(z_{i,j} | x_i, l_j)$ are defined by Eq. (4.8) and Eq. (4.9). A factor graph representation, considering for simplicity only a two landmarks l_1 and l_2, is shown in Figure 4.2. Performing MAP inference over the belief $b[X_k]$, one can write $b[X_k] = \mathcal{N}(X_k^*, \Sigma_k)$, with appropriate mean vector X_k^* and covariance matrix Σ_k.

The space of candidate actions in SLAM setting contains all control sequences $u_{k+1:k+L-1}$, where L is the planning horizon and can vary between different candidates. Typically finite set of candidates is pooled from this infinite space according to their relevance to robot's current destination or to loop-closure maneuver, for example through simulation [39] and sampling [1, 38]. Similar to Eq. (2.6), future *belief* $b[X_{k+L}] \doteq \mathbb{P}(X_{k+L} | Z_{0:k+L}, u_{0:k+L-1})$ for particular candidate action $a = u_{k+1:k+L-1}$ can be explicitly written as

$$b[X_{k+L}] \propto b[X_k] \prod_{l=k+1}^{k+L} \Big\{ \mathbb{P}(x_l | x_{l-1}, u_{l-1}) \prod_{j=1}^{n_l} \mathbb{P}(z_{l,j} | x_l, l_j) \Big\}, \tag{4.12}$$

where X_{k+L} is the state vector at the L-th look ahead step. It contains all variables from the current state vector X_k and is augmented by new robot poses $X_{new} = \{x_{k+1}, \ldots, x_{k+L}\}$. Also note that in Eq. (4.12) we consider only new observations of landmarks that were already mapped till time t_k. It is also possible to reason about observing not yet mapped landmarks [18] but it is outside scope of this work.

Following the model from Section 3.1, the candidate's factor graph $G(a) = (\mathcal{F}^{new}(a), X_{new}, \mathcal{E}_{new})$ will contain all new robot poses connected by motion model factors $\mathcal{F}^{new}(a) = \{f_{k+1}^M, \ldots, f_{k+L-1}^M\}$ with appropriate motion models $\{\bar{f}(x_{k+1}, x_{k+2}), \ldots,$
$\bar{f}(x_{k+L-1}, x_{k+L})\}$, whereas factors from $\mathcal{F}^{conn}(a)$, which *connect* old variables X_k and new variables X_{new}, will contain one motion model factor f_k^M (with motion model $\bar{f}(x_k, x_{k+1})$) and all of observation model factors connecting new poses with observed landmarks (see Figure 4.2).

Following the general formulation, the posterior information matrix of belief $b[X_{k+L}]$,

i.e. Λ_{k+L}, can be constructed by first augmenting the current information matrix $\Lambda_k \equiv \Sigma_k^{-1}$ with L zero block-rows and block-columns, each block having dimension n_p of robot pose variable, to get $\Lambda_{k+L}^{Aug} \in \mathbb{R}^{N \times N}$ with $N = n + L \cdot n_p$, and thereafter adding to it new information, as illustrated in Figure 2.1 (see e.g. [21]):

$$\Lambda_{k+L} = \Lambda_{k+L}^{Aug} + \sum_{l=k+1}^{k+L} \left\{ F_l^T \cdot \Sigma_{\omega,l}^{-1} \cdot F_l + \sum_{j=1}^{n_l} H_j^T \cdot \Sigma_{v,l,j}^{-1} \cdot H_j \right\}, \tag{4.13}$$

where $F_l \doteq \nabla_x f$ and $H_j \doteq \nabla_x h$ are augmented Jacobian matrices of all new factors in Eq. (4.12) (motion and observation terms all together), linearized about the current estimate of X_k and about initial values of newly introduced robot poses.

Again, after stacking together all new Jacobians in the above equation and combining all noise matrices into a block-diagonal matrix, we will get the same posterior information expression as in Eq. (2.10).

Note that the block-columns of matrix $A \in \mathbb{R}^{m \times N}$ from Eq. (2.10) represent all old robot poses, mapped till now landmarks and new robot poses from L-horizon future. A's block-rows represent new motion and observation factors from Eq. (4.12). As mentioned before, only *involved* variables will have non-zero values in their block-columns. It is not difficult to see that in SLAM the *involved* ones are: all new robot poses, current robot pose x_k and all landmarks that will be observed following current candidate's actions. Block-columns of all other variables in A will be zeros.

The rest of the problem definition (objective functions, `unfocused` and `focused` settings) for the active SLAM problem is identical to the general formulation in Chapter 2.

Solution - *rAMDL* applied to SLAM

The Augmented BSP problem for SLAM case, described in previous section, can be naturally solved by our general approach from Section 3.3. However we will make one step further and provide solution tailored specifically to the SLAM domain, as example of applying *rAMDL* to a real problem and in order to show the underlying structure of the SLAM solution.

First, let us model informative partitions of Jacobian matrices B and D from Eq. (3.17), ${}^I B_{old}$, B_{new} and D_{new} (see also Figure 3.4), for one of the candidate actions, action a. As was mentioned above, the factors from action's factor graph $G(a)$, $\mathcal{F}^{new}(a)$, contain all new motion model factors from Eq. (4.12), except for factor f_k^M. Therefore, D_{new} will have the following

form:

$$
\begin{matrix} & (\underline{\text{columns of } x_{k+1}, \ldots, x_{k+L}}) \\ D_{new} = & \begin{pmatrix} \text{block-row for } \bar{f}(x_{k+1}, x_{k+2}) \\ \vdots \\ \text{block-row for } \bar{f}(x_{k+L-1}, x_{k+L}) \end{pmatrix} = \end{matrix}
$$

$$
\begin{matrix} & (\underline{x_{k+1}}) & (\underline{x_{k+2}}) & (\underline{x_{k+3}}) & (\underline{x_{k+4}}) & \cdots & (\underline{x_{k+L-2}}) & (\underline{x_{k+L-1}}) & (\underline{x_{k+L}}) \\ \Psi_{new}^{-\frac{1}{2}} \cdot & \begin{pmatrix} \mathbb{F}_{k+1} & -I & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \mathbb{F}_{k+2} & -I & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \mathbb{F}_{k+3} & -I & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbb{F}_{k+4} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -I & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & \mathbb{F}_{k+L-2} & -I & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \mathbb{F}_{k+L-1} & -I \end{pmatrix} \end{matrix} \doteq \Psi_{new}^{-\frac{1}{2}} \cdot \widetilde{D}_{new}
$$

(4.14)

where $\mathbb{F}_{k+l} \doteq \nabla_x f\Big|_{x=x_{k+l}}$ is Jacobian of motion model function f from Eq. (4.8) with respect to x_{k+l}, $-I$ is Jacobian of $\bar{f}$ from Eq. (4.10) with respect to second pose and is actually an identity matrix with dimension equal to dimension of robot pose. Matrix Ψ_{new} is block-diagonal, combining all noise matrices of $\mathcal{F}^{new}(a)$ factors. Additionally, we denote by $\widetilde{D}_{new}$ the Jacobian entries of D_{new} not weighted by factors' noise Ψ_{new}.

Assume that following a's controls the set of landmarks $L_a \subseteq L_k$ will be observed. Also, define set of all new observation factors $\mathcal{F}^{obs}(a)$ as

$$
\mathcal{F}^{obs}(a) = \{ \textit{factor } f_i^O \textit{ with observation model } h_i(x, l) : x \in X_{new}, l \in L_a, 1 \leq i \leq n_o \} \quad (4.15)
$$

where n_o is number of such factors. Thus, the *connecting* factors are $\mathcal{F}^{conn}(a) = \{f_k^M, \mathcal{F}^{obs}(a)\}$; and *involved* old variables will be ${}^I X_{old} = \{L_a, x_k\}$, containing x_k because of first factor's motion

model $\bar{f}(x_k, x_{k+1})$. Therefore, ${}^{I}B_{old}$ and B_{new} will be:

$$\left({}^{I}B_{old} \quad B_{new}\right) = \begin{pmatrix} \text{block-row for } \bar{f}(x_k, x_{k+1}) \\ \text{block-row for } h_1 \\ \vdots \\ \text{block-row for } h_{n_o} \end{pmatrix} = \begin{matrix} (\underline{\text{columns of } L_a}, x_k, x_{k+1}, \ldots, x_{k+L}) \end{matrix}$$

$$\begin{matrix} (\underline{L_a}) & (\underline{x_k}) & (\underline{x_{k+1}}) & \cdots & (\underline{x_{k+L}}) \end{matrix}$$

$$\Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} 0 & \mathbb{F}_k & -I & \cdots & 0 \\ \mathbb{H}_1^{L_a} & 0 & \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{L_a} & 0 & \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L}} \end{pmatrix} \tag{4.16}$$

$${}^{I}B_{old} = \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} 0 & \mathbb{F}_k \\ \mathbb{H}_1^{L_a} & 0 \\ \vdots & \vdots \\ \mathbb{H}_{n_o}^{L_a} & 0 \end{pmatrix} = \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} 0 & \mathbb{F}_k \\ \mathbb{H}^{L_a} & 0 \end{pmatrix}, \quad \mathbb{H}^{L_a} \doteq \begin{pmatrix} \mathbb{H}_1^{L_a} \\ \vdots \\ \mathbb{H}_{n_o}^{L_a} \end{pmatrix} \tag{4.17}$$

$$B_{new} = \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} -I & \cdots & 0 \\ \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L}} \end{pmatrix} = \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix},$$

$$\mathbb{F} \doteq \begin{pmatrix} -I & \cdots & 0 \end{pmatrix}, \quad \mathbb{H}^{X_{new}} \doteq \begin{pmatrix} \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L}} \end{pmatrix} \tag{4.18}$$

where $\mathbb{H}_i^{L_a} \doteq \nabla_{L_a} h_i$ is Jacobian of i-th observation factor h_i from Eq. (4.9) with respect to variables L_a, and thus only one of its block-columns, corresponding to observed landmark, is being not-zero. $\mathbb{H}_i^{x_{k+l}} \doteq \nabla_{x_{k+l}} h_i$ is Jacobian of h_i with respect to x_{k+l}, and therefore is being non-zero only if factor's observation was taken from pose x_{k+l}. Matrix Ψ_{conn} is block-diagonal, combining all noise matrices of $\mathcal{F}^{conn}(a)$ factors.

As can be seen from the above, the Jacobian matrices ${}^{I}B_{old}$, B_{new} and D_{new} are sparse and can be efficiently manipulated. More specifically, the information matrix of factor graph $G(a)$, $\Lambda_a = D_{new}^T \cdot D_{new}$, which is required in our approach, can be calculated fast as a product of sparse matrices $\Lambda_a = \widetilde{D}_{new}^T \cdot \Psi_{new}^{-1} \cdot \widetilde{D}_{new}$ due to the formulation in Eq. (4.14); additionally, it can be shown to be singular and block-tridiagonal.

The matrix C_1 from Eq. (3.24) can also be reduced to following form:

$$C_1 = I_{m_{conn}} + \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T & \mathbb{F}_k \cdot \Sigma_k^{M,\{x_k/L_a\}} \cdot (\mathbb{H}^{L_a})^T \\ \mathbb{H}^{L_a} \cdot \Sigma_k^{M,\{L_a/x_k\}} \cdot \mathbb{F}_k^T & \mathbb{H}^{L_a} \cdot \Sigma_k^{M,L_a} \cdot (\mathbb{H}^{L_a})^T \end{pmatrix} \cdot \Psi_{conn}^{-\frac{1}{2}} =$$

$$\Psi_{conn}^{-\frac{1}{2}} \cdot \left[\Psi_{conn} + \begin{pmatrix} \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T & \mathbb{F}_k \cdot \Sigma_k^{M,\{x_k/L_a\}} \cdot (\mathbb{H}^{L_a})^T \\ \mathbb{H}^{L_a} \cdot \Sigma_k^{M,\{L_a/x_k\}} \cdot \mathbb{F}_k^T & \mathbb{H}^{L_a} \cdot \Sigma_k^{M,L_a} \cdot (\mathbb{H}^{L_a})^T \end{pmatrix} \right] \cdot \Psi_{conn}^{-\frac{1}{2}} \doteq$$

$$\Psi_{conn}^{-\frac{1}{2}} \cdot C_2 \cdot \Psi_{conn}^{-\frac{1}{2}}, \quad (4.19)$$

$$C_2 \doteq \Psi_{conn} + \begin{pmatrix} \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T & \mathbb{F}_k \cdot \Sigma_k^{M,\{x_k/L_a\}} \cdot (\mathbb{H}^{L_a})^T \\ \mathbb{H}^{L_a} \cdot \Sigma_k^{M,\{L_a/x_k\}} \cdot \mathbb{F}_k^T & \mathbb{H}^{L_a} \cdot \Sigma_k^{M,L_a} \cdot (\mathbb{H}^{L_a})^T \end{pmatrix} \quad (4.20)$$

where $\Sigma_k^{M,\{x_k/L_a\}}$ is the prior cross-covariance between variables x_k and L_a.

Additionally, C_1's determinant and its inverse can be calculated through:

$$\left|C_1\right| = \frac{\left|C_2\right|}{\left|\Psi_{conn}\right|}, \quad C_1^{-1} = \Psi_{conn}^{\frac{1}{2}} \cdot C_2^{-1} \cdot \Psi_{conn}^{\frac{1}{2}}. \quad (4.21)$$

Next, we can calculate term $B_{new}^T \cdot C_1^{-1} \cdot B_{new}$ from Eq. (3.22) as:

$$B_{new}^T \cdot C_1^{-1} \cdot B_{new} = \begin{pmatrix} \mathbb{F}^T & (\mathbb{H}^{X_{new}})^T \end{pmatrix} \cdot \Psi_{conn}^{-\frac{1}{2}} \cdot \Psi_{conn}^{\frac{1}{2}} \cdot C_2^{-1} \cdot \Psi_{conn}^{\frac{1}{2}} \cdot \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix} =$$

$$\begin{pmatrix} \mathbb{F}^T & (\mathbb{H}^{X_{new}})^T \end{pmatrix} \cdot C_2^{-1} \cdot \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix} = \widetilde{B}_{new}^T \cdot C_2^{-1} \cdot \widetilde{B}_{new} \quad (4.22)$$

where $\widetilde{B}_{new} \doteq \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix}$ contains the Jacobian entries of B_{new} not weighted by factors' noise Ψ_{conn}. Then, the `unfocused` IG objective from Eq. (2.16) in SLAM setting is given by

$$J_{IG}(a) = \frac{n' \cdot \gamma}{2} - \frac{1}{2} \ln\left|\Psi_{conn}\right| + \frac{1}{2} \ln\left|C_2\right| + \frac{1}{2} \ln\left|\widetilde{B}_{new}^T \cdot C_2^{-1} \cdot \widetilde{B}_{new} + \widetilde{D}_{new}^T \cdot \Psi_{new}^{-1} \cdot \widetilde{D}_{new}\right|. \quad (4.23)$$

Above we have shown in detail how our *rAMDL* approach can be applied to information-based SLAM planning problem types. The derived Eq. (4.23) is very similar to the general solution from Eq. (3.22), having exactly the same runtime complexity. However, within both Eq. (4.23) and Eq. (4.20) we can see a clear separation between noise of factor model and the actual Jacobian entries. Such a separation can provide further theoretical insight about how different terms of the SLAM problem affect the information impact of candidate action $a = u_{k:k+L-1}$. Moreover it can provide a good starting point for derivation of $J_{IG}(a)$'s gradient with respect to $u_{k:k+L-1}$ which in turn can be used for gradient-descend algorithms that search for locally optimal controls [21, 41]. Note the variable ordering in the above equation serves only for visualization; the derivation remains valid for an arbitrary variable ordering.

Additionally, for the sake of completeness we also provide SLAM-specific solution for `focused` cases, where we consider either reducing entropy of the last pose ($X^F_{k+L} \equiv x_{k+L}$) or of all the mapped landmarks ($X^F_k \equiv L_k$). The corresponding derivation can be found in Appendix (9.7 and 9.8).

4.3 Graph Reduction

It is a known fact that in long-term SLAM applications state dimension of smoothing techniques can grow unboundedly. In such cases even most efficient state-of-the-art estimation algorithms like iSAM2 [24] can become slow and will not support online operation. Approaches like graph reduction and graph sparsification try to tackle the problem by reducing the number of variables [16, 28, 35] and sparsifying entries of information matrix [4, 14, 31, 42], respectively.

Graph reduction requires first to select nodes to expel. In such cases, having a state vector X with variables $\{x_1, .., x_n\}$, it would be logical to remove the most uncertain node, say x_i, without which the rest of the variables $\overline{X_i} \doteq \{X \setminus x_i\}$ would have the smallest entropy $\mathcal{H}(\overline{X_i})$. In this section we outline a new approach for such a selection which is closely related to our *rAMDL* technique.

Similarly to the `focused` objective function from Eq. (2.17), the best choice for expelled variable x_i^* among state variables will minimize the following objective function:

$$x_i^* = \arg\min_{x_i \in X} J_{GR}(x_i) = \mathcal{H}(\overline{X_i}) = \frac{(n - n_x) \cdot \gamma}{2} - \frac{1}{2} \ln \left| \Lambda^{M, \overline{X_i}} \right|, \tag{4.24}$$

where $\mathcal{H}(\overline{X_i})$ is entropy of the state variables without x_i, and n_x is x_i's dimension.

Using Equation (3.9) from our approach, in order to calculate $\Lambda^{M, \overline{X_i}}$, we can reduce our objective function to

$$J_{GR}(x_i) = \frac{(n - n_x) \cdot \gamma}{2} - \frac{1}{2} \ln \left| \Lambda \right| + \frac{1}{2} \ln \left| \Lambda^{x_i} \right| \tag{4.25}$$

where Λ is information matrix of whole X, and Λ^{x_i} is its partition related to variable x_i.

Given that all x_i variables have the same dimension n_x, eventually we can conclude that optimal x_i^* will also minimize

$$x_i^* = \arg\min_{x_i \in X} J_{GR}(x_i) = \ln \left| \Lambda^{x_i} \right| \tag{4.26}$$

which practically implies calculating determinant of every partition Λ^{x_i} and choosing the state variable x_i with minimal determinant value. In case where all x_i are scalars, $\left| \Lambda^{x_i} \right|$ is just a value from diagonal of the information matrix Λ. In case where x_i's dimension is n_x, we will have to calculate determinants of n matrices, each one of dimension $n_x \times n_x$. Taking into account that n_x is usually not big at all (e.g. 3D pose has dimension of 6), the overall calculation is very fast and is just $O(n)$.

Chapter 5

Alternative Approaches

We compare the presented *rAMDL* approach with two alternatives, namely *From-Scratch* and *iSAM* techniques.

In *From-Scratch*, the posterior information matrix Λ_{k+L} is computed by adding new information $A^T \cdot A$, followed by calculation of its determinant. In `focused` scenario the marginal information matrix of X^F_{k+L} is retrieved through Schur Complement performed on Λ_{k+L}, and its determinant is then computed. The complexity of both focused and unfocused scenarios is governed by the term $O(N^3)$, with N being posterior state dimension.

The second alternative, uses the iSAM algorithm [24] to incrementally update the posterior. Here the (linearized) system is represented by a square root information matrix R_k, which is encoded, while exploiting sparsity, by the Bayes tree data structure. The posterior matrix R_{k+L} is acquired (e.g. via Givens rotations [24] or another incremental factorization update method), and then the determinant is calculated $|\Lambda_{k+L}| = \prod_{i=1}^{N} r_{ii}^2$, with r_{ii} being the ith entry on the diagonal of triangular R_{k+L}. For focused case, the marginal covariance matrix of X^F_{k+L} is computed by recursive covariance per-entry equations [23] that exploit sparsity of matrix R_{k+L}. The time complexity of this approach grows with state dimension and is discussed in more detail in [23, 24].

While the *iSAM* technique outperforms batch *From-Scratch*, it still requires calculating R_{k+L} for each action, which can be expensive, particularly in loop closures, and requires copy/clone of the original matrix R_k. In contrast, in *rAMDL*, the per candidate action calculation (e.g. in Eq. (3.20)) has constant complexity in general, given the prior marginal covariance terms that are calculated only once.

Chapter 6

Results

In this section we evaluate the performance of the proposed approach and compare it to alternative approaches considering *unfocused* and *focused* instantiations of several fundamental problems: sensor deployment, measurement selection, and autonomous navigation in unknown environments.

In sensor deployment, each candidate action represents a set of possible locations for deploying a sensor, with a single sensor deployment corresponding to a unary factor. We consider a nonmyopic setting and let each candidate action represent 2 sensor locations. In the measurement selection problem, we consider a greedy decision making paradigm in the context of aerial visual SLAM with pairwise factors.

Further, we present simulation results of applying our approach to autonomous navigation in unknown environments (both *unfocused* and *focused* cases) on synthetic and real-world datasets. The robot has to visit a sequence of goals while minimizing an objective function comprising two terms (to be defined in the sequel): distance to goal, and an uncertainty metric. Candidate actions are nonmyopic and involve multiple new and old state variables.

In all cases, the presented simulations reflect the performance of different approaches developed within this research, and alternative methods that are described in Chapter 5. In Table 6.1 we summarize the considered approaches in each of the above problems, and refer to appropriate equations for each case.

The code is implemented in Matlab; for measurement selection and autonomous navigation we use the GTSAM library [10, 24]. All scenarios were executed on a Linux machine with i7 2.40 GHz processor and 32 Gb of memory.

6.1 Sensor Deployment (*focused* and *unfocused*)

In this section we apply our approach *rAMDL* to the sensor deployment problem, considering both *focused* and *unfocused* instantiations of this problem (see Section 4.1 for detailed formulation). The prior of the sensor field is represented by information matrix Λ and it is dense as usual in problem of sensor deployment.

We compare our *rAMDL* approach against the batch *From-Scratch* technique that is described

Problem	Approach	Equations/Section
Sensor Deployment, Section 6.1	*rAMDL* `Unfocused` *rAMDL* `Focused` *Sequential rAMDL* *Partitions* *From-Scratch*, `Unfocused` & `Focused`	Eq. (3.7) Eq. (3.13) Eq. (4.6), (4.7) Givens Rotations & Eq. (3.10) Chapter 5
Measurement Selection, Section 6.2	*rAMDL* `Unfocused` *iSAM* `Unfocused`	Eq. (3.7) Chapter 5
Autonomous Navigation, Section 6.3	*rAMDL* `Unfocused` *rAMDL-Extended* `Unfocused` *rAMDL* `Focused New` *rAMDL-Extended* `Focused New` *rAMDL* `Focused Old` *rAMDL-Extended* `Focused Old` *From-Scratch*, `Unfocused` & `Focused` *iSAM*, `Unfocused` & `Focused`	Eq. (3.20), (3.21) Eq. (3.22), (3.24) Eq. (3.25), (3.21) Eq. (3.27), (3.24) Eq. (3.28), (3.29) Eq. (3.30), (3.31) Chapter 5 Chapter 5

Table 6.1: Considered approaches in different problems from Chapter 6, along with their appropriate equations.

in Chapter 5, as also against the *Sequential rAMDL* described in Section 4.1 which does not require marginal covariance computation at each decision.

While decision making involves evaluating action impact for all candidate actions $\mathcal{A}$, we first analyze action impact calculation ($J_{IG}(a)$) for a single candidate $a \in \mathcal{A}$, comparing *rAMDL* to the *From-Scratch* approach for the `unfocused` case. Figure 6.1 shows these timing results as a function of state dimension n (Figure 6.1a) and as function of Jacobian A's height m (Figure 6.1b). As expected, n effects running time of both the *From-Scratch* technique and calculation of Σ_k (inverse of Λ_k which is dense in case of sensor deployment), while m only effects calculation of IG objective of *rAMDL* (red line).

One might think, based on Figures 6.1a-6.1b, that the proposed approach is slower than *From-Scratch* alternative because of the time needed for inverse calculation to get Σ_k. Yet, it is exactly here that our calculation re-use paradigm comes into play (see Section 3.4): this calculation is performed only *once* for all candidate actions $\mathcal{A}$, while, given Σ_k, calculating IG for each action is no longer a function of n.

The substantial reduction in running time of our approach, compared to the *From-Scratch* approach, can be clearly seen in Figure 6.1c, which considers the entire decision making problem, i.e. evaluation of all candidate actions $\mathcal{A}$. The figure shows running time for sequential decision making, where at each time instant we choose the best locations of 2 sensors, with around $|\mathcal{A}| = 10^5$ candidate actions. The number of all sensor locations is $n = 625$ in this example. Overall, 15 sequential decisions were made. As seen, decision making using our approach requires only about 5 seconds, while the the *From-Scratch* approach requires about 400 seconds.

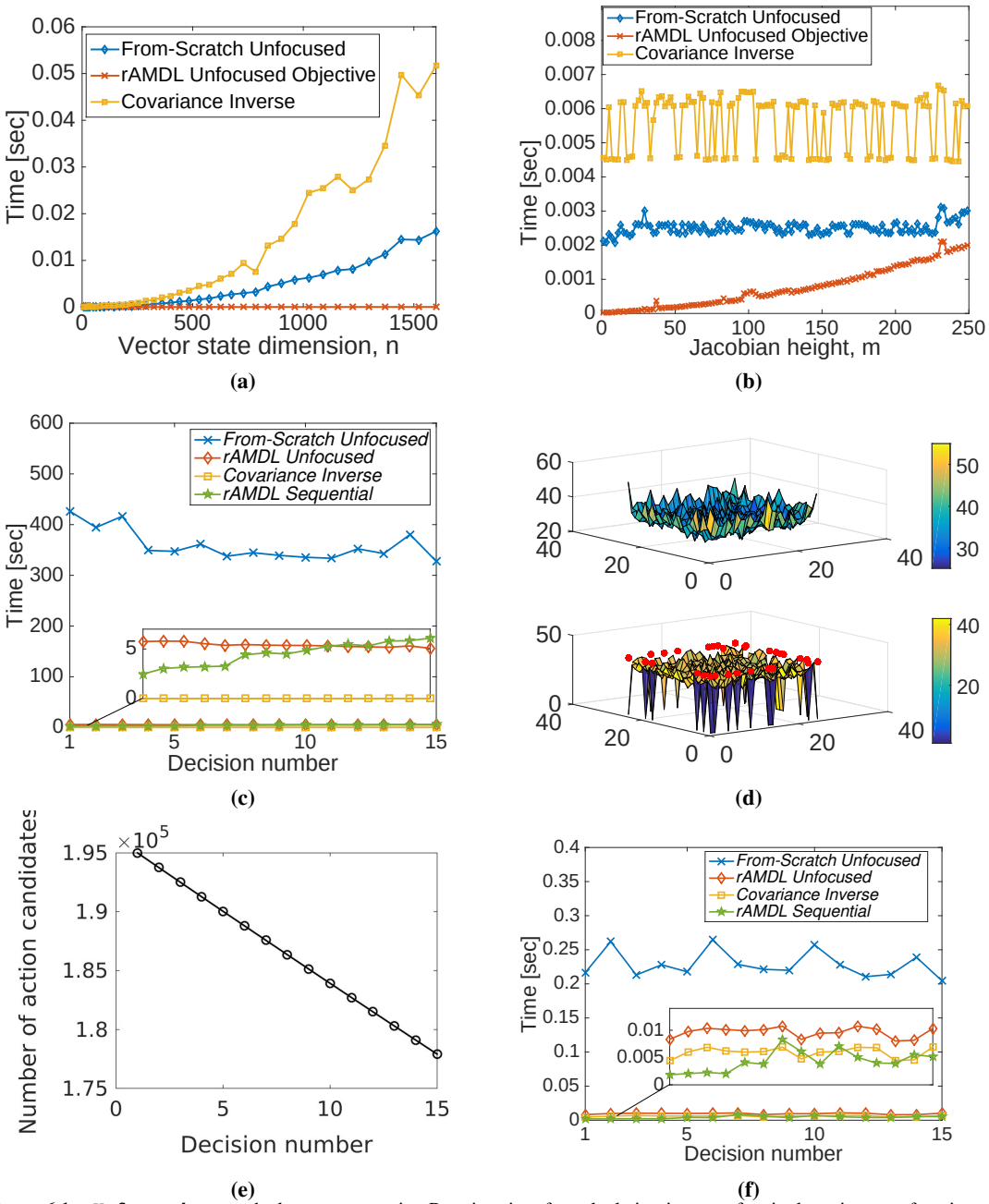

Figure 6.1: `Unfocused` sensor deployment scenario. Running time for calculating impact of a single action as a function of state dimension n (a) and as a function of Jacobian A's height m (b). In (a), $m = 2$, while in (b) $n = 625$. *rAMDL* `Unfocused Objective` represents only calculation time of candidates' impacts (IG objective for all actions), without one-time calculation of prior covariance; *Covariance Inverse* represents the time it took to calculate covariance matrix Σ_k from dense information matrix Λ_k, $\Sigma_k = \Lambda_k^{-1}$. (c) Running time for sequential decision making, i.e. evaluating impact of all candidate actions, each representing candidate locations of 2 sensors. (d) prior and final uncertainty of the field, with red dots marking selected locations. (e) number of action candidates per decision. (f) running time for sequential decision making, with number of candidates limited to 100.

Sequential rAMDL technique is not always faster than *rAMDL* , as can be seen in Figure 6.1c. As described in Section 4.1 this technique will be more superior in cases where covariance calculation takes significant part of whole decision calculation. We can see that this is the case in Figure 6.1f, where number of candidates is limited to 100, and where the covariance calculation time is the biggest part in the decision making of *rAMDL* approach. There we can see that

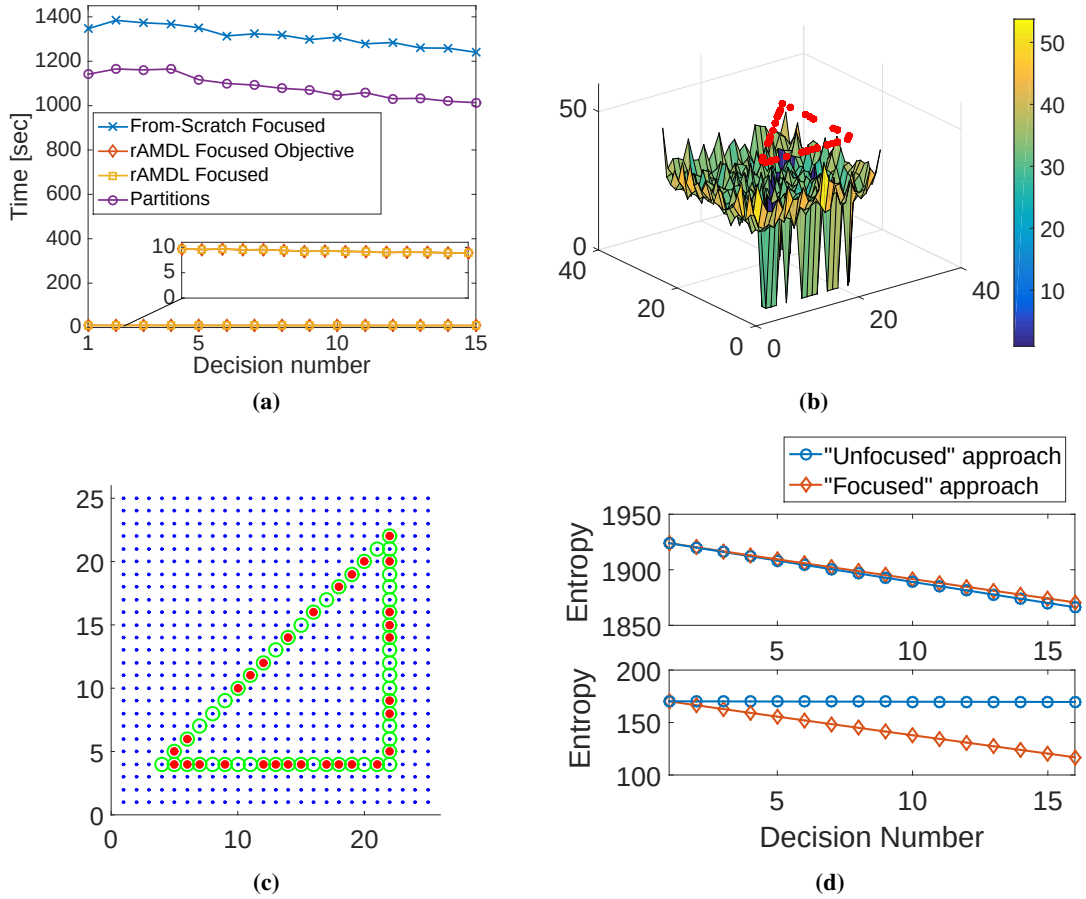

Figure 6.2: `Focused` sensor deployment scenario, (a) overall time it took to make decision with different approaches; *rAMDL* `Focused Objective` represents only calculation of candidates' impacts (IG objective for all actions) while *rAMDL* `Focused` - both one-time calculation of prior covariance Σ_k and candidates' evaluation. (b) Final uncertainty of the field, with red dots marking selected locations. (c) `Focused` set of variables (green circles) and locations selected by algorithm (red dots). (d) Overall system entropy (above) and entropy of `focused` set (bottom) after each decision, with blue line representing `unfocused` algorithm, and red line - `focused` algorithm. Note - all `unfocused` methods make exactly the same decisions, with difference only in their runtime complexity. Same is also true for all `focused` methods.

Sequential rAMDL provides better performance than all other alternatives.

We now consider the `focused` version of the sensor deployment problem (Eq. 2.17). In other words, the goal is to find sensor locations that maximally reduce uncertainty about chosen `focused` variables X^F. We have 54 such variables, which are shown in Figure 6.2c, while the rest of the problem setup remains identical to the `unfocused` case.

In Figure 6.2 we show the corresponding results of *rAMDL* , compared to the *From-Scratch*. The latter first calculates, for each candidate action, the posterior $\Lambda^+ = \Lambda + A^T A$, followed by calculation of Schur complement $\Lambda^{M,F}$ of the `focused` set X^F, and its determinant $\left|\Lambda^{M,F}\right|$ in order to get $J^F_{\mathcal{H}}(a)$ (Eq. 2.17). We also compare to an additional approach, termed *Partitions*, which uses Givens rotations to compute R^+ and instead of performing Schur complement, calculates the posterior entropy of the `focused` set via Eq. (3.10). This equation is one of our main contributions, being an essential step in the derivation of our approach, and we show here that comparing to *From-Scratch* technique, the *Partitions* approach is considerably faster. Our `focused` approach applies the matrix determinant lemma, transforming Eq. (3.10) to Eq. (3.13),

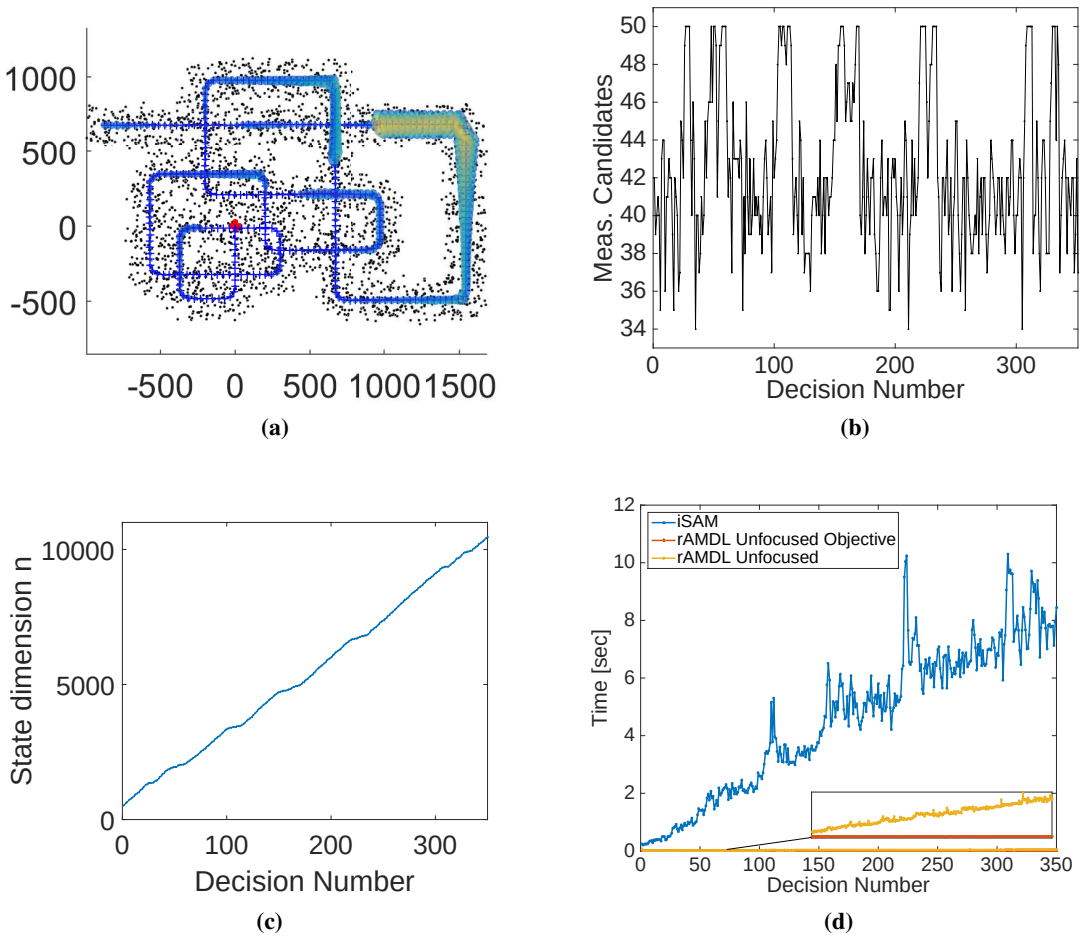

Figure 6.3: Measurement selection scenario, (a) simulated trajectory of robot; black dots are the landmarks, blue marks and surrounding ellipses are the estimated trajectory along with the uncertainty covariance for each time instant, red mark is robot's initial pose; (b) number of measurement candidates per decision; (c) state's dimension n per decision; (d) overall time it took to evaluate impacts of pose's all measurements, with different approaches; *rAMDL* `Unfocused Objective` represents only calculation of candidates' impacts (IG objective for all actions) while *rAMDL* `Unfocused` - both one-time calculation of marginal covariance $\Sigma_k^{M,X_{All}}$ and candidates' evaluation.

which, together with the *re-use* concept (Section 3.4), makes it possible to drastically reduce running time as shown in Figure 6.2a (10 seconds versus about 1000 in *Partitions* and 1300 in *From-Scratch*).

6.2 Measurement Selection in SLAM

In this section we consider a measurement selection problem (see Section 3.5) within a visual aerial SLAM framework, where one has to choose the most informative image feature observations from the numerous image features typically calculated for each incoming new image.

We demonstrate application of our approach in this problem, which, in contrast to the sensor selection problem, involves pairwise factors of the type $p(z_{i,j}|x_i, l_j)$, relating between an image observation $z_{i,j}$, camera pose x_i and landmark l_j.

A top view of the considered aerial scenario is shown in Figure 6.3a: an aerial vehicle performs visual SLAM, mapping the environment and at the same time localizing itself. The figure shows the landmarks and the estimated trajectory, along with the uncertainty covariance for each time instant. One can clearly see the impact of loop closure observations on the latter. In the considered scenario there are about 25000 landmarks and roughly 500 image features in each view.

The number of image features that correspond to previously seen landmarks is relatively small (around 30-50, see Figure 6.3b), which corresponds to a much smaller set of actions $\mathcal{A}$ compared to the sensor deployment problem (Section 6.1) where the cardinality of $\mathcal{A}$ was huge (10^5). Such a dataset was chosen on purpose in order to show the behavior of the proposed algorithm in domains with small number of candidates. Also, in this scenario the actions are myopic since the measurements are greedily selected.

Additionally, as opposed to the sensor deployment problem, in the current problem, state dimensionality n grows with time as more poses and landmarks are added into inference (see Figure 6.3c) and the information matrix is sparse.

Figure 6.3d shows the timing results for choosing 10 most informative image observations comparing the proposed *rAMDL* to the *iSAM* approach (computing posterior square root information matrix using iSAM, and then calculating determinant, see Chapter 5). This BSP problem is solved sequentially, each time a new image is acquired. As seen, our approach *rAMDL* is substantially faster than the *iSAM*, while providing identical results (the same decisions). In particular, running time of the *iSAM* approach for the last time index with $n = 10000$ state dimensionality, is around 7 seconds. In contrast, *rAMDL* takes about 0.05 seconds: calculation time of action impacts via calculation re-use is negligible (red line), while the one-time calculation of marginal covariance $\Sigma_k^{M,X_{All}}$ (yellow line) is efficiently performed, in the current implementation, via sparse factorization techniques using GTSAM [10, 24].

6.3 Autonomous Navigation in Unknown Environment

In this section we present simulation results of applying our approach to autonomous navigation in unknown environments (both `unfocused` and `focused` cases) on synthetic and real-world datasets.

In the synthetic scenario (Figure 6.4c), the robot's task is to visit a predefined set of goals $\mathcal{G} = \{G_1, .., G_{14}\}$ in unknown environment while reducing an uncertainty metric. More specifically, the state vector X_k contains all robot poses and landmarks mapped till time t_k (see Section 4.2). At each point of time, the robot autonomously selects an optimal non-myopic action $a = u_{k:k+L-1}$, performs its first control u_k and subsequently observes landmarks in radius of 900 meters from its new position. The landmarks can be either old (seen before) or new (met first time). Next, a SLAM solution is calculated given these new observations and a motion model. To that end, the factor graph from the previous inference time is updated with new observation and motion model factors, and new variable nodes, representing current robot pose and new landmarks, are added (see Section 4.2). Afterwards, next action is chosen and executed,

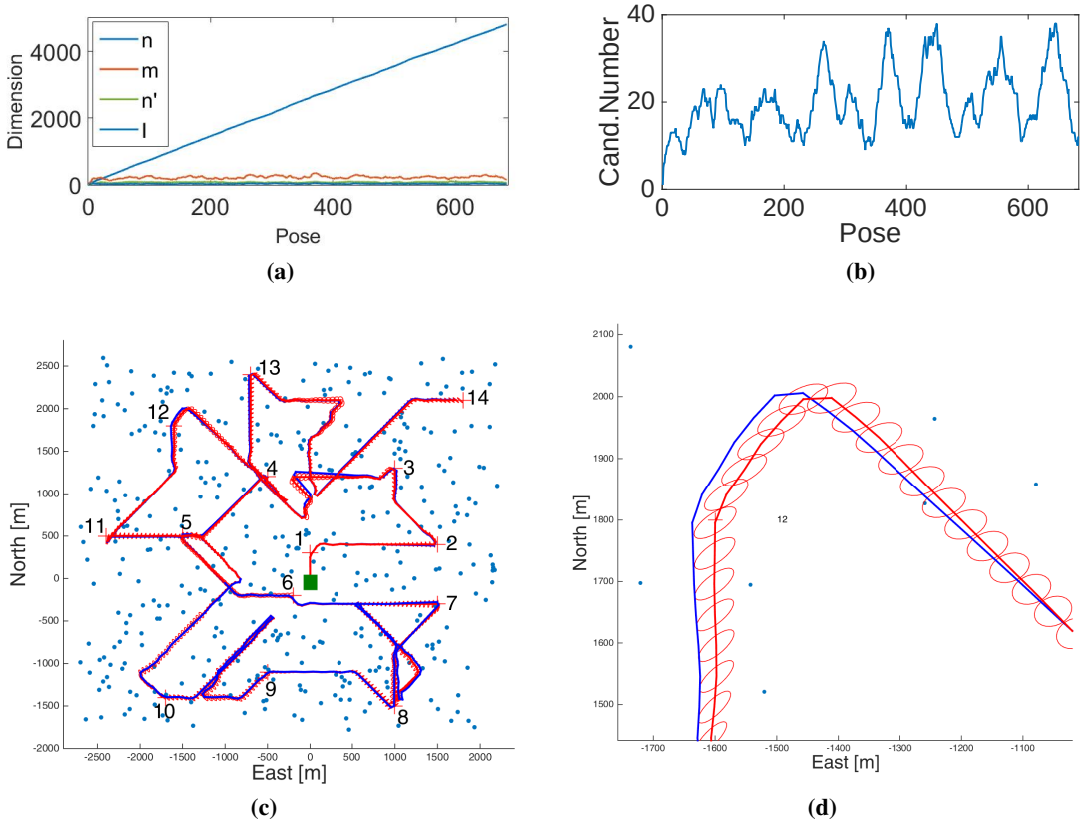

Figure 6.4: *Focused* BSP scenario with *focused* robot's last pose. (a) Dimensions of the BSP problem (state dimension, average number of new factor terms, average number of new variables, average number of old involved variables) at each time; (b) Number of action candidates at each time; (c) Final robot trajectory. Blue dots are mapped landmarks, red line with small ellipses is estimated trajectory with pose covariances, blue line is the real trajectory, red pluses with numbers beside them are robot's goals. Green mark is robot's start position; (d) Zoom-in of robot's trajectory near goal 12.

and so on.

The set of candidate actions $\mathcal{A}$ contains one action that navigates the robot from its current pose x_k to the current goal G_i from a predefined set $\mathcal{G}$ (see Figure 6.4c); it also contains a set of "loop-closure" actions which are generated in the following way. We start by taking all mapped landmarks in radius of 1000 meters from the robot's current pose. We cluster these landmarks, similarly to [26], and obtain a set of landmark clusters. Each cluster's center g_{cl} represents a "loop-closure" goal and contributes a "loop-closure" action $a_{cl} = u_{k:k+L-1}$ that navigates robot the from x_k to g_{cl}.

Each action in $\mathcal{A}$, taking the robot from x_k to location g, is constructed by first discretizing the map into grid and thereafter searching for optimal trajectory from current position to g using an A^* search algorithm, similarly to [21, 26]. The optimal candidate action is chosen by evaluating an objective which has the following two terms: distance to current goal G_i, and a term of uncertainty:

$$J(a) = d(x_{k+L}, G_i) + J^F_{\mathcal{H}/IG}(a). \tag{6.1}$$

In scenarios from Figures 6.4, 6.5, 6.6 and 6.8 we consider as term of uncertainty the entropy $J^F_{\mathcal{H}}(a)$ of the last pose x_{k+L} in planning segment (Section 3.3.3), while in scenario from Figure

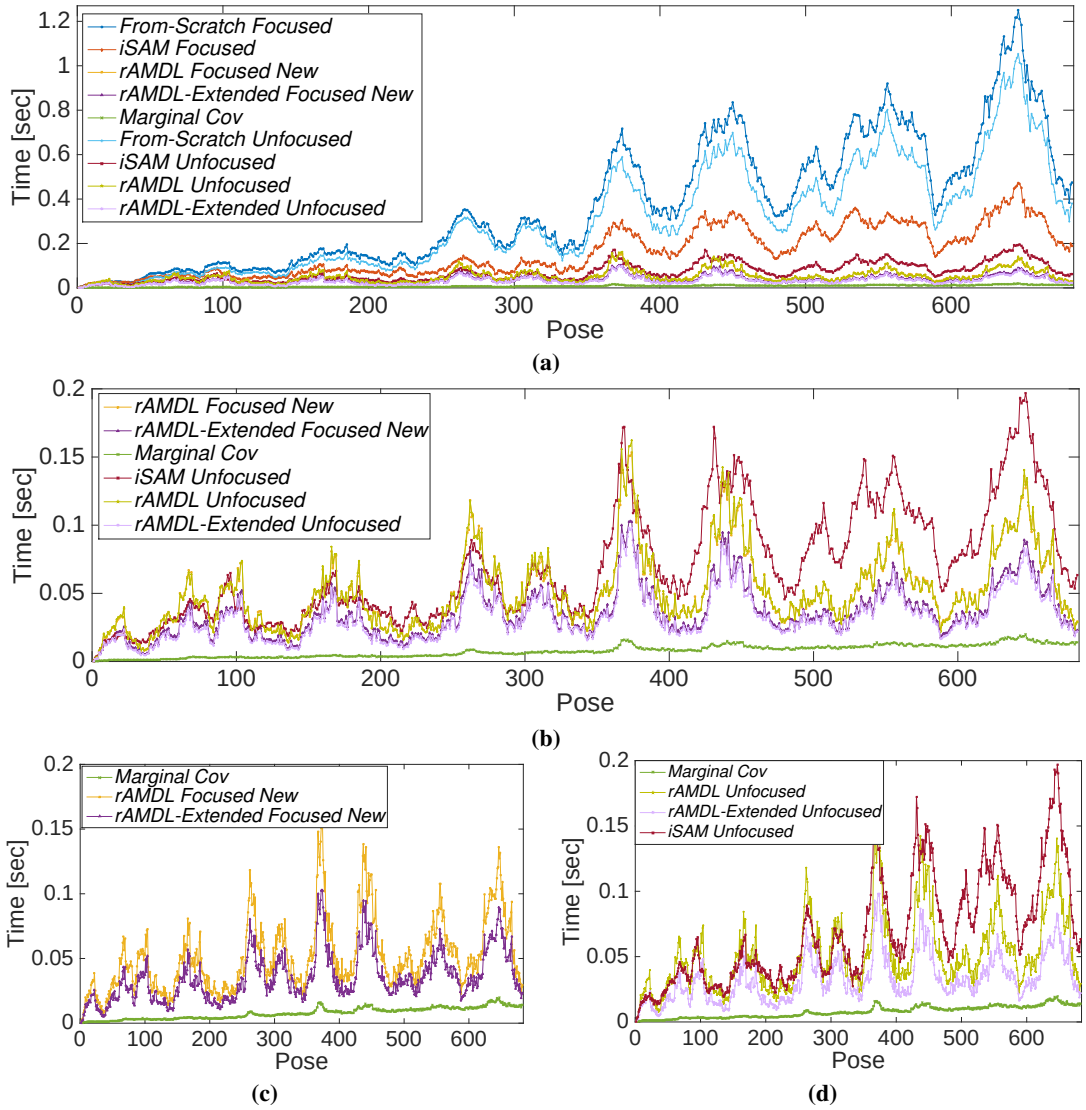

Figure 6.5: *Focused* BSP scenario with *focused* robot's last pose. (a) Running time of planning, i.e. evaluating impact of all candidate actions, each representing possible trajectory; Results are shown both for *focused* and *unfocused* cases; (b) Zoom of fastest approaches from (a); (c) *Focused* approaches from (b). Note that *iSAM Focused* is not depicted because as seen in (a) it is much slower comparing to other *focused* techniques; (d) *Unfocused* approaches from (b). The lowest line, labeled *Marginal Cov*, represents time it took to calculate prior marginal covariance $\Sigma_k^{M,X_{All}}$ in *rAMDL* approach (see Section 3.4). As can be seen, while *rAMDL* technique (*Unfocused* and *Focused*) is faster than *From-Scratch* and *iSAM*, the *rAMDL-Extended* gives even better performance. Further, it is interesting to note that performance of *Unfocused* and *Focused* *rAMDL* is almost the same, as also performance of *Unfocused* and *Focused* *rAMDL-Extended* .

6.7 we use instead the IG of mapped till now landmarks $J_{IG}^{F}(a)$ (Section 3.3.3). Note that the running time presented in the figures refers *only* to the uncertainty term, since it is the focus of this research and because calculation complexity of the first term (euclidean distance $d(x_{k+L}, G_i)$) is relatively insignificant. As can be seen from above, we consider a non-myopic setting and let each candidate action represent trajectory of various length. Limiting the clustering process to a specific radius is done in order to bound the horizon lag of candidate actions.

In parallel, in scenarios from Figures 6.5 and 6.6, an *unfocused* uncertainty objective $J_{IG}(a)$ is calculated (Section 3.3.2), mainly for the purpose of performance comparison between

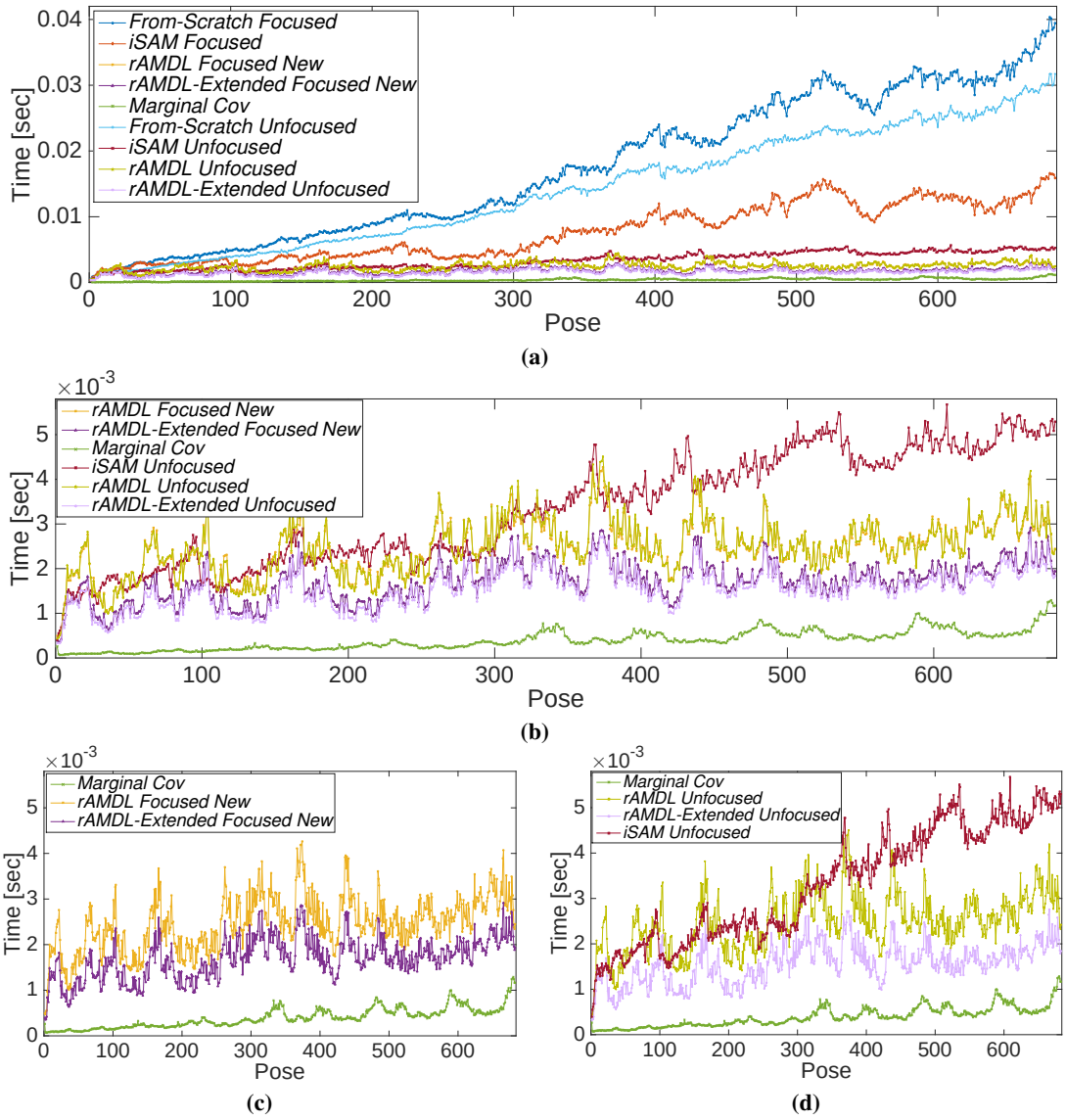

Figure 6.6: `Focused` BSP scenario with `focused` robot's last pose. Running times from Figure 6.5 normalized by number of candidates.

`focused` and `unfocused` cases. The robot's motion is controlled only by `focused` objective function.

Four techniques were applied to solve the planning problem - more common techniques *From-Scratch* and *iSAM* (Chapter 5), and the proposed techniques - our general approach *rAMDL* , and its extension *rAMDL-Extended* that exploits the Jacobian inner structure from Eq. (3.17) (see Table 6.1, and Sections 3.3.2 and 3.3.3). The calculated values of the objective function were numerically compared to validate that all four approaches are calculating exactly the same metric, thus yielding the same decisions and only differ in running time.

In Figures 6.5-6.6 it can be clearly seen that while *iSAM* is faster than *From-Scratch*, the running time of both techniques is growing with state dimensionality, as was mentioned before. On the other hand, the running time of *rAMDL* approach is shown to be bounded, due to horizon

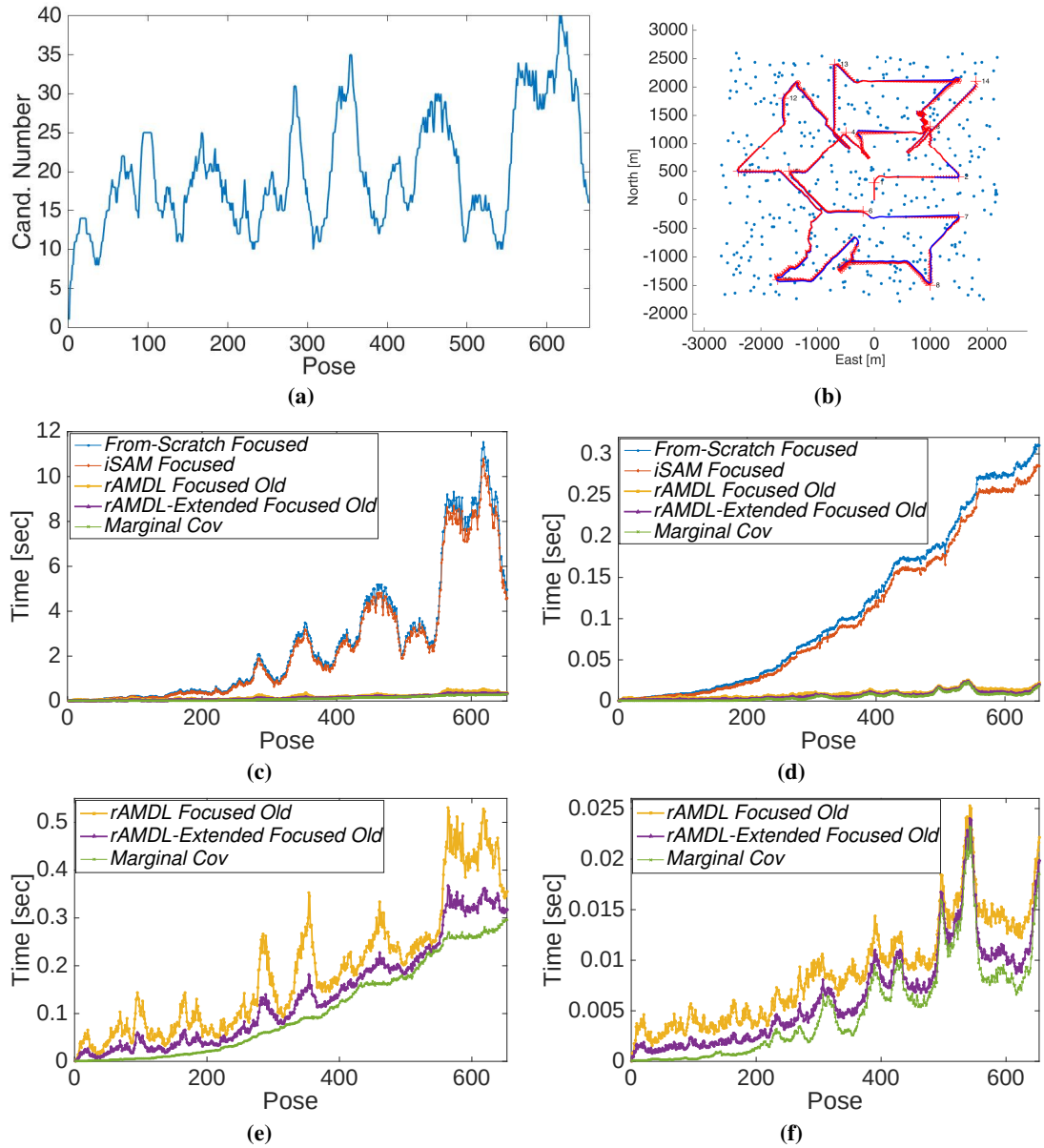

Figure 6.7: `Focused` BSP scenario with `focused` landmarks. (a) Number of action candidates at each time; (b) Final robot trajectory; (c) Running time of planning, i.e. evaluating impact of all candidate actions, each representing possible trajectory; (d) Running time from (c) normalized by number of candidates; (e) Zoom of fastest approaches from (c); (f) Zoom of fastest approaches from (d). The lowest line, labeled *Marginal Cov*, represents time it took to calculate prior marginal covariance $\Sigma_k^{M,X_{All}}$ in *rAMDL* approach (see Section 3.4).

lag of all candidate actions being limited (see Figure 6.4a). The number of candidate actions in our scenario is around 20 at each planning phase (Figure 6.4b). Even with such relatively small candidate set *rAMDL* approach is faster by order than its alternatives *iSAM* and *From-Scratch*, while *rAMDL-Extended* approach is the fastest between all of them. This trend appears to be correct for both `focused` and `unfocused` objective functions, though for the later, *iSAM* comes very close to *rAMDL* technique.

While comparing running time of both *From-Scratch* and *iSAM* in `focused` and `unfocused` objective functions, it is easy to see that `unfocused` case is evaluated much faster. The reason

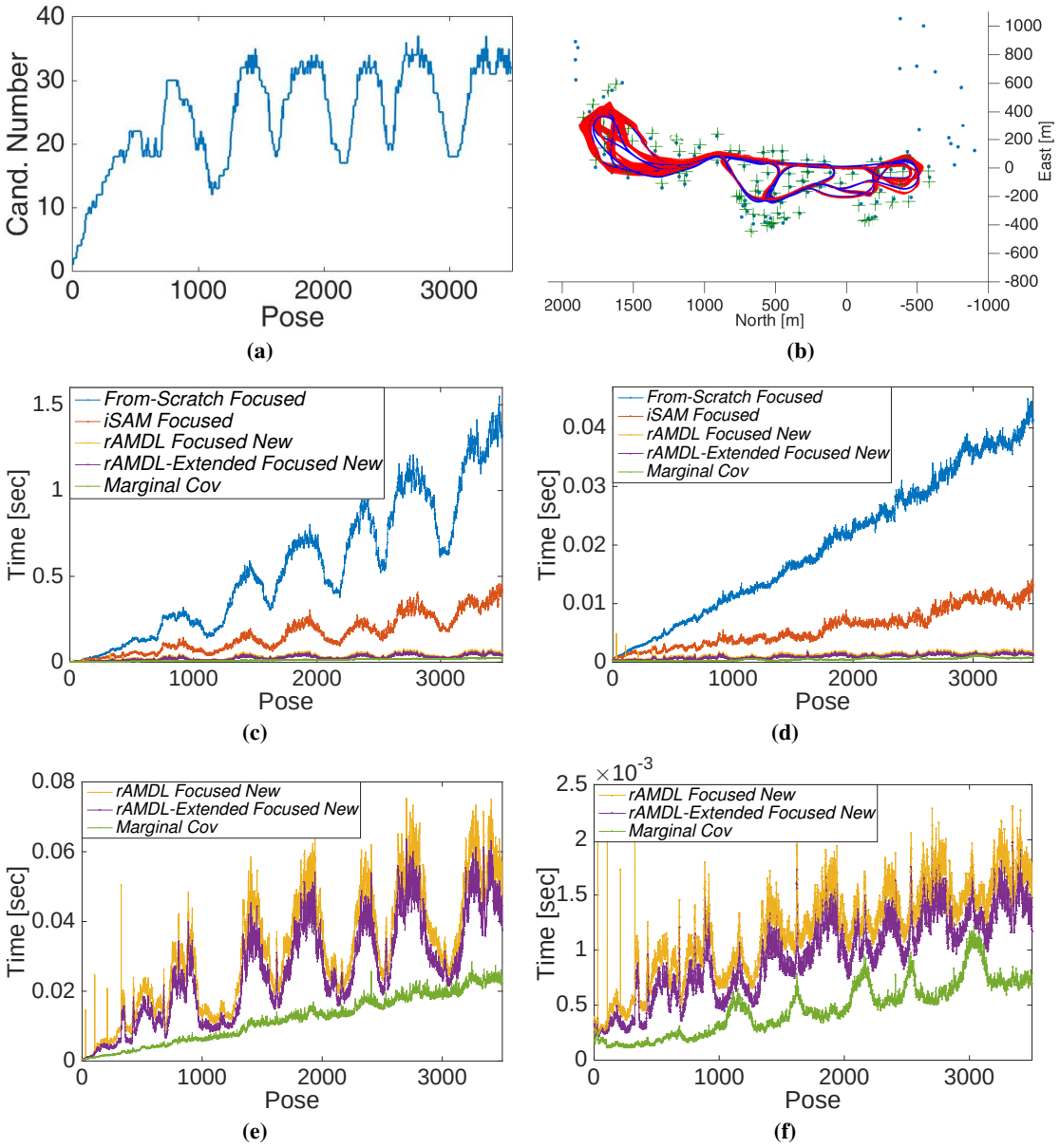

Figure 6.8: *Focused* BSP scenario with *focused* robot's last pose, using Victoria Park dataset. (a) Number of action candidates at each time; (b) Final robot trajectory; (c) Running time of planning, i.e. evaluating impact of all candidate actions, each representing possible trajectory; (d) Running time from (c) normalized by number of candidates; (e) Zoom of fastest approaches from (c); (f) Zoom of fastest approaches from (d). The lowest line, labeled *Marginal Cov*, represents time it took to calculate prior marginal covariance $\Sigma_k^{M,X_{All}}$ in *rAMDL* approach (see Section 3.4).

for this is that *focused* calculations contain computation of marginal covariance of *focused* variable (last pose x_{k+L}) for each candidate action, which requires marginalization over the posterior information matrix Λ_{k+L}. Whereas this can be performed efficiently by exploiting the sparsity of matrix Λ_{k+L} [23], the time complexity is significantly affected by variable elimination ordering of iSAM algorithm [24]. While in our simulation we did not modify the default ordering of iSAM (COLAMD heuristic), different strategies of ordering can be a point for future investigation.

In contrast, for *rAMDL* approach both *unfocused* and *focused* objective functions

(Eq. (3.20) and (3.25)) have a similar complexity, which is supported by the shown times. The same is correct for *rAMDL-Extended* approach (Eq. (3.22) and (3.27)).

Next, we repeated our autonomous navigation scenario but this time, X^F_{k+L} contained only landmarks seen by time k (see Figure 6.7). The IG of such `focused` set X^F_{k+L} can be used as objective function for example in case when we want to improve 3D reconstruction quality. As can be seen in Figure 6.7, this `focused` set causes both *From-Scratch* and *iSAM* techniques to be much slower compared to their performance in the first scenario, where X^F_{k+L} contained only x_{k+L}. The reason for this is that X^F_{k+L}'s dimension is much higher here, representing dimensions of all landmarks, and computation of its marginal covariance is significantly more expensive. In contrast, performance of *rAMDL* has been barely changed thanks to *re-use* of calculations (see Section 3.4). Moreover, *rAMDL-Extended* performs even better than *rAMDL* , with candidate action impact evaluation being insignificant compared with the one-time calculation of marginal covariance, as can be seen in Figures 6.7e-6.7f.

We also performed a hybrid simulation where part of the real-world Victoria Park dataset [12] was used for offline planning (see Figure 6.8). At each timestep we collected candidate actions by clustering landmarks seen till that time, just as it was done in the first simulation. Further, we considered a focused objective function for each candidate with X^F_{k+L} containing only x_{k+L}. After evaluating all candidates, the robot was moved to the next pose according to the dataset. Recalling that our main contribution is to reduce time complexity, such an evaluation allowed us to compare time performance of all the considered techniques, despite not actually using the calculated actions in the hybrid simulation. As can be seen, also here *rAMDL* and *rAMDL-Extended* outperform both of their alternatives, *From-Scratch* and *iSAM*, keeping the same trends that were observed in previous simulations.

Chapter 7

Conclusions and Future Work

We developed a computationally efficient and exact approach for non-myopic *focused* and *unfocused* belief space planning (BSP) in both augmented and non-augmented settings, in high dimensional state spaces. As a key contribution we developed an augmented version of the well-known general matrix determinant lemma and use both of them to efficiently evaluate the impact of each candidate action on posterior entropy, without explicitly calculating the posterior information (or covariance) matrices. The second ingredient of our approach is the *re-use* of calculations, that exploits the fact that many calculations are shared among different candidate actions. Our approach drastically reduces running time compared to the state of the art, especially when the set of candidate actions is large, with running time being independent of state dimensionality that increases over time in many of BSP domains. The approach was examined in three problems, sensor deployment, measurement selection in visual SLAM and autonomous navigation in unknown environments, using both simulated and real-world datasets, and exhibiting in each superior performance compared to the state of the art, and reducing running time by several orders of magnitude (e.g. 5 versus 400 seconds in sensor deployment).

7.1 Future Work

In this work we have developed an novel approach to solve the information-theoretic BSP problem in computationally efficient way. Still, number of extensions and improvements may be done in order to further reduce the calculation requirements and in order avoid the maximal likelihood assumption that was taken through our approach. Below we will list some directions for the future research.

1. In many BSP scenarios different candidate actions will share some of their newly introduced factor terms. For example, in mobile robotics scenario where there are two trajectory candidates with some part of their trajectory being the same, both candidates will have same factors representing the shared trajectory part. For such scenario, it can be easily shown that the Jacobian matrices of both candidates will share some of their rows. It is possible to exploit this fact in order to provide additional re-use of calculation and reduce the *rAMDL* time-consumption even more.

2. In this thesis we have taken the maximal likelihood assumption while propagating the posterior belief. This assumption ignores the fact that obtaining measurements (e.g. capturing in camera frame the specific landmark while passing near it) is itself probabilistic process and its uncertainty should be accounted for. There are different ways to model and consider such uncertainty within the BSP solution, and it is another possible extension of *rAMDL* method.

3. The developed *rAMDL* technique is a general approach. As such, it could be also applicable to a multi-robot setting, at least in its centralized formulation. Still, it may be interesting to investigate if additional reduction of calculations can be achieved for the multi-robot case, and if a decentralized multi-robot formulation can be handled in a similar manner.

Chapter 8

Appendix - Related Publications

List of Publications

Publications based on this Master's research:

1. D. Kopitkov, V. Indelman. "Computationally Efficient Decision Making and Belief Space Planning in High-Dimensional State Spaces," in *The 5th Israeli Conference on Robotics (ICR)*, Herzliya, Israel, April 2016

2. D. Kopitkov, V. Indelman. "Computationally Efficient Active Inference in High-Dimensional State Spaces," in *AI for Long-term Autonomy, workshop in conjunction with IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016

3. D. Kopitkov, V. Indelman. "Computationally Efficient Decision Making Under Uncertainty in High-Dimensional State Spaces," *in International Conference on Intelligent Robots and Systems (IROS) 2016*

4. D. Kopitkov, V. Indelman. "Computationally Efficient Belief Space Planning via Augmented Matrix Determinant Lemma and Re-Use of Calculations," *in IEEE Robotics and Automation Letters (RA-L)*, 2017

5. D. Kopitkov, V. Indelman, "Computationally Efficient Belief Space Planning via Augmented Matrix Determinant Lemma and Re-Use of Calculations," *in International Conference on Robotics and Automation (ICRA 2017)*

6. D. Kopitkov, 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," *in International Journal of Robotics Research (IJRR)*, **submitted**

Chapter 9

Appendix - Proof of Lemmas

9.1 Proof of Lemma 3.3.1

Problem definition: Given a positive definite and symmetric matrix $\Lambda \in \mathbb{R}^{n \times n}$ (e.g. a prior information matrix) and its inverse Σ (prior covariance matrix), first Λ is augmented by k zero rows and columns and the result is stored in Λ^{Aug}. Then we have matrix $A \in \mathbb{R}^{m \times (n+k)}$ and calculate $\Lambda^{+} = \Lambda^{Aug} + A^{T} \cdot A$ (see Figure 2.1). We would like to express the determinant of Λ^{+} in terms of Λ and Σ.

We start by modeling the matrix Λ^{Aug} through Σ. By introducing k new variables, before adding any new constraints involving these variables, we can say that new variables are uncorrelated with old variables, and their uncertainty is infinite (nothing yet is known about them). Then the appropriate covariance matrix after augmentation, Σ^{Aug}, can just be created by adding k zero rows and columns to Σ, and setting new diagonal entries with parameter θ, noting that $\theta \to \infty$:

$$\Sigma^{Aug} = \begin{bmatrix} \Sigma & 0 \\ 0 & \theta \cdot I \end{bmatrix}. \tag{A1}$$

Next, note that inverse of Σ^{Aug} is given by the following expression:

$$(\Sigma^{Aug})^{-1} = \begin{bmatrix} \Lambda & 0 \\ 0 & \epsilon \cdot I \end{bmatrix}. \tag{A2}$$

where $\epsilon \doteq \frac{1}{\theta}$. Taking limit $\epsilon \to 0$ into account, we can see that the above equation converges to Λ^{Aug} as it was defined above. Then in the limit we will have that $(\Lambda^{Aug})^{-1} = \Sigma^{Aug}$. Also note that $\epsilon \to 0$, even that it never becomes zero, $\epsilon \neq 0$, thus if needed we can divide by ϵ without worry.

Taking into account the limit of ϵ, expressing Λ^{Aug} through Eq. (A2) will not change the problem definition. But such a model allows to inverse Λ^{Aug}:

$$(\Lambda^{Aug})^{-1} = \Sigma^{Aug} = \begin{bmatrix} \Sigma & 0 \\ 0 & \theta \cdot I \end{bmatrix}, \tag{A3}$$

and therefore to use the generalized matrix determinant lemma [13]:

$$\left|\Lambda^{+}\right| = \left|\Lambda^{Aug}\right| \cdot \left|I_m + A \cdot \Sigma^{Aug} \cdot A^T\right| = \left|\Lambda\right| \cdot \epsilon^k \cdot \left|I_m + A_{old} \cdot \Sigma \cdot A_{old}^T + \theta \cdot A_{new} \cdot A_{new}^T\right| \tag{A4}$$

where matrices $A_{old} \in \mathbb{R}^{m \times n}$ and $A_{new} \in \mathbb{R}^{m \times k}$ are constructed from A by retrieving columns of only old n variables and of only new k variables, respectively (see Figure 3.4).

Using the matrix determinant lemma once more, we get:

$$\left|\Lambda^{+}\right| = \left|\Lambda\right| \cdot \epsilon^k \cdot \left|\Delta\right| \cdot \left|I_k + \theta \cdot A_{new}^T \cdot \Delta^{-1} \cdot A_{new}\right| \tag{A5}$$

where $\Delta \doteq I_m + A_{old} \cdot \Sigma \cdot A_{old}^T$.

Moving ϵ inside the last determinant term, we have:

$$\left|\Lambda^{+}\right| = \left|\Lambda\right| \cdot \left|\Delta\right| \cdot \left|\epsilon \cdot I_k + \epsilon \cdot \theta \cdot A_{new}^T \cdot \Delta^{-1} \cdot A_{new}\right| \tag{A6}$$

Recalling that $\epsilon \to 0$ and $\epsilon \cdot \theta = 1$, we will get to:

$$\left|\Lambda^{+}\right| = \left|\Lambda\right| \cdot \left|\Delta\right| \cdot \left|A_{new}^T \cdot \Delta^{-1} \cdot A_{new}\right| \tag{A7}$$

And the augmented determinant ratio will be:

$$\frac{\left|\Lambda^{+}\right|}{\left|\Lambda\right|} = \left|I_m + A_{old} \cdot \Sigma \cdot A_{old}^T\right| \cdot \left|A_{new}^T \cdot (I_m + A_{old} \cdot \Sigma \cdot A_{old}^T)^{-1} \cdot A_{new}\right| = \left|\Delta\right| \cdot \left|A_{new}^T \cdot \Delta^{-1} \cdot A_{new}\right| \tag{A8}$$

■

9.2 Proof of Lemma 3.3.2

For A's structure given in Eq. (3.17), the Δ from Eq. (3.15) will be:

$$\Delta = I_m + \begin{pmatrix} B_{old} \\ 0 \end{pmatrix} \cdot \Sigma \cdot \begin{pmatrix} B_{old}^T & 0 \end{pmatrix} = \begin{pmatrix} \Delta_1 & 0 \\ 0 & I_{m_{new}} \end{pmatrix} \tag{A9}$$

where $\Delta_1 = I_{m_{conn}} + B_{old} \cdot \Sigma \cdot B_{old}^T$, $m_{conn} = \mathcal{M}(\mathcal{F}^{conn}(a))$ and $m_{new} = \mathcal{M}(\mathcal{F}^{new}(a))$.

Then we can conclude that:

$$\left|\Delta\right| = \left|\Delta_1\right| \tag{A10}$$

and that:

$$\Delta^{-1} = \begin{pmatrix} \Delta_1^{-1} & 0 \\ 0 & I_{m_{new}} \end{pmatrix} \tag{A11}$$

Now, by exploiting structure of A_{new} we will get:

$$A_{new}^T \cdot \Delta^{-1} \cdot A_{new} = \begin{pmatrix} B_{new}^T & D_{new}^T \end{pmatrix} \cdot \begin{pmatrix} \Delta_1^{-1} & 0 \\ 0 & I_{m_{new}} \end{pmatrix} \cdot \begin{pmatrix} B_{new} \\ D_{new} \end{pmatrix} = B_{new}^T \cdot \Delta_1^{-1} \cdot B_{new} + D_{new}^T \cdot D_{new} \quad \text{(A12)}$$

Then we can conclude that the augmented determinant lemma will be:

$$\frac{\left|\Lambda^+\right|}{\left|\Lambda\right|} = \left|\Delta_1\right| \cdot \left|B_{new}^T \cdot \Delta_1^{-1} \cdot B_{new} + D_{new}^T \cdot D_{new}\right| \quad \text{(A13)}$$

■

9.3 Proof of Lemma 3.3.3

Consider the scenario of `focused` Augmented BSP where the focused set X_{k+L}^F contains only newly added variables as defined in Section 3.3.3, with appropriate illustration shown in Figure 3.5.

First let us overview the various partitions of Jacobian A which are relevant to our current problem (Figure 3.5). A_{old}, A_{new}, ${}^I A_{old}$ and ${}^{\neg I} A_{old}$ were already introduced in previous sections. Further, we can partition A_{new} into A_{new}^F - columns of new variables that are focused $X_{new}^F \equiv X_{k+L}^F \in \mathbb{R}^{n_F}$, and A_{new}^U - columns of new unfocused variables X_{new}^U. Considering the figure, the set of all unfocused variables in X_{k+L} will be $X_{k+L}^R \doteq \{X_{old} \cup X_{new}^U\} \in \mathbb{R}^{n_R}$, such that $N = n_F + n_R$, providing another A's partition $A_R = [A_{old}, A_{new}^U]$.

Next, we partition the posterior information matrix Λ_{k+L} respectively to the defined above sets X_{k+L}^F and X_{k+L}^R as

$$\Lambda_{k+L} = \begin{bmatrix} \Lambda_{k+L}^R & \Lambda_{k+L}^{R,F} \\ (\Lambda_{k+L}^{R,F})^T & \Lambda_{k+L}^F \end{bmatrix}. \quad \text{(B14)}$$

As was shown in Eq. (3.9), determinant of the marginal covariance of X_{k+L}^F can be calculated through:

$$\left|\Sigma_{k+L}^{M,F}\right| = \frac{\left|\Lambda_{k+L}^R\right|}{\left|\Lambda_{k+L}\right|} \quad \text{(B15)}$$

Now let us focus on Λ_{k+L}^R term from the right side.

From Eq. (2.10) we can see that partition of posterior information matrix Λ_{k+L}^R can be calculated as:

$$\Lambda_{k+L}^R = \Lambda_k^{Aug,R} + A_R^T A_R \quad \text{(B16)}$$

where $\Lambda_k^{Aug,R}$ can be constructed by augmenting Λ_k with zero rows and columns in number of X_{new}^U's dimension (see Figure 3.5). The above equation has augmented determinant form as defined in Section 3.3.1, and so the augmented determinant lemma can be applied on it. Using

Eq. (3.15) we have:

$$\frac{\left|\Lambda^R_{k+L}\right|}{\left|\Lambda_k\right|} = \left|C\right| \cdot \left|(A^U_{new})^T \cdot C^{-1} \cdot A^U_{new}\right| \tag{B17}$$

where C is defined in Eq. (3.21).

Next, dividing Eq. (B17) by Eq. (3.19), we get

$$\left|\Sigma^{M,F}_{k+L}\right| = \frac{\left|\Lambda^R_{k+L}\right|}{\left|\Lambda_k\right|} \cdot \frac{\left|\Lambda_k\right|}{\left|\Lambda_{k+L}\right|} = \frac{\left|\Lambda^R_{k+L}\right|}{\left|\Lambda_{k+L}\right|} = \frac{\left|(A^U_{new})^T \cdot C^{-1} \cdot A^U_{new}\right|}{\left|A^T_{new} \cdot C^{-1} \cdot A_{new}\right|}, \tag{B18}$$

and posterior entropy of X^F_{k+L} is given by

$$J^F_{\mathcal{H}}(a) = \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln \left|(A^U_{new})^T \cdot C^{-1} \cdot A^U_{new}\right| - \frac{1}{2} \ln \left|A^T_{new} \cdot C^{-1} \cdot A_{new}\right|. \tag{B19}$$

■

Note that the variables inside information matrices do not have to be ordered in any particular way, and that the provided above proof is correct for any ordering whatsoever.

9.4 Proof of Lemma 3.3.4

For A's structure given in Eq. (3.17), term $A^T_{new} \cdot C^{-1} \cdot A_{new}$ from Eq. (B19), similarly to Eq. (A12), will be:

$$A^T_{new} \cdot C^{-1} \cdot A_{new} = B^T_{new} \cdot C_1^{-1} \cdot B_{new} + D^T_{new} \cdot D_{new}, \tag{B20}$$

where C_1 is defined in Eq. (3.24).

In the same way we can conclude (see Figure 3.5) that:

$$(A^U_{new})^T \cdot C^{-1} \cdot A^U_{new} = (B^U_{new})^T \cdot C_1^{-1} \cdot B^U_{new} + (D^U_{new})^T \cdot D^U_{new}. \tag{B21}$$

Therefore, posterior entropy of X^F_{k+L} from Eq. (B19) is given by

$$J^F_{\mathcal{H}}(a) = \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln \left|(B^U_{new})^T \cdot C_1^{-1} \cdot B^U_{new} + \Lambda^{U|F}_a\right| - \frac{1}{2} \ln \left|B^T_{new} \cdot C_1^{-1} \cdot B_{new} + \Lambda_a\right|, \tag{B22}$$

where $\Lambda_a = D^T_{new} \cdot D_{new}$ is information matrix of action's factor graph $G(a)$, and where $\Lambda^{U|F}_a = (D^U_{new})^T \cdot D^U_{new}$ is information matrix of variables X^U_{new} conditioned on $X^F_{new} \equiv X^F_{k+L}$ and calculated from distribution represented by $G(a)$.

■

Note that the variables inside information matrices do not have to be ordered in any particular way, and that the provided above proof is correct for any ordering whatsoever.

9.5 Proof of Lemma 3.3.5

Consider the scenario of `focused` Augmented BSP where the focused set X^F_{k+L} contains only old variables, with appropriate illustration shown in Figure 3.6 and with various partitions of Jacobian A defined in Section 3.3.3.

First, let us look again over relevant partitions of Jacobian A (Figure 3.6). The A_{old}, A_{new}, ${}^{I}A_{old}$ and ${}^{\neg I}A_{old}$ were already introduced in previous sections. From the figure we can see that ${}^{\neg I}A_{old}$ can further be separated into ${}^{\neg I}A^{U}_{old}$ - columns of old variables that are both not involved and unfocused (${}^{\neg I}X^{U}_{old}$), and ${}^{\neg I}A^{F}_{old}$ - columns of old variables that are both not involved and focused (${}^{\neg I}X^{F}_{old}$). Additionally, ${}^{I}A_{old}$ can be partitioned into ${}^{I}A^{U}_{old}$ - columns of old variables that are both involved and unfocused (${}^{I}X^{U}_{old}$), and ${}^{I}A^{F}_{old}$ - columns of old variables that are both involved and focused (${}^{I}X^{F}_{old}$) (see Table 3.2). The set of focused variables is then $X^F_{k+L} = \{{}^{\neg I}X^{F}_{old} \cup {}^{I}X^{F}_{old}\} \in \mathbb{R}^{n_F}$, containing both involved and not involved variables. We will denote $X^F_{k+L} \doteq X^F_k$ to remind us that focused set of variables is part of both X_{k+L} and X_k.

Likewise, the set of all remained, unfocused variables is $X^R_{k+L} \doteq \{{}^{\neg I}X^{U}_{old} \cup {}^{I}X^{U}_{old} \cup X_{new}\} \in \mathbb{R}^{n_R}$, containing all new variables and some of old ones (which can be involved or not involved), and providing A's partition $A_R = [{}^{\neg I}A^{U}_{old}, {}^{I}A^{U}_{old}, A_{new}]$. Moreover, for purpose of simplification of coming equations we'll denote set of old variables inside X^R_{k+L} by X^R_{old}, having that $X^R_{old} \doteq \{{}^{\neg I}X^{U}_{old} \cup {}^{I}X^{U}_{old}\}$, with appropriate Jacobian partition $A^R_{old} \doteq \left[{}^{\neg I}A^{U}_{old}, {}^{I}A^{U}_{old}\right]$.

Next, noticing that $X_k = \{X^F_k \cup X^R_{old}\}$ we can partition the prior information matrix Λ_k respectively

$$\Lambda_k = \begin{bmatrix} \Lambda^F_k & \Lambda^{F,R_{old}}_k \\ (\Lambda^{F,R_{old}}_k)^T & \Lambda^{R_{old}}_k \end{bmatrix}. \tag{C23}$$

Similarly, due to $X_{k+L} = \{X^F_k \cup X^R_{old} \cup X_{new}\}$ and $X^R_{k+L} \doteq \{X^R_{old} \cup X_{new}\}$ the posterior information matrix Λ_{k+L} can be respectively partitioned in next two forms:

$$\Lambda_{k+L} = \begin{bmatrix} \Lambda^F_{k+L} & \Lambda^{F,R_{old}}_{k+L} & \Lambda^{F,X_{new}}_{k+L} \\ (\Lambda^{F,R_{old}}_{k+L})^T & \Lambda^{R_{old}}_{k+L} & \Lambda^{R_{old},X_{new}}_{k+L} \\ (\Lambda^{F,X_{new}}_{k+L})^T & (\Lambda^{R_{old},X_{new}}_{k+L})^T & \Lambda^{X_{new}}_{k+L} \end{bmatrix} = \begin{bmatrix} \Lambda^F_{k+L} & \Lambda^{F,R}_{k+L} \\ (\Lambda^{F,R}_{k+L})^T & \Lambda^{R}_{k+L} \end{bmatrix} \tag{C24}$$

with

$$\Lambda^R_{k+L} = \begin{bmatrix} \Lambda^{R_{old}}_{k+L} & \Lambda^{R_{old},X_{new}}_{k+L} \\ (\Lambda^{R_{old},X_{new}}_{k+L})^T & \Lambda^{X_{new}}_{k+L} \end{bmatrix}. \tag{C25}$$

We can see from above partitions (C23)-(C25) that posterior information partition Λ^R_{k+L} of X^R_{k+L} is simply the augmentation of prior information partition $\Lambda^{R_{old}}_k$ and can be calculated as:

$$\Lambda^R_{k+L} = \Lambda^{Aug,R_{old}}_k + A^T_R A_R \tag{C26}$$

where $\Lambda^{Aug,R_{old}}_k$ can be constructed by first taking partition of prior information matrix Λ_k related to X^R_{old}, $\Lambda^{R_{old}}_k$, and augmenting it with n' zero rows and columns (see Figure 3.6), where n' is just number of newly introduced variables. The above equation has augmented determinant

form as defined in Section 3.3.1, and so the augmented determinant lemma can be applied also here. Using Eq. (3.15) we have:

$$\frac{\left|\Lambda^R_{k+L}\right|}{\left|\Lambda^{R_{old}}_k\right|} = \left|S\right| \cdot \left|A^T_{new} \cdot S^{-1} \cdot A_{new}\right| \tag{C27}$$

$$S = I_m + A^R_{old} \cdot (\Lambda^{R_{old}}_k)^{-1} \cdot (A^R_{old})^T \tag{C28}$$

Then by combining the Eq. (B15), Eq. (3.19) and the above equations, we can see that:

$$\frac{\left|\Sigma^{M,F}_{k+L}\right|}{\left|\Sigma^{M,F}_k\right|} = \frac{\left|\Lambda^R_{k+L}\right|}{\left|\Lambda_{k+L}\right|} \cdot \frac{\left|\Lambda_k\right|}{\left|\Lambda^{R_{old}}_k\right|} = \frac{\left|S\right| \cdot \left|A^T_{new} \cdot S^{-1} \cdot A_{new}\right|}{\left|C\right| \cdot \left|A^T_{new} \cdot C^{-1} \cdot A_{new}\right|} \tag{C29}$$

where C is defined in Eq. (3.21).

And apparently the IG of X^F_{k+L} can be calculated as:

$$J^F_{IG}(a) = \mathcal{H}(X^F_k) - \mathcal{H}(X^F_{k+L}) = \frac{1}{2}\ln\left|\Sigma^{M,F}_k\right| - \frac{1}{2}\ln\left|\Sigma^{M,F}_{k+L}\right| = \\ \frac{1}{2}(\ln\left|C\right| + \ln\left|A^T_{new} \cdot C^{-1} \cdot A_{new}\right| - \ln\left|S\right| - \ln\left|A^T_{new} \cdot S^{-1} \cdot A_{new}\right|), \tag{C30}$$

Next, S term can be further reduced. It is clear that $(\Lambda^{R_{old}}_k)^{-1} = \Sigma^{R_{old}|F}_k$, or namely the prior conditional covariance matrix of X^R_{old} conditioned on X^F_k. Moreover, due to sparsity of A^R_{old} (its sub-block ${}^{\neg I}A^U_{old}$ contains only zeros) we will actually need only entries of matrix $\Sigma^{R_{old}|F}_k$ that belong to variables involved in new terms of Eq. (2.6) (see Figure 3.6) and can conclude that:

$$S = I_m + A^R_{old} \cdot \Sigma^{R_{old}|F}_k \cdot (A^R_{old})^T = I_m + {}^IA^U_{old} \cdot \Sigma^{{}^IX^U_{old}|F}_k \cdot ({}^IA^U_{old})^T \tag{C31}$$

■

Note that the variables inside information matrices do not have to be ordered in any particular way, and that the provided above proof is correct for any ordering whatsoever.

9.6 Proof of Lemma 3.3.6

For A's structure given in Eq. (3.17), term S from Eq. (C31) will be:

$$S = I_m + A^R_{old} \cdot \Sigma^{R_{old}|F}_k \cdot (A^R_{old})^T = I_m + \begin{pmatrix} B^R_{old} \\ 0 \end{pmatrix} \cdot \Sigma^{R_{old}|F}_k \cdot \begin{pmatrix} (B^R_{old})^T & 0 \end{pmatrix} = \begin{pmatrix} S_1 & 0 \\ 0 & I_{m_{new}} \end{pmatrix} \tag{C32}$$

where $S_1 = I_{m_{conn}} + B^R_{old} \cdot \Sigma^{R_{old}|F}_k \cdot (B^R_{old})^T$, $m_{conn} = \mathcal{M}(\mathcal{F}^{conn}(a))$ and $m_{new} = \mathcal{M}(\mathcal{F}^{new}(a))$.

Then we can conclude that:

$$\left|S\right| = \left|S_1\right| \tag{C33}$$

and that:

$$S^{-1} = \begin{pmatrix} S_1^{-1} & 0 \\ 0 & I_{m_{new}} \end{pmatrix}, \tag{C34}$$

and similarly to Eq. (C31) (see also Figure 3.6) we have that:

$$S_1 = I_{m_{conn}} + B^R_{old} \cdot \Sigma_k^{R_{old}|F} \cdot (B^R_{old})^T = I_m + {}^I B^U_{old} \cdot \Sigma_k^{{}^I X^U_{old}|F} \cdot ({}^I B^U_{old})^T \tag{C35}$$

Next, term $A^T_{new} \cdot S^{-1} \cdot A_{new}$ from Eq. (C30), similarly to Eq. (A12), will be:

$$A^T_{new} \cdot S^{-1} \cdot A_{new} = B^T_{new} \cdot S_1^{-1} \cdot B_{new} + D^T_{new} \cdot D_{new}, \tag{C36}$$

with S_1 defined in Eq. (C35).

Then, by applying equations (B20), (C33), (C36) and notion $\left|C\right| = \left|C_1\right|$, the IG of $X^F_{k+L} \subseteq X_{old}$ from Eq. (C30) can be calculated as:

$$\begin{aligned} J^F_{IG}(a) = \frac{1}{2}(\ln\left|C_1\right| + \ln\left|B^T_{new} \cdot C_1^{-1} \cdot B_{new} + D^T_{new} \cdot D_{new}\right| - \\ \ln\left|S_1\right| - \ln\left|B^T_{new} \cdot S_1^{-1} \cdot B_{new} + D^T_{new} \cdot D_{new}\right|) = \\ \frac{1}{2}(\ln\left|C_1\right| + \ln\left|B^T_{new} \cdot C_1^{-1} \cdot B_{new} + \Lambda_a\right| - \ln\left|S_1\right| - \ln\left|B^T_{new} \cdot S_1^{-1} \cdot B_{new} + \Lambda_a\right|), \end{aligned} \tag{C37}$$

where C_1 is defined in Eq. (3.24), and where $\Lambda_a = D^T_{new} \cdot D_{new}$ is information matrix of action's factor graph $G(a)$.

■

Note that the variables inside information matrices do not have to be ordered in any particular way, and that the provided above proof is correct for any ordering whatsoever.

9.7 SLAM Solution - focus on last pose $X^F_{k+L} \equiv x_{k+L}$

For $X^F_{k+L} \equiv x_{k+L}$ the `focused` entropy objective in SLAM setting is given by Eq. (3.27). Here, we will exploit the inner structure of Jacobian partitions in SLAM scenario (see Eq. (4.14)-(4.18)) in order to provide solution tailored specifically to SLAM domain. It will provide an illustrated example of applying *rAMDL* to real problem.

From Eq. (4.18) we can see that B^U_{new} has next form:

$$B^U_{new} = \Psi^{-\frac{1}{2}}_{conn} \cdot \begin{pmatrix} \underline{(x_{k+1})} & \cdots & \underline{(x_{k+L-1})} \\ -I & \cdots & 0 \\ \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L-1}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L-1}} \end{pmatrix} = \Psi^{-\frac{1}{2}}_{conn} \cdot \begin{pmatrix} \mathbb{F}^U \\ \mathbb{H}^U_{X_{new}} \end{pmatrix},$$

$$\mathbb{F}^U \doteq \begin{pmatrix} -I & \cdots & 0 \end{pmatrix}, \quad \mathbb{H}^U_{X_{new}} \doteq \begin{pmatrix} \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L-1}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L-1}} \end{pmatrix} \tag{C38}$$

where $\mathbb{H}_i^{x_{k+l}} \doteq \nabla_{x_{k+l}} h_i$ is Jacobian of h_i with respect to x_{k+l}, and therefore is being non-zero only if factor's observation was taken from pose x_{k+l}. Note that in SLAM case the X^U_{new} (all new and `unfocused` variables) is $\{x_{k+1}, \ldots, x_{k+L-1}\}$.

Similarly to Eq. (4.22), the term $(B^U_{new})^T \cdot C_1^{-1} \cdot B^U_{new}$ from Eq. (3.27) can be calculated as:

$$(B^U_{new})^T \cdot C_1^{-1} \cdot B^U_{new} = \begin{pmatrix} (\mathbb{F}^U)^T & (\mathbb{H}^U_{X_{new}})^T \end{pmatrix} \cdot \Psi^{-\frac{1}{2}}_{conn} \cdot \Psi^{\frac{1}{2}}_{conn} \cdot C_2^{-1} \cdot \Psi^{\frac{1}{2}}_{conn} \cdot \Psi^{-\frac{1}{2}}_{conn} \cdot \begin{pmatrix} \mathbb{F}^U \\ \mathbb{H}^U_{X_{new}} \end{pmatrix} =$$

$$\begin{pmatrix} (\mathbb{F}^U)^T & (\mathbb{H}^U_{X_{new}})^T \end{pmatrix} \cdot C_2^{-1} \cdot \begin{pmatrix} \mathbb{F}^U \\ \mathbb{H}^U_{X_{new}} \end{pmatrix} = (\widetilde{B}^U_{new})^T \cdot C_2^{-1} \cdot \widetilde{B}^U_{new} \tag{C39}$$

where $\widetilde{B}^U_{new}$:

$$\widetilde{B}^U_{new} \doteq \begin{pmatrix} \underline{(x_{k+1})} & \cdots & \underline{(x_{k+L-1})} \\ -I & \cdots & 0 \\ \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L-1}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L-1}} \end{pmatrix} = \begin{pmatrix} \mathbb{F}^U \\ \mathbb{H}^U_{X_{new}} \end{pmatrix} \tag{C40}$$

contains the Jacobian entries of B^U_{new} not weighted by factors' noise Ψ_{conn}.

Additionally, from Eq. (4.14) we can derive structure of D^U_{new} which is also used in Eq. (3.27):

$$D^U_{new} = \Psi^{-\frac{1}{2}}_{new} \cdot \begin{array}{c} \begin{array}{ccccccc} (\underline{x_{k+1}}) & (\underline{x_{k+2}}) & (\underline{x_{k+3}}) & (\underline{x_{k+4}}) & \cdots & (\underline{x_{k+L-2}}) & (\underline{x_{k+L-1}}) \end{array} \\ \begin{pmatrix} \mathbb{F}_{k+1} & -I & 0 & 0 & \cdots & 0 & 0 \\ 0 & \mathbb{F}_{k+2} & -I & 0 & \cdots & 0 & 0 \\ 0 & 0 & \mathbb{F}_{k+3} & -I & \cdots & 0 & 0 \\ 0 & 0 & 0 & \mathbb{F}_{k+4} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -I & 0 \\ 0 & 0 & 0 & 0 & \cdots & \mathbb{F}_{k+L-2} & -I \\ 0 & 0 & 0 & 0 & \cdots & 0 & \mathbb{F}_{k+L-1} \end{pmatrix} \end{array} \doteq \Psi^{-\frac{1}{2}}_{new} \cdot \widetilde{D}^U_{new} \tag{C41}$$

that due to its sparsity will allow fast calculation of $\Lambda^{U|F}_a$:

$$\Lambda^{U|F}_a = (D^U_{new})^T \cdot D^U_{new} = (\widetilde{D}^U_{new})^T \cdot \Psi^{-1}_{new} \cdot \widetilde{D}^U_{new}. \tag{C42}$$

Finally, placing all derived notations into Eq. (3.27), we will get to the SLAM-specific solution for entropy of robot's last pose:

$$J^F_{\mathcal{H}}(a) = \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln \left| (\widetilde{B}^U_{new})^T \cdot C_2^{-1} \cdot \widetilde{B}^U_{new} + (\widetilde{D}^U_{new})^T \cdot \Psi^{-1}_{new} \cdot \widetilde{D}^U_{new} \right| - \frac{1}{2} \ln \left| \widetilde{B}^T_{new} \cdot C_2^{-1} \cdot \widetilde{B}_{new} + \widetilde{D}^T_{new} \cdot \Psi^{-1}_{new} \cdot \widetilde{D}_{new} \right|, \tag{C43}$$

where C_2 is defined in Eq. (4.20). ■

Note that the variables inside information matrices do not have to be ordered in any particular way, and that the provided above proof is correct for any ordering whatsoever.

9.8 SLAM Solution - focus on mapped landmarks $X^F_{k+L} \equiv L_k$

The `focused` IG of $X^F_k \equiv L_k$ in SLAM setting is given by Eq. (3.30). Here, we will exploit the inner structure of Jacobian partitions in SLAM scenario (see Eq. (4.14)-(4.18)) in order to provide solution tailored specifically to SLAM domain. It will provide an illustrated example of applying *rAMDL* to real problem.

First, note that all old *involved* and `unfocused` variables ${}^I X^U_{old}$ contain only the current robot's pose x_k. Thus, from Eq. (4.17) we can see that relevant partition of Jacobian B, the ${}^I B^U_{old}$

used in Eq. (3.31), has the following inner structure:

$$ {}^{I}B^{U}_{old} = \Psi^{-\frac{1}{2}}_{conn} \cdot \begin{pmatrix} \mathbb{F}_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{C44} $$

Using the above identity, the matrix S_1 from Eq. (3.31) can also be reduced to next form:

$$ \begin{aligned} S_1 = I_{m_{conn}} + {}^{I}B^{U}_{old} \cdot \Sigma_k^{{}^{I}X^{U}_{old}|F} \cdot ({}^{I}B^{U}_{old})^T = \\ I_{m_{conn}} + \Psi^{-\frac{1}{2}}_{conn} \cdot \begin{pmatrix} \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T & 0 \\ 0 & 0 \end{pmatrix} \cdot \Psi^{-\frac{1}{2}}_{conn} = \\ \Psi^{-\frac{1}{2}}_{conn} \cdot \left[\Psi_{conn} + \begin{pmatrix} \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T & 0 \\ 0 & 0 \end{pmatrix} \right] \cdot \Psi^{-\frac{1}{2}}_{conn} \doteq \\ \Psi^{-\frac{1}{2}}_{conn} \cdot S_2 \cdot \Psi^{-\frac{1}{2}}_{conn}, \end{aligned} \tag{C45} $$

$$ S_2 \doteq \Psi_{conn} + \begin{pmatrix} \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \Sigma_{\omega,k} + \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T & 0 \\ 0 & \Psi_{obs} \end{pmatrix} \tag{C46} $$

where is $\Sigma_{\omega,k}$ is noise matrix from motion model (Eq. (4.8)), and matrix Ψ_{obs} is block-diagonal, combining all noise matrices of $\mathcal{F}^{obs}(a)$ factors:

$$ \Psi_{conn} = \begin{pmatrix} \Sigma_{\omega,k} & 0 \\ 0 & \Psi_{obs} \end{pmatrix}. \tag{C47} $$

Further, let us define matrix S_3:

$$ S_3 \doteq \Sigma_{\omega,k} + \mathbb{F}_k \cdot \Sigma_k^{M,x_k} \cdot \mathbb{F}_k^T \tag{C48} $$

Now we can see that S_1's determinant and inverse can be calculated through:

$$ \left|S_1\right| = \frac{\left|S_2\right|}{\left|\Psi_{conn}\right|} = \frac{\left|S_3\right| \cdot \left|\Psi_{obs}\right|}{\left|\Psi_{conn}\right|} = \frac{\left|S_3\right|}{\left|\Sigma_{\omega,k}\right|}, \tag{C49} $$

$$ S_1^{-1} = \Psi^{\frac{1}{2}}_{conn} \cdot S_2^{-1} \cdot \Psi^{\frac{1}{2}}_{conn} = \Psi^{\frac{1}{2}}_{conn} \cdot \begin{pmatrix} S_3^{-1} & 0 \\ 0 & \Psi^{-1}_{obs} \end{pmatrix} \cdot \Psi^{\frac{1}{2}}_{conn}. \tag{C50} $$

Similarly to Eq. (4.22), the term $B_{new}^T \cdot S_1^{-1} \cdot B_{new}$ from Eq. (3.30) can be calculated as:

$$
\begin{aligned}
B_{new}^T \cdot S_1^{-1} \cdot B_{new} = \\
&\begin{pmatrix} \mathbb{F}^T & (\mathbb{H}^{X_{new}})^T \end{pmatrix} \cdot \Psi_{conn}^{-\frac{1}{2}} \cdot \Psi_{conn}^{\frac{1}{2}} \cdot S_2^{-1} \cdot \Psi_{conn}^{\frac{1}{2}} \cdot \Psi_{conn}^{-\frac{1}{2}} \cdot \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix} = \\
&\begin{pmatrix} \mathbb{F}^T & (\mathbb{H}^{X_{new}})^T \end{pmatrix} \cdot S_2^{-1} \cdot \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix} = \\
&\begin{pmatrix} \mathbb{F}^T & (\mathbb{H}^{X_{new}})^T \end{pmatrix} \cdot \begin{pmatrix} S_3^{-1} & 0 \\ 0 & \Psi_{obs}^{-1} \end{pmatrix} \cdot \begin{pmatrix} \mathbb{F} \\ \mathbb{H}^{X_{new}} \end{pmatrix} = \\
&\mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F} + (\mathbb{H}^{X_{new}})^T \cdot \Psi_{obs}^{-1} \cdot \mathbb{H}^{X_{new}}
\end{aligned} \tag{C51}
$$

where $\mathbb{F}$ and $\mathbb{H}^{X_{new}}$ are defined in Eq. (4.18) as

$$
\mathbb{F} \doteq \begin{pmatrix} -I & \cdots & 0 \end{pmatrix}, \quad \mathbb{H}^{X_{new}} \doteq \begin{pmatrix} \mathbb{H}_1^{x_{k+1}} & \cdots & \mathbb{H}_1^{x_{k+L}} \\ \vdots & \ddots & \vdots \\ \mathbb{H}_{n_o}^{x_{k+1}} & \cdots & \mathbb{H}_{n_o}^{x_{k+L}} \end{pmatrix}. \tag{C52}
$$

Thus we can see that $\mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F}$ from Eq. (C51) is $L \cdot n_p \times L \cdot n_p$ matrix (n_p is the robot pose's dimension and L is the horizon length) which has non-zero entries only at its $n_p \times n_p$ top left corner:

$$
\mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F} = \begin{pmatrix} S_3^{-1} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{C53}
$$

Finally, placing all derived notations into Eq. (3.30), we will get to the SLAM-specific solution for IG of already mapped landmarks L_k:

$$
\begin{aligned}
J_{IG}^F(a) = \frac{1}{2}(\ln\left|C_2\right| - \ln\left|\Psi_{conn}\right| + \ln\left|\widetilde{B}_{new}^T \cdot C_2^{-1} \cdot \widetilde{B}_{new} + \widetilde{D}_{new}^T \cdot \Psi_{new}^{-1} \cdot \widetilde{D}_{new}\right| - \\
- \ln\left|S_3\right| + \ln\left|\Sigma_{\omega,k}\right| - \ln\left|\mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F} + (\mathbb{H}^{X_{new}})^T \cdot \Psi_{obs}^{-1} \cdot \mathbb{H}^{X_{new}} + \widetilde{D}_{new}^T \cdot \Psi_{new}^{-1} \cdot \widetilde{D}_{new}\right|) = \\
= \frac{1}{2}(\ln\left|C_2\right| + \ln\left|\widetilde{B}_{new}^T \cdot C_2^{-1} \cdot \widetilde{B}_{new} + \widetilde{D}_{new}^T \cdot \Psi_{new}^{-1} \cdot \widetilde{D}_{new}\right| - \\
- \ln\left|S_3\right| - \ln\left|\mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F} + (\mathbb{H}^{X_{new}})^T \cdot \Psi_{obs}^{-1} \cdot \mathbb{H}^{X_{new}} + \widetilde{D}_{new}^T \cdot \Psi_{new}^{-1} \cdot \widetilde{D}_{new}\right| - \ln\left|\Psi_{obs}\right|)
\end{aligned} \tag{C54}
$$

where C_2 is defined in Eq. (4.20). Note that matrix S_3 will be the same for all candidates. Therefore, the terms S_3, $\ln\left|S_3\right|$ and $\mathbb{F}^T \cdot S_3^{-1} \cdot \mathbb{F}$ can be calculated only one time and shared between the candidates thereafter. Additionally, the terms C_2, $\widetilde{B}_{new}^T \cdot C_2^{-1} \cdot \widetilde{B}_{new}$, $\widetilde{D}_{new}^T \cdot \Psi_{new}^{-1} \cdot \widetilde{D}_{new}$ and $(\mathbb{H}^{X_{new}})^T \cdot \Psi_{obs}^{-1} \cdot \mathbb{H}^{X_{new}}$ can be calculated efficiently through sparse matrix operators since we know the exact inner structure of all involved matrix operands. The overall complexity of above SLAM solution is the same as in Eq. (3.30), $O(\mathcal{M}(\mathcal{F}^{conn}(a))^3 + n'^3)$. ■

Note that the variables inside information matrices do not have to be ordered in any particular

way, and that the provided above proof is correct for any ordering whatsoever.

9.9 Proof of Lemmas 3.6.1 and 3.6.2

Proof of Lemma 3.6.1

The basic definition of mutual information $I(a|b)$ is:

$$I(a|b) = \mathcal{H}(a) - \mathcal{H}(a|b) \tag{C55}$$

Using definition of entropy for Gaussian distribution from Eq. (2.15) we get to:

$$I(a|b) = \frac{1}{2} \ln \frac{\left|\Sigma_k^{M,a}\right|}{\left|\Sigma_k^{a|b}\right|} \tag{C56}$$

where $\Sigma_k^{M,a}$ is marginal covariance matrix of set a, and $\Sigma_k^{a|b}$ is conditional covariance matrix of set a, conditioned on set b.

Now, if current state vector contains only a and b, or in other words $X_k = \{a, b\}$, then $\left|\Sigma_k^{a|b}\right| = \frac{1}{\left|\Lambda_k^a\right|}$, with Λ_k^a being partition of information matrix Λ_k with respect to variables from a. Then by using Eq. (3.9) we will get:

$$I(a|b) = \frac{1}{2} \ln \frac{\left|\Lambda_k^a\right| \cdot \left|\Lambda_k^b\right|}{\left|\Lambda_k\right|} \tag{C57}$$

■

Proof of Lemma 3.6.2

Consider case when X_k contains additional variables except for subsets a and b. Let's join all other variables in subset r, with $X_k = \{a, b, r\}$ and where r is not empty. By applying Eq. (3.9) we can see that:

$$\left|\Sigma_k^{M,a}\right| = \frac{\left|\Sigma_k^{M,(a,b)}\right|}{\left|\Sigma_k^{a|b}\right|}, \quad \text{and therefore} \quad \left|\Sigma_k^{a|b}\right| = \frac{\left|\Sigma_k^{M,(a,b)}\right|}{\left|\Sigma_k^{M,b}\right|}, \tag{C58}$$

as also

$$\left|\Sigma_k^{M,a}\right| = \frac{\left|\Lambda_k^{(b,r))}\right|}{\left|\Lambda_k\right|} \tag{C59}$$

$$\left|\Sigma_k^{M,b}\right| = \frac{\left|\Lambda_k^{(a,r))}\right|}{\left|\Lambda_k\right|} \tag{C60}$$

$$\left|\Sigma_k^{M,(a,b)}\right| = \frac{\left|\Lambda_k^r\right|}{\left|\Lambda_k\right|} \tag{C61}$$

Then, after combining (C58), (C60) and (C61), another conclusion will be:

$$\left|\Sigma_k^{a|b}\right| = \frac{\left|\Lambda_k^r\right|}{\left|\Lambda_k^{(a,r)}\right|} \tag{C62}$$

And after combining (C56), (C59) and (C62), we get the final expression for information gain:

$$I(a|b) = \frac{1}{2} \ln \frac{\left|\Lambda_k^{(a,r)}\right| \cdot \left|\Lambda_k^{(b,r)}\right|}{\left|\Lambda_k\right| \cdot \left|\Lambda_k^r\right|} \tag{C63}$$

■

Bibliography

[1] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato. Firm: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements. *Intl. J. of Robotics Research*, 2014.

[2] Shi Bai, Jinkun Wang, Fanfei Chen, and Brendan Englot. Information-theoretic exploration with bayesian optimization. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[3] Zhaojun Bai, Gark Fahey, and Gene Golub. Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, 74(1):71–89, 1996.

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

[5] L. Carlone, A. Censi, and F. Dellaert. Selecting good measurements via l1 relaxation: A convex approach for robust estimation over graphs. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2667–2674, 2014.

[6] S. M. Chaves, J. M. Walls, E. Galceran, and R. M. Eustice. Risk aversion in belief-space planning under measurement acquisition uncertainty. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[7] M. Chli and A. J. Davison. Active matching for visual tracking. *Robotics and Autonomous Systems*, 57(12):1173–1187, December 2009.

[8] T.A. Davis, J.R. Gilbert, S.I. Larimore, and E.G. Ng. A column approximate minimum degree ordering algorithm. *ACM Trans. Math. Softw.*, 30(3):353–376, 2004.

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

[10] F. Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology, September 2012.

[11] G.H. Golub and R.J. Plemmons. Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition. *Linear Algebra and Its Applications*, 34:3–28, Dec 1980.

[12] J Guivant, J Nieto, and E Nebot. Victoria park dataset, 2012.

[13] David A Harville. Matrix algebra from a statistician's perspective. *Technometrics*, 40(2):164–164, 1998.

[14] G. Huang, M. Kaess, and J.J. Leonard. Consistent sparsification for graph optimization. In *Proc. of the European Conference on Mobile Robots (ECMR)*, pages 150 – 157, 2012.

[15] S. Huang, N. Kwok, G. Dissanayake, Q. Ha, and G. Fang. Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1091–1096, 2005.

[16] V. Ila, J. M. Porta, and J. Andrade-Cetto. Information-based compact Pose SLAM. *IEEE Trans. Robotics*, 26(1), 2010. In press.

[17] Viorela Ila, Lukas Polok, Marek Solony, Pavel Smrz, and Pavel Zemcik. Fast covariance recovery in incremental nonlinear least square solvers. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4636–4643. IEEE, 2015.

[18] V. Indelman. Towards cooperative multi-robot belief space planning in unknown environments. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, September 2015.

[19] V. Indelman. Towards information-theoretic decision making in a conservative information space. In *American Control Conference*, pages 2420–2426, July 2015.

[20] 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.

[21] 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.

[22] 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.

[23] M. Kaess and F. Dellaert. Covariance recovery from a square root information matrix for data association. *Robotics and Autonomous Systems*, 57(12):1198–1210, 2009.

[24] 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*, 31:217–236, Feb 2012.

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

[26] A. Kim and R. M. Eustice. Active visual slam for robotic area coverage: Theory and experiment. *Intl. J. of Robotics Research*, 2014.

[27] 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.

[28] 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.

[29] F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2), February 2001.

[30] D. Levine and J. P. How. Sensor selection in high-dimensional gaussian trees with nuisances. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2211–2219, 2013.

[31] Mladen Mazuran, Gian Diego Tipaldi, Luciano Spinello, and Wolfram Burgard. Nonlinear graph sparsification for slam. In *Robotics: Science and Systems (RSS)*, pages 1–8, 2014.

[32] Beipeng Mu, Ali-akbar Agha-mohammadi, Liam Paull, Mathew Graham, Jonathan How, and John Leonard. Two-stage focused inference for resource-constrained collision-free navigation. In *Robotics: Science and Systems (RSS)*, 2015.

[33] Diane Valerie Ouellette. Schur complements and statistics. *Linear Algebra and its Applications*, 36:187–295, 1981.

[34] S. Patil, G. Kahn, M. Laskey, J. Schulman, K. Goldberg, and P. Abbeel. Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation. In *Intl. Workshop on the Algorithmic Foundations of Robotics*, 2014.

[35] Liam Paull, Guoquan Huang, and John J Leonard. A unified resource-constrained framework for graph slam. In *ICRA*, pages 1–8, 2016.

[36] J. Pineau, G. J. Gordon, and S. Thrun. Anytime point-based approximations for large pomdps. *J. of Artificial Intelligence Research*, 27:335–380, 2006.

[37] 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.

[38] S. Prentice and N. Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *Intl. J. of Robotics Research*, 2009.

[39] 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.

[40] R. Valencia, M. Morta, J. Andrade-Cetto, and J.M. Porta. Planning reliable paths with pose SLAM. *IEEE Trans. Robotics*, 2013.

[41] 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.

[42] J. Vial, H. Durrant-Whyte, and T. Bailey. Conservative sparsification for efficient and consistent approximate estimation. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 886–893. IEEE, 2011.

[43] J. M. Walls, S. M. Chaves, E. Galceran, and R. M. Eustice. Belief space planning for underwater cooperative localization. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[44] Zhengyuan Zhu and Michael L Stein. Spatial sampling design for prediction with estimated parameters. *Journal of Agricultural, Biological, and Environmental Statistics*, 11(1):24–44, 2006.

[45] Dale L Zimmerman. Optimal network design for spatial prediction, covariance parameter estimation, and empirical prediction. *Environmetrics*, 17(6):635–652, 2006.

פר פעולה אינו נדרש יותר. במקום זה, אנחנו משתמשים במשפט הדטרמיננטה (Lemma, MDL Matrix Determinant) ומראים שבשביל חישוב תוספת אינפורמציה של פעולה מספיק לדעת חלקים שונים של מטריצת Jacobian של פעולה ואיברים ספציפיים ממטריצת הקוואריאנס האפריורית (a priori covariance matrix). במקרה ומטריצת הקוואריאנס אינה זמינה, ניתן לחשב את האיברים הספציפיים שלה באזרת מטריצת האינפורמציה (או מטריצת שורש שלה). אנחנו מבצעים חישוב חד־פעמי של איברי הקוואריאנס האפריורית שתלוי במימד המצב (n), ואחריו מחשבים את תוספת האינפורמציה של כל פעולה עם סיבוכיות חישובית שבכלל לא תלויה ב־n. לצורך פיתוח השיטה שלנו למקרה שבו מימד וקטור מצב גודל, אנחנו מפתחים משפט מתמטי חדש לחישוב דטרמיננטה של סכום מטריצות עם הגדלת מימד המטריצה (Augmented Matrix Determinant Lemma, AMDL), ובנוסף מראים שאפשר לשתף חישובים בזמן חישוב האינפורמציה של כל אחת מפעולות המועמדים. ה־AMDL ושיתוף החישובים הם הרכיבים העיקריים של השיטה שלנו. בשילוב עם ייצוג גרפי של הבעיה, הם מסתכמים בשיטה שלנו שפותרת את כל תתי־המקרים השונים של תכנון בצורה כללית ויעילה חישובית, שמתחשבת במקורות אי וודאות וסטוכסטיות שונים.

אנחנו בוחנים את השיטה שלנו בכל המקרים האפשריים - עם מימד וקטור מצב קבוע או גודל, ועם תכנון לא "מרוכז" ותכנון "מרוכז", ומשווים אותה לשיטות המקובלות בספרות בשנים האחרונות. בשביל המטרה הזו, אנחנו משתמשים גם בסימולציות וירטואליות וגם בסצנות מהעולם האמיתי, ופותרים בעיית תכנון אינפורמטיבי באפליקציות של ניווט אוטונומי בסביבה שאינה ידועה מראש, פריסה יעילה של חיישנים ובחירת מדידות אינפורמטיביות. אנחנו מראים שהשיטה שלנו דורשת משמעותית פחות זמן ריצה ללא פגיעה כלשהי באיכות התכנון. למשל, בבעית פריסה יעילה של חיישנים השיטה שלנו מבצעת תכנון לא "מרוכז" תוך 5 שניות לעומת שיטות אחרות שעושות אותו תכנון תוך 400 שניות. בנוסף, בבעיית ניווט אוטונומי בסביבה שאינה ידועה מראש, בשלבים סופיים של הסימולציה השיטות הסטנדרטיות זקוקות ל־10 שניות בשביל לבצע את התכנון ה־"מרוכז" כאשר השיטה שלנו פותרת את אותה בעיה תוך פחות משניה אחת.

תקציר

תכנון וקבלת החלטות תחת אי ודאות הינן בעיות יסוד בתחום של בינה מלאכותית, רובוטיקה והניווט האוטונומי. בבעיות אלו, וקטור הנעלמים המייצג את מצב המערכת אינו ידוע בודאות, בגלל מדידות סטוכסטיות וחישה לא מושלמת. במקרים אלו, וקטור הנעלמים, המכיל למשל מיקומים של רובוט ושל אובייקטים בעולם (landmarks), ניתן לייצוג על ידי פילוג (distribution) המוגדר במרחב המצב. הפילוג ידוע גם בשם אמונה (belief) ובעיית תכנון מתבצעת במרחב הסתברותי (belief space planning, BSP) בין אמונות שונות המתקבלות עבור פעולות אפשריות שונות. מודל זה מאפשר פתרון של בעיות שונות, כמו למשל פריסה יעילה של חיישנים, תכנון אינפורמטיבי, מיפוי ואיכון בו זמני אקטיבי (Active SLAM), וניווט אוטונומי בסביבה הלא ידועה מראש.

תכנון אינפורמטיבי במרחב אמונה הוא תת־בעיה של BSP היכן שהמטרה היא למצוא פעולה עם אי־וודאות פוסטריורית מינימלית של האמונה. התכנון הזה יכול לשמש אותנו למשל בשביל למצוא מסלול למטרה עם מדידות הכי פחות רועשות ולשערך את המפה הלא ידועה מראש בצורה הכי מדוייקת. פתרון בעיה זו כרוך בדרך כלל בחישוב אנטרופיה (entropy) של אמונה והינו מאוד יקר חישובית. בפרט, אלגוריתמים מהשנים האחרונות לפתרון תכנון אינפורמטיבי מחשבים אמונה פוסטריורית לכל פעולה נבחנת בעזרת עדכון מטריצת האינפורמציה או מטריצת הקוואריאנס (covariance), ומחשבים דטרמיננטה של מטריצה זו אשר נדרשת לחישוב האנטרופיה. סיבוכיות של חישוב כזה של דטרמיננטה הוא $O(n^3)$, איפה שה־n הוא מימד של וקטור נעלמים מערכתיים ובדרך כלל מהווה מספר גדול מאוד (> 10000). בעקבות זה, הסיבוכיות של האלגוריטמים הרגילים היא $O(n^3)$ פר מועמד, שזה איטי מאוד, מה שמקשה על השימוש בתכנון אינפורמטיבי לאפליקציות של זמן אמיתי.

במחקר זה אנחנו מפתחים שיטה חדשה לפתרון של תכנון אינפורמטיבי במרחב אמונה שהינה כללית ויעילה חישובית. אנחנו מראים שאפשר לחלק את התכנון האינפורמטיבי למספר של תתי־מקרים ומפתחים פיתרון לכל תת־מקרה כזה. קודם כל, לפי גודל וקטור המצב אפשר להפריד בין מקרה שבו וקטור הנעלמים נישאר זהה אחרי ביצוע פעולה ומקרה שבו נעלמים חדשים (כמו מיקומים חדשים של רובוט) מתווספים לוקטור אחרי שהפעולה התבצעה. אנחנו נראה ששני המקרים האלו הינם מתמטית שונים ונציג פיתרון יעיל לכל אחד מהם. בנוסף, אנחנו פותרים כאן גם תכנון לא "מרוכז" וגם תכנון "מרוכז", כאשר בסוג הראשון אנחנו רוצים להקטין אי וודאות של כל הנעלמים בתוך וקטור המצב, ובשני – רק אי וודאות של כמה נעלמים בודדים.

בשונה משיטות אחרות, השיטה שלנו אינה דורשת חישוב של אמונות פוסטריוריות וגם לא זקוקה לדטרמינטות של מטריצות ענקיות, ובעקבות זה משיגה שיפור משמעותי בזמן ריצה. אנחנו מבטאים את בעיית התכנון בעזרת המודלים הגרפיים ומראים שבייצוג כזה הבעיה היא הרבה יותר פשוטה ואינטואיטיבית בייחס לפורמליזציה הרגילה. בנוסף, אנחנו מוכיחים שחישוב אמונה פוסטריורי

המחקר נעשה בהנחיית פרופסור ואדים אינדלמן בפקולטה להנדסת אוירונוטיקה וחלל

תודות

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

תכנון יעיל במרחב הסתברותי רב ממדי תוך שימוש בחישובים קודמים וניצול דלילות

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים בתוכנית מערכות אוטונומיות

דמיטרי קופיטקוב

הוגש לסנט הטכניון — מכון טכנולוגי לישראל
אדר תשע"ז חיפה מרץ 2017

תכנון יעיל במרחב הסתברותי רב ממדי תוך שימוש בחישובים קודמים וניצול דלילות

דמיטרי קופיטקוב