Computationally Efficient Belief Space Planning via Augmented Matrix Determinant Lemma and Reuse of Calculations

Dmitry Kopitkov and Vadim Indelman

***Abstract*—We develop a computationally efficient approach for evaluating the information theoretic term within belief space planning (BSP) considering both unfocused and focused problem settings, where uncertainty reduction of the entire system or only of chosen variables is of interest, respectively. State-of-the-art approaches typically calculate, for each candidate action, the posterior information (or covariance) matrix and its determinant (required for entropy). In contrast, our approach reduces run-time complexity by avoiding these calculations, 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 reused when evaluating impact of different candidate actions. These two key ingredients result in a computationally efficient 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 the problem of autonomous navigation in unknown environments.**

***Index Terms*—SLAM, AI reasoning methods, learning and adaptive systems, optimization and optimal control.**

I. Introduction

PLANNING under uncertainty is a fundamental problem in robotics and is required in numerous applications such as autonomous driving, surveillance, active perception and active SLAM, where machines need to autonomously determine next actions to reliably realize a certain objective or task in a best way. 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) approaches reason how this distribution (the *belief*) evolves as a result of candidate actions and future expected observations.

The BSP problem is an instantiation of a partially observable Markov decision process (POMDP), with the latter being computationally intractable [1] 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. 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 [2], simulation based [3] and sampling based approaches [4], [5]. On the other hand, approaches that avoid discretization are often termed direct trajectory optimization methods (e.g. [6]–[9]); these approaches typically calculate from a given nominal solution a locally-optimal one.

The user-specified objective function includes different terms that vary from one application to another. These terms, however, typically include an information-theoretic term that quantifies uncertainty reduction (or information gain). Evaluating this term first involves belief propagation given the candidate action(s), and then calculating entropy or information gain of that *belief* [6], [10], [11].

For Gaussian distributions, these calculations involve computing determinant of a posteriori information (or covariance) matrices, which has $O(n^3)$ complexity, with n denoting state dimension [12]–[14]. In applications that involve high-dimensional state spaces, such as sensor deployment and SLAM, these calculations can become computationally expensive, making on-line decision making a challenge. Moreover, state of the art approaches typically perform these calculations from scratch for *each* candidate action.

Recently we developed a computationally efficient approach for decision making under uncertainty by applying matrix determinant lemma and reusing calculation between all candidates [13]. The developed algorithm gave improved per-candidate runtime performance which turned to be independent of state dimensions, thus significantly reducing running time. Yet, that approach considered state dimensionality to be fixed and thus can be applied only to a limited type of problems (e.g. sensor deployment).

In contrast, in this paper we focus on belief space planning problems that involve state augmentation within future beliefs due to the introduction of new variables (e.g. future robot poses). Examples include autonomous navigation in unknown environments and active SLAM. As we discuss in the sequel,

Manuscript received August 25, 2016; revised November 16, 2016; accepted December 19, 2016. Date of publication December 28, 2016; date of current version January 16, 2017. This paper was recommended for publication by Associate Editor J.-L. Blanco and Editor C. Stachniss upon evaluation of the reviewers' comments. This work was supported by the Israel Science Foundation.

D. Kopitkov is with the Technion Autonomous Systems Program, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: dimkak@tx.technion.ac.il).

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

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2016.2645894

our previous approach [13] is not directly applicable to this type of problems.

In this paper we provide the essential extensions necessary to apply similar in nature ideas to [13], to the realm of belief space planning in high-dimensional state spaces. Like in [13] the *key idea* is to use the (augmented) matrix determinant lemma to calculate action impact with complexity *independent* of state dimensionality n, while *re-using* calculations between impact evaluation for different candidate actions. Unlike our previous approach [13], the solution provided herein supports introduction of new variables within future beliefs, as well as general observation and motion models in a non-myopic problem setting.

In particular, we consider both unfocusedBSP and focusedBSP problems. In the former case, one is interested in reducing the uncertainty of the entire joint state, while focusedBSP approaches seek to reduce uncertainty of only predetermined subset of variables. Interestingly, the two cases (unfocused and focusedBSP) can have significantly different optimal actions, with an optimal solution for the unfocused case potentially yielding bad performance for the focused case, and vice versa [15].

Previously, matrix determinant lemma and calculation re-use were also applied to improve complexity of informative measurement selection specifically in Pose-SLAM [16], yet it was limited to unfocused setting with not-augmenting myopic candidate actions.

Calculating the posterior information matrix in both cases 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 Eq. (4)). Unfortunately, the matrix determinant lemma is not applicable to the mentioned augmented prior information matrix since the latter is singular (even though the posterior information matrix is a full rank matrix). 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 belief space planning approach, considering both unfocused and focused cases.

While the set of focused variables can be very small, exact standard approaches compute the marginal posterior covariance (information) matrix, for each action, which involves a computationally expensive Schur complement operation [17]. In contrast, we develop a new method to calculate posterior entropy of focused variables in BSP setting which does not involve computation of a posteriori covariance (information) matrices. We consider two instantiations of the focusedBSP problem, that differ in the identity of the focused variables (see Fig. 1). In combination with our AMDL and *re-use* calculations concepts, our approach for focusedBSP is significantly faster, while being exact, compared to the state of the art.

To summarize, our main contributions in this paper are as follows: (a) 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 (b) we develop an approach for a nonmyopic focused and unfocused belief space planning in high-dimensional state spaces that uses the augmented matrix determinant lemma to avoid calculating determinants

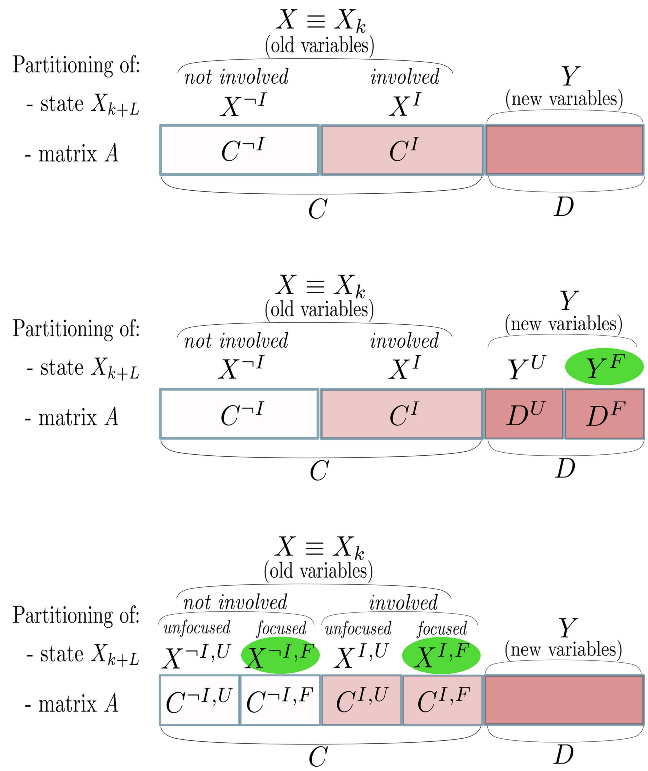

Fig. 1. Partitions of Jacobian $A \in \mathbb{R}^{m \times N} = [C \in \mathbb{R}^{m \times n}, D \in \mathbb{R}^{m \times n'}]$ and state vector X_{k+L}. Note: the shown variable ordering is only for illustration, while the developed approach supports *any* arbitrary variable ordering. Also note that all white $C^{\neg I}$ blocks (belonging to *not involved* variables) consist of only zeros. (a) Unfocused scenario. (b) Focused ($X^F_{k+L} \subseteq X$) scenario. (c) Focused ($X^F_{k+L} \subseteq Y$) scenario.

of large matrices, with per-candidate complexity independent of state dimension; (c) we show how calculations can be *re-used* when evaluating impacts of different candidate actions; (d) we integrate calculations *re-use* concept and AMDL into general and highly efficient BSP solver, naming this approach *rAMDL*. Due to space limitation, this paper is accompanied with supplementary material [18].

II. Notations and Problem Definition

Consider a high-dimensional problem-specific state vector $X_k \in \mathbb{R}^n$ at time t_k and observations $Z_{0:k}$, all observations acquired up to t_k. Each measurement in $Z_{0:k}$ can be represented through specific observation model and provides information about real value of subset of X_k. Moreover, for better intuition we will consider SLAM as main context example, though the presented in this paper BSP solution can be applied to any general estimation problem (with only requirement for measurements to be conditionally independent with Gaussian noise). As such the X_k represents robot poses per each time step and environment-related variables (f.e. mapped landmarks). Further, we model robot motion dynamics and sensor observations through:

$$x_{i+1} = f(x_i, u_i) + \omega_i \ \ , \ \ Z_i = h(X^o_i) + \upsilon_i \tag{1}$$

where u_i is control at time t_i, $X^o_i \subseteq X_i$ indicates the involved subset of states in the observation function $h(\cdot)$ (f.e. robot and landmark poses for projection measurement), and the motion and measurement noises are Gaussian and thus denote $\omega_i \sim \mathcal{N}(0, \Sigma_{\omega,i})$ and $\upsilon_i \sim \mathcal{N}(0, \Sigma_{\upsilon,i})$ with corresponding covariance matrices $\Sigma_{\omega,i}$ and $\Sigma_{\upsilon,i}$.

The joint pdf of state vector can be written then as

$$\mathbb{P}(X_k|Z_{0:k}, u_{0:k-1}) = \eta \mathbb{P}(x_0) \prod_{i=1}^{k} \mathbb{P}(x_i|x_{i-1}, u_{i-1}) \mathbb{P}(Z_i|X_i^o) \tag{2}$$

where η is a normalization constant and $p(x_0)$ is a prior on the first pose. The motion and observation models $\mathbb{P}(x_i|x_{i-1}, u_{i-1})$ and $\mathbb{P}(Z_i|X_i^o)$ are defined by Eq. (1).

Note that such formulation already implies that multiple observations $z_{i,j}$ can be taken at each time t_i, with $Z_i = \{z_{i,1}, ..., z_{i,n_i}\}$ and n_i being the number of such observations. Moreover, in context of SLAM we will have $\mathbb{P}(Z_i|X_i^o) = \prod_{j=1}^{n_i} \mathbb{P}(z_{i,j}|x_i, l_j)$ where robot at position x_i observes landmark l_j.

The maximum a posteriori (MAP) estimation of X_k can be efficiently calculated by exploiting sparsity and re-use of calculations (see e.g. [19]), providing estimation mean $\hat{X}_k$ and covariance Σ_k, and current state *belief* $b[X_k] \doteq \mathbb{P}(X_k|Z_{0:k}, u_{0:k-1}) = \mathcal{N}(\hat{X}_k, \Sigma_k)$.

In the context of BSP, we typically reason about the evolution of future beliefs $b[X_{k+l}]$ at different look-ahead steps l for different candidate sequence of actions. More specifically, let us focus on the belief $b[X_{k+L}] \doteq \mathbb{P}(X_{k+L}|Z_{0:k+L}, u_{0:k+L-1})$, where L is the planning horizon, and $u_{k+1:k+L-1}$ is the considered candidate action. Similar to Eq. (2), this belief can be explicitly written as

$$b[X_{k+L}] = \eta b[X_k] \prod_{l=k+1}^{k+L} \mathbb{P}(x_l|x_{l-1}, u_{l-1}) \mathbb{P}(Z_l|X_l^o) \tag{3}$$

Similar expressions can be also written for any other look ahead step l. Although the above belief is conditioned on future observations $Z_{k+1:k+L}$, their actual values are unknown. However, as will be seen below, evaluating the information-theoretic term only involves the Jacobians and not the actual values of $Z_{k+1:k+L}$. One can go further and account for the fact that measurements are noisy [6], [7], and incorporate reasoning if a future measurement will indeed be acquired [6], [8], [20]; however, this is outside the scope of this paper.

As seen in Eq. (3), the joint state X_{k+L} includes new variables (with respect to the current state X_k), i.e. future robot states. 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. (3).

Consequently, 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 $\Lambda_{k+L}^{\text{Aug}} \in \mathbb{R}^{N \times N}$, and thereafter adding to it new information (see e.g. [6]):

$$\Lambda_{k+L} = \Lambda_{k+L}^{\text{Aug}} + \sum_{l=k+1}^{k+L} F_l^T \cdot \Sigma_{\omega,l}^{-1} \cdot F_l + \sum_{l=k+1}^{k+L} H_l^T \cdot \Sigma_{\upsilon,l}^{-1} \cdot H_l \tag{4}$$

where $F_l \doteq \nabla_x f$ and $H_l \doteq \nabla_x h$ are augmented Jacobian matrices of all new factors in Eq. (3) (motion and observation terms all together), linearized about the current estimate of X_k and about initial values of newly introduced variables.

After stacking all new Jacobians in Eq. (4) together into one single matrix $\widetilde{A}$, and combining all noise matrices into block-diagonal Φ, we will get

$$\Lambda_{k+L} = \Lambda_{k+L}^{\text{Aug}} + \widetilde{A}^T \cdot \Phi^{-1} \cdot \widetilde{A} = \Lambda_{k+L}^{\text{Aug}} + A^T \cdot A \tag{5}$$

where $A \doteq \Phi^{-\frac{1}{2}} \cdot \widetilde{A}$ is $m \times N$ matrix that represents both Jacobians and noise covariances of all new factor terms in Eq. (3). The above equation can be considered as a single iteration of Gauss-Newton optimization and, similar to prior work [6], [7], [10], we assume it sufficiently captures the impact of action $u_{k:k+L-1}$. Under this assumption, the posterior information matrix Λ_{k+L} is independent of (unknown) future observations $Z_{k+1:k+L}$ [6].

Each block row of matrix A represents a single factor from new terms in Eq. (3) and has sparse structure. Only a limited number of its sub-blocks is non-zero, i.e. sub-blocks that correspond to the *involved* variables in the relevant factor.

For notational convenience, we define the set of candidate actions by $\mathcal{A} = \{a_1, a_2, .., a_k\}$ with appropriate Jacobian matrices $\Phi_A = \{A_1, A_2, .., A_k\}$. While the planning horizon is not explicitly shown, each $a \in \mathcal{A}$ can represent a sequence of controls, e.g. $a = u_{k:k+L-1}$ for L look ahead steps.

A general objective function in BSP can be written as [6]:

$$J(a) \doteq \mathop{\mathbb{E}}_{Z_{k+1:k+L}} \left\{ \sum_{l=0}^{L-1} c_l(b[X_{k+l}], u_{k+l}) + c_L(b[X_{k+L}]) \right\} \tag{6}$$

with L immediate cost functions c_l, for each look-ahead step, and one cost function for terminal future belief c_L. The c_L usually contains number of different terms related to aspects such as information measure of future belief, distance to goal and energy spent on control. Arguably the information term contains the heaviest calculations of J. Thus in this paper we will focus only on information-theoretic term of terminal belief 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 other, as will be shown in Section III. Therefore, we consider objective functions $J_{\mathcal{H}}(a) \doteq \mathcal{H}(b[X_{k+L}])$ and $J_{\text{IG}}(a) \doteq \mathcal{H}(b[X_k]) - \mathcal{H}(b[X_{k+L}])$, where the belief $b[X_{k+L}]$ is a function of the controls $a = u_{k:k+L-1}$, see Eq. (3). 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 $a^* = \arg\max_{a \in \mathcal{A}} J_{\text{IG}}(a)$ 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. $\mathcal{H}(b[X_{k+L}]) \equiv \mathcal{H}(\Lambda_{k+L})$ and the objective functions can be expressed as

$$J_{\mathcal{H}}(a) = \frac{N \cdot \gamma}{2} - \frac{1}{2} \ln \left| \Lambda_{k+L} \right| \tag{7}$$

$$J_{\text{IG}}(a) = \frac{n' \cdot \gamma}{2} + \frac{1}{2} \ln \frac{\left| \Lambda_{k+L} \right|}{\left| \Lambda_k \right|} \tag{8}$$

where $\gamma \doteq 1 + \ln(2\pi)$, and Λ_{k+L} can be calculated according to Eq. (5). Thus, evaluating J requires determinant calculation of an $N \times N$ matrix, which is in general $O(N^3)$, per candidate action $a \in \mathcal{A}$.

So far, the exposition referred to `unfocused BSP`, 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`.

`Focused BSP` is another important problem, where in contrast to the former case, only a subset of variables is of interest (see, e.g., [15], [17], [21]). For example one can look for action that reduces uncertainty of robot's final pose. The complexity of such problem is much higher and proposed techniques succeeded to solve it in $O(kn^3)$ [15], [21] with k being size of candidate set, and in $O(\tilde{n}^4)$ [17] with $\tilde{n}$ being size of involved clique.

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

$$J^F_{\mathcal{H}}(a) = \mathcal{H}(X^F_{k+L}) = \frac{n_F \cdot \gamma}{2} + \frac{1}{2} \ln \left| \Sigma^F_{k+L} \right| \tag{9}$$

where n_F is the dimension of X^F_{k+L}, and Σ^F_{k+L} is the posterior marginal covariance of X^F_{k+L}, 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. In the following section we develop a computationally more efficient approach that addresses both `unfocused` and `focused BSP` problems. As will be seen, this approach naturally supports non-myopic planning with arbitrary motion and observation models, and it is in particular attractive to belief space planning in high-dimensional state spaces.

III. Approach

Our approach, *rAMDL*, is based on several key ingredients to significantly reduce computational complexity of the BSP problem as defined in Section II. In Section III-A we extend the well-known matrix determinant lemma for the matrix augmentation case. We then discuss in Sections III-B and III-C how this extension can be used within `unfocused` and `focusedBSP`. In Section III-D 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.

A. Augmented Matrix Determinant Lemma (AMDL)

In order to simplify calculation of IG within BSP (Eq. (8)) one could resort to matrix determinant lemma and calculation re-use, similar to our previous work [13]. However, due to zero-padding, the information matrix $\Lambda^{\text{Aug}}_{k+L}$ is singular and thus the matrix determinant lemma, and calculation re-use, 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 $\Lambda^+ = \Lambda^{\text{Aug}} + A^T \cdot A$ (see also Eq. (5)), and dropping the time indices to avoid clutter, our objective is to express the determinant of Λ^+ in terms of Λ and $\Sigma = \Lambda^{-1}$.

TABLE I
Different Partitions of State Variables

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	subset of X_{k+L} with old variables, i.e. X_k
Y	subset of X_{k+L} with new variables
X^I	subset of X with variables *involved* in new terms in Eq. (3)
$X^{\neg I}$	subset of X with variables *not involved* in new terms in Eq. (3)
	`Focused BSP` ($X^F_{k+L} \subseteq Y$), Section III-C1
Y^F	subset of Y with `focused` variables
Y^U	subset of Y with `unfocused` variables
	`Focused BSP` ($X^F_{k+L} \subseteq X$), Section III-C2
$X^{I,F}$	subset of X^I with `focused` variables
$X^{I,U}$	subset of X^I with `unfocused` variables
$X^{\neg I,F}$	subset of $X^{\neg I}$ with `focused` variables
$X^{\neg I,U}$	subset of $X^{\neg I}$ with `unfocused` variables

Main Notations: We Use F and U for `focused` and `unfocused` Variables, Respectively; I and $\neg I$ for *involved* and *not involved* Variables Respectively; X (Without Subscript) for Old Variables and Y for New Variables.

Lemma 1: The ratio of determinants of Λ^+ and Λ can be calculated through:

$$\frac{\left|\Lambda^+\right|}{\left|\Lambda\right|} = \left|\Delta\right| \cdot \left| D^T \cdot \Delta^{-1} \cdot D \right| \tag{10}$$

with $\Delta \doteq I_m + C \cdot \Sigma \cdot C^T$, where the matrices $C \in \mathbb{R}^{m \times n}$ and $D \in \mathbb{R}^{m \times n'}$ are constructed from A by retrieving columns of only old n variables (denoted as X) and only new n' variables (denoted as Y), respectively (see Fig. 1(a)).

The proof of Lemma 1 is given in Appendix A in [18].

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

B. Unfocused BSP Through IG and Calculation Re-Use

Here we show how the augmented matrix determinant lemma from Section III-A can be used to efficiently calculate the `unfocused` IG as defined in Eq. (8).

First we introduce different partitions of the joint state X_{k+L}, and the corresponding sub-matrices in the Jacobian matrix A from Eq. (5) (see Table I and Fig. 1). Recall definitions of Y and X (see Section III-A) and let X^I and $X^{\neg I}$ denote, respectively, the involved and the uninvolved state variables in the new terms in Eq. (3). We represent by C^I and $C^{\neg I}$ the columns of matrix A that correspond to the state variables X^I and $X^{\neg I}$, respectively (see Fig. 1(a)). Note, $C^{\neg I} \equiv 0$.

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

$$\frac{\left|\Lambda_{k+L}\right|}{\left|\Lambda_k\right|} = \left|P\right| \cdot \left| D^T \cdot P^{-1} \cdot D \right| \tag{11}$$

where $P \doteq I_m + C \cdot \Sigma_k \cdot C^T$.

Consequently, the IG objective (8) can be re-written as

$$J_{\mathrm{IG}}(a) = \frac{n' \cdot \gamma}{2} + \frac{1}{2}\ln\left|P\right| + \frac{1}{2}\ln\left|D^T \cdot P^{-1} \cdot D\right|. \quad (12)$$

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

$$P = I_m + C^I \cdot \Sigma_k^{X^I} \cdot (C^I)^T. \quad (13)$$

Observe that, given $\Sigma_k^{X^I}$, all terms in Eq. (13) have relatively small dimensions and P can be computed efficiently for each candidate action, with time complexity $O(m^3)$ not depending anymore on state dimension n, similarly to our previous work [13]. Calculation of the inverse $P^{-1} \in \mathbb{R}^{m \times m}$, which is required in Eq. (12), is also $O(m^3)$ and will also not depend on n. The run-time of overall calculation in Eq. (12) will have complexity $O(m^3 + n'^3)$ and will depend only on number of new factors m 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/m lead to a bigger advantage of our approach vs the alternatives (see Section V).

It is worthwhile to mention a specific case, where $m = n'$, which happens for example, 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. (12) will be reduced to $J_{\mathrm{IG}}(a) = \frac{n' \cdot \gamma}{2} + \ln\left|D\right|$. In other words, the information gain in such case depends only on the partition D of A (see Fig. 1(a)), while the prior Λ_k is not involved in the calculations at all.

Remark 1: It is possible that posterior state dimension will be different for different candidate actions (e.g. see Section V). 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. (12) 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 paper.

C. Focused BSP

The `focused` scenario 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, as illustrated in Fig. 1(b), i.e. $X_{k+L}^F \subseteq Y$. 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 Fig. 1(c), i.e. $X_{k+L}^F \subseteq X \equiv X_k$. 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 2: In most cases the actual variable ordering will be more sporadic than the one depicted in Fig. 1. For example, iSAM [19] determines variable ordering using COLAMD [22] to enhance sparsity of the square root information matrix. We note our approach applies to any arbitrary variable ordering, with the equations derived herein remaining unchanged.

1) **`Focused BSP`** **($X_{k+L}^F \subseteq Y$)** *:* First we define additional partitions of Jacobian A (see Fig. 1(b)). The submatrices C, D, C^I and $C^{\neg I}$ were already introduced in the sections above. We now further partition D into D^F and D^U, that correspond, respectively, to columns of new variables that are focused and unfocused. Denote the former set of variables as YF and the latter as Y^U (see also Table I). Note, $Y^F \equiv X_{k+L}^F$.

Lemma 2: The posterior entropy of Y^F (Eq. (9)) is given by

$$J_{\mathcal{H}}^F(a) = \frac{n_F \cdot \gamma}{2} + \frac{1}{2}\ln\left|(D^U)^T \cdot P^{-1} \cdot D^U\right| - \frac{1}{2}\ln\left|D^T \cdot P^{-1} \cdot D\right| \quad (14)$$

where P is defined in Eq. (13).

The proof of Lemma 2 is given in Appendix B in [18].

We got an exact solution for $J_{\mathcal{H}}^F(a)$ that, given $\Sigma_k^{X^I}$, can be calculated efficiently with complexity $O(m^3 + n'^3)$, similarly to unfocused BSP in Section III-B. In Section III-D we will explain how the prior marginal covariance term ($\Sigma_k^{X^I}$) can be efficiently retrieved, providing a fast solution for `focused` BSP.

2) **`Focused BSP`** **($X_{k+L}^F \subseteq X$)** *:* Similarly to the previous section, we first introduce additional partitions of Jacobian A for the considered case (see Fig. 1(c)). From the figure we can see that $C^{\neg I}$ can further be partitioned into $C^{\neg I,U}$ and $C^{\neg I,F}$. In particular, C^{-I} represents columns of old variables that are both not involved and unfocused, and $C^{\neg I}$ represents columns of old variables that are both not involved and focused. We denote the former group of variables by $X^{\neg I,U}$ and the latter by $X^{\neg I,F}$ (see Table I). Likewise, C^I can be partitioned into $C^{I,U}$ and $C^{I,F}$, representing old involved variables that are, respectively, unfocused ($X^{I,U}$) or focused ($X^{I,F}$). Note that in this case, the set of focused variables is $X_{k+L}^F = X_k^F = \{X^{\neg I,F} old \cup X^{I,F}\}$.

Lemma 3: The `focused` IG of C_k^F is given by

$$J_{\mathrm{IG}}^F(a) = \frac{1}{2}(\ln\left|P\right| + \ln\left|D^T \cdot P^{-1} \cdot D\right| - \ln\left|S\right| - \ln\left|D^T \cdot S^{-1} \cdot D\right|), \quad (15)$$

where P is defined in Eq. (13), and

$$S \doteq I_m + C^{I,U} \cdot \Sigma_k^{X^{I,U}|F} \cdot (C^{I,U})^T \quad (16)$$

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

The proof of Lemma 3 is given in Appendix C in [18].

Similarly to the cases discussed above (Sections III-B and III-C1), given $\Sigma_k^{X^I}$ and $\Sigma_k^{X^{I,U}|F}$, calculation of $J_{\mathrm{IG}}^F(a)$ per each action a can be performed efficiently with complexity $O(m^3 + n'^3)$, independently of state dimension n. The next section presents our approach to calculate the appropriate entries in the prior covariance only once and re-use the result whenever required.

D. Re-Use Technique

As we have seen above, `unfocusedBSP` (Section III-B) and `focusedBSP` ($X_{k+L}^F \subseteq Y$) (Section III-C1) problems require prior marginal covariance of the involved variables,

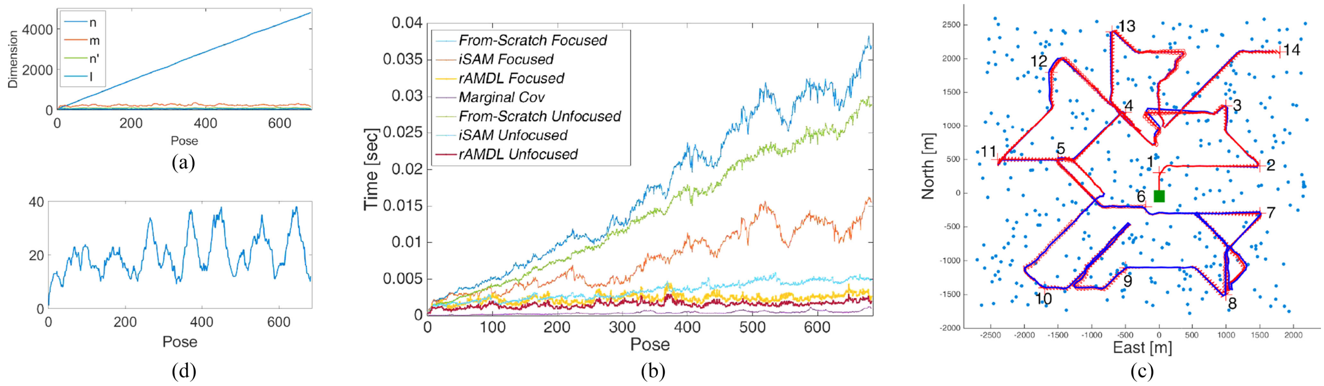

Fig. 2. Focused BSP scenario with focused robot's last pose. (a) Dimensions of the BSP problem (state dimension, average number of new terms, average number of new variables, average number of old involved variables) at each time; (b) Running time of planning, i.e. evaluating impact of all candidate actions, each representing possible trajectory, normalized by number of candidates; Results are shown both for focused and unfocused cases. The lowest line, labeled *Marginal Cov*, represents time it took to calculate prior marginal covariance $\Sigma_k^{X_{All}}$ in *rAMDL* approach (see Section III-D). (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) Number of action candidates at each time.

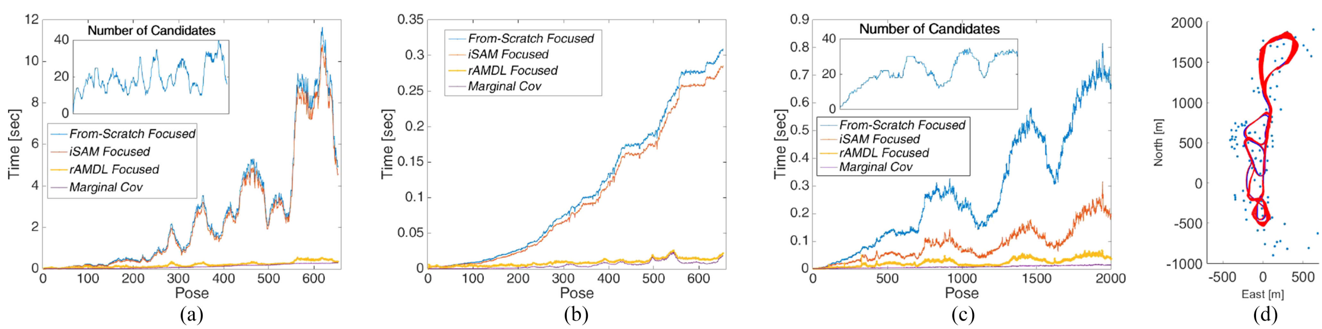

Fig. 3. (a)-(b) FocusedBSP scenario with focused landmarks: (a) Running time of planning, i.e. evaluating impact of all candidate actions, each representing possible trajectory; (b) Running time from (a) normalized by number of candidates. (c)-(d) FocusedBSP scenario using Victoria Park dataset: (c) Running time of planning; (d) Final robot trajectory. The lowest line in (a)-(c), labeled *Marginal Cov*, represents time it took to calculate prior marginal covariance $\Sigma_k^{X_{All}}$ in *rAMDL* approach (see Section III-D).

i.e. X^I, in order to use the developed expressions given by Eqs. (12) and (14). 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, and conceptually similar to our previous work [13], 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 (3) for at least one candidate action. We can now perform a one-time calculation of the prior marginal covariance for this set, i.e. $\Sigma_k^{X_{All}}$. 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 [24]. In Section V we show that calculation time of $\Sigma_k^{X_{All}}$ while exploiting sparsity [23], [25] is relatively small comparing to total decision making time. Still, the more detailed discussion about complexity of covariance retrieval can be found in publications [23], [24].

For focused BSP $(X_{k+L}^F \subseteq X)$ case (Section III-C2), we also need the term $\Sigma_k^{X^{I,U}|F}$ (see Eq. (16)). This term can be computed by first calculating the prior marginal covariance $\Sigma_k^{\{X^{I,U},F\}}$ for the set of variables $\{X^{I,U}, X_k^F\}$, and then writing the Schur complement over the relevant partitions in $\Sigma_k^{\{X^{I,U},F\}}$

$$\Sigma_k^{X^{I,U}|F} = \Sigma_k^{X^{I,U}} - \Sigma_k^{X^{I,U}/F} \cdot (\Sigma_k^F)^{-1} \cdot \Sigma_k^{F/X^{I,U}}. \tag{17}$$

Consequently, we can use a one-time calculation also for the focused BSP $(X_{k+L}^F \subseteq X)$ case as follows. Let us extend the set X_{All} to contain also all focused variables. Once $\Sigma_k^{X_{All}}$ is calculated, $\Sigma_k^{\{X^{I,U},F\}}$ will be just its partition and can be easily retrieved from it. As a result, the calculation of $\Sigma_k^{X^{I,U}|F}$ per candidate action becomes computationally cheap (through Eq. (17)). Furthermore, term $(\Sigma_k^F)^{-1}$ can be calculated only once for all candidates.

Remark 3: It is worth mentioning that in practice some of the rows of C^I and $C^{I,U}$ will be zero. Taking this fact into account allows to further reduce calculations. The corresponding expressions are omitted here due to page limit.

Remark 4: In full SLAM formulation, where X_k contains both robot poses and landmarks, and where only landmarks mapped by time t_k are the focused variables (e.g. to improve map accuracy), it is possible to reduce expressions of `focused BSP` ($X^F_{k+L} \subseteq X$) even further. It can be shown that $X^{I,U} \equiv x_k$ where x_k denotes the robot's current pose; as a result, $\Sigma_k^{X^{I,U}|F}$ becomes the same for all candidate actions and 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.

IV. 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 [19] to incrementally update the posterior. Here the (linearized) system is represented by a squared 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 [19] 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 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 [19], [23].

While this 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 in Eq. (12) has constant complexity in general, given the prior marginal covariance terms that are calculated only once.

V. Results

In this section we present simulation results of applying our approach to autonomous navigation (both `unfocused` and `focused` cases) on synthetic and real-world datasets. Robot must visit set of goals in unknown environment while minimizing objective of uncertainty. The code is implemented in Matlab; we use the GTSAM library [19], [26]. All scenarios were executed on a Linux machine with i7 2.40 GHz processor and 32 Gb of memory.

In our synthetic scenario (Fig. 2(c)), the robot's task is to sequentially visit a predefined set of goals $\mathcal{G} = \{G_1, .., G_{14}\}$ in unknown environment while reducing an uncertainty objective metric. More specifically, at each timestep robot selects best non-myopic action $a = u_{k:k+L-1}$, executes its first control u_k and observes landmarks in radius of 900 meters that can be old (seen before) and new (met first time). Further, the state estimation is updated, with X_k containing all robot poses and mapped landmarks till time t_k. Candidate actions contain one action that navigates robot to current goal G_i from a predefined set $\mathcal{G}$ (see Fig. 2(c)) and a set of "loop-closure" actions that are generated as following. First, landmarks seen till current time in radius of 1000 meters from robot's current position are clustered, similar to [10]. Each cluster's center g_{cl} represents "loop-closure" target and contributes a candidate action $a_{\text{cl}} = u_{k:k+L-1}$ that takes robot to g_{cl}. The a_{cl} is constructed by first descretizing map into grid and thereafter searching for optimal trajectory from current position to g_{cl} through A^* search algorithm, similarly to [6], [10]. The objective has two terms - distance to the current goal G_i and uncertainty of the last pose (Eq. (14)) in planning segment $J(a) = d(x_{k+L}, G_i) + J^F_{\mathcal{H}}(a)$. However, the presented running time refers only to the uncertainty term, as it is the focus of this paper and because calculation complexity of first term (euclidean distance) is relatively insignificant. As can be seen from above, we consider a nonmyopic setting and let each candidate action represent trajectory of various length. The state augmentation is done by introducing new robot poses representing the trajectory. Limiting the clustering process to a specific radius is done in order to bound the horizon lag of candidate actions.

In parallel, unfocused uncertainty objective $J_{\text{IG}}(a)$ is calculated (Eq. (12)), only for the purpose of performance comparison between `focused` and `unfocused` cases. The robot's motion is controlled only by focused objective.

Three techniques were applied to solve the planning problem - more common techniques *From-Scratch* and *iSAM* (Section IV) and the proposed *rAMDL* technique (Sections III-B and III-C1). The calculated values of objective were numerically compared to validate that all three approaches are calculating exactly the same metric.

In Fig. 2(b) it can be clearly seen that while *iSAM* is faster than *From-Scratch*, time of both techniques is growing with state dimension, as was mentioned before. On the other hand, time of the *rAMDL* approach is shown to be bounded, due to horizon lag of all candidates being limited (see Fig. 2(a)). Number of candidates in our scenario is around 20 (Fig. 2(d)). Even with such relatively small candidate set our approach is faster by order than its alternatives. This trend appears to be correct for both `focused` and `unfocused` objective functions, though for the later the *iSAM* comes very close to *rAMDL* technique.

While comparing time that it took both *From-Scratch* and *iSAM* to calculate `focused` vs `unfocused` objective functions, it is easy to see that unfocused one is done much faster. The reason 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 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 [19]. While in our simulation we did not modified the default ordering of iSAM (COLAMD heuristic), different strategies of ordering can be point for future investigation.

For *rAMDL* approach both `focused` and `unfocused` objective functions (Eq. (12) and (14)) have similar complexity which is supported by plotted times.

Next, we repeated our navigation scenario but this time, X_{k+L}^F contained only landmarks seen by time k (see Fig. 3(a) and (b)). Such `focused` set causes both *From-Scratch* and *iSAM* techniques to be much slower comparing to their performance on the first scenario where X_{k+L}^F contained only x_{k+L}. The reason for this is that X_{k+L}^F's dimension is much higher here, representing dimension of all landmarks, and computation of its marginal covariance is much more expensive. In the contrast, performance of *rAMDL* has been barely changed due to beneficial *re-use* of calculations.

We also performed a hybrid simulation where part of real-world Victoria Park dataset was used for offline planning (see Fig. 3(c) and (d)). At each timestep we collected candidate actions by clustering landmarks seen till that time, just as it was done in our first simulation. Further, we calculated focused objective function for each candidate with X_{k+L}^F containing only x_{k+L}. After evaluating all candidates, robot was moved to the next pose according to the dataset. Reminding that our main contribution is too reduce time complexity, even though the candidate metrics were not really used in hybrid scenario, it allowed us to compare time performance of all presented techniques. As can be seen, also here the *rAMDL* outperforms both of its alternatives, keeping the same trends that were observed in previous simulations.

VI. Conclusions

We developed a computationally efficient and exact approach for non-myopic `focused` and `unfocused` belief space planning (BSP) in high dimensional state spaces. As a key contribution we developed an augmented version of the well-known matrix determinant lemma and use it 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 a BSP setting. The approach was examined in simulation and in a real-world dataset considering the problem of autonomous navigation in unknown environments, exhibiting superior performance compared to the state of the art.

References

[1] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1, pp. 99–134, 1998.

[2] J. Pineau, G. J. Gordon, and S. Thrun, "Anytime point-based approximations for large pomdps." *J. Artif. Intell. Res.*, vol. 27, pp. 335–380, 2006.

[3] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-blackwellized particle filters," in *Proc. Robot., Sci. Syst.*, 2005, pp. 65–72.

[4] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robot. Res.*, vol. 28, no. 11/12, pp. 1448–1465, 2009.

[5] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "Firm: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements," *Int. J. Robot. Res.*, vol. 33, no. 2, pp. 268–304, 2014.

[6] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 849–882, 2015.

[7] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *Int. J. Robot. Res.*, vol. 31, no. 11, pp. 1263–1278, 2012.

[8] J. M. Walls, S. M. Chaves, E. Galceran, and R. M. Eustice, "Belief space planning for underwater cooperative localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 2264–2271.

[9] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Pérez, "Belief space planning assuming maximum likelihood observations," in *Proc. Robot., Sci. Syst.*, Zaragoza, Spain, 2010, pp. 587–593.

[10] A. Kim and R. M. Eustice, "Active visual slam for robotic area coverage: Theory and experiment," *Int. J. Robot. Res.*, vol. 34, no. 4/5, pp. 457–475, 2014.

[11] R. Valencia, M. Morta, J. Andrade-Cetto, and J. Porta, "Planning reliable paths with pose SLAM," *IEEE Trans. Robot.*, vol. 29, no. 4, pp. 1050–1059, Aug. 2013.

[12] Z. Bai, G. Fahey, and G. Golub, "Some large-scale matrix computation problems," *J. Comput. Appl. Math.*, vol. 74, no. 1, pp. 71–89, 1996.

[13] D. Kopitkov and V. Indelman, "Computationally efficient decision making under uncertainty in high-dimensional state spaces," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 1793–1800.

[14] V. Indelman, "No correlations involved: Decision making under uncertainty in a conservative sparse information space," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 407–414, Jan. 2016.

[15] D. Levine and J. P. How, "Sensor selection in high-dimensional Gaussian trees with nuisances," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2211–2219.

[16] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact Pose SLAM," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 78–93, Feb. 2010. [Online]. Available: http://dx.doi.org/10.1109/TRO.2009.2034435

[17] B. Mu, A.-A. Agha-mohammadi, L. Paull, M. Graham, J. How, and J. Leonard, "Two-stage focused inference for resource-constrained collision-free navigation," in *Proc. Robot., Sci. Syst.*, 2015.

[18] D. Kopitkov and V. Indelman, "Computationally efficient belief space planning via augmented matrix determinant lemma and re-use of calculations—Supplementary material," Technion—Israel Inst. Techn., Haifa, Israel, Tech. Rep. ANPL-2016-02, 2016. [Online]. Available: http://vindelman.technion.ac.il/Publications/Kopitkov17ral_Supplementary.pdf

[19] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Intl. J. Robot. Res.*, vol. 31, pp. 217–236, Feb. 2012.

[20] S. M. Chaves, J. M. Walls, E. Galceran, and R. M. Eustice, "Risk aversion in belief-space planning under measurement acquisition uncertainty," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots Syst.*, 2015, pp. 2079–2086.

[21] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, 2008.

[22] T. Davis, J. Gilbert, S. Larimore, and E. Ng, "A column approximate minimum degree ordering algorithm," *ACM Trans. Math. Softw.*, vol. 30, no. 3, pp. 353–376, 2004.

[23] M. Kaess and F. Dellaert, "Covariance recovery from a square root information matrix for data association," *Robot. Auton. Syst.*, vol. 57, no. 12, pp. 1198–1210, 2009.

[24] V. Ila, L. Polok, M. Solony, P. Smrz, and P. Zemcik, "Fast covariance recovery in incremental nonlinear least square solvers," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 4636–4643.

[25] G. Golub and R. Plemmons, "Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition," *Linear Algebra Appl.*, vol. 34, pp. 3–28, Dec. 1980.

[26] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. GT-RIM-CP&R-2012-002, Sep. 2012.