

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.xxxx.DOI

# Speeding up Gaussian Belief Space Planning for Underwater Robots Through a Covariance Upper Bound

HUAN YU<sup>1,2</sup>, WENJIE LU<sup>2</sup>, DIKAI LIU<sup>2</sup>, YONGQIANG HAN<sup>1</sup>, QINGHE WU<sup>1</sup>,

<sup>1</sup>School of Automation, Beijing Institute of Technology, Beijing, 100081 China

<sup>2</sup>Centre for Autonomous Systems, University of Technology Sydney, NSW, 2007, Australia

Corresponding author: Yongqiang Han (hanyongqiang213@163.com).

This work was supported in part by the Australian Research Council (ARC) Linkage Project (LP150100935), the Roads and Maritime Services of NSW, the Centre for Autonomous Systems (CAS) at the University of Technology Sydney, and China Scholarship Council (CSC).

**ABSTRACT** Existing belief space motion planning methods are not efficient for underwater robots that are subject to spatially varying motion and sensing uncertainties arising from the non-uniform current disturbances and landmark populations, respectively. Based on a closed-loop stochastic control framework, we propose a fast Gaussian belief space planning approach for coupled optimization of trajectory, localization and control, resulting in a non-linear programming problem (NLP). In particular, as opposed to advancing the covariance by a Kalman filter in the existing literature, we utilize an upper bound of the trace propagation of the covariance, thereby avoiding to solve Riccati equations and thus, reducing the computational complexity. The NLP is then efficiently solved by sequential quadratic programming based on the initial solutions obtained from RRT-connect. These initials lie in multiple homotopy classes guaranteed by H-signature discrimination, leading to global optimality with probability one as the number of samples in RRT-connect goes to infinity. Numerical simulations on holonomic and non-holonomic autonomous underwater vehicles (AUVs) and an Intervention-AUV with a manipulator in cluttered underwater environments demonstrate that optimal and collision-free trajectories with low localization uncertainty are obtained efficiently.

**INDEX TERMS** Underwater robots, motion planning, belief space, localization uncertainty, stochastic control.

## I. INTRODUCTION

UNMANNED underwater vehicles (UUVs) include autonomous underwater vehicles (AUVs) and non-autonomous remotely operated underwater vehicles (ROVs) – controlled and powered from the surface by an operator/pilot via an umbilical or using remote control [1]. These vehicles are a crucial part of oceanic exploration and exploitation. With applications in underwater environment mapping, surveillance, and oceanographic data collection (e.g., biological and chemical) [2]–[4], underwater robots are required to autonomously execute increasingly complicated tasks in uncertain underwater environments. In recent years, the development of underwater perception and navigation technologies, such as sonar, cameras, and USBL (underwater ultra-short baseline), also made it possible to develop more intelligent and autonomous ROVs [5], [6]. In addition to

autonomously moving in various underwater environments, Intervention-AUVs (I-AUVs) equipped with multi-degree of freedom (DOF) manipulators can perform complex intervention tasks such as construction, rescue and salvage [7], [8].

For a given task of AUVs, other than perception and control, path/motion planning, which solves an optimal trajectory that connects the start and goal configuration while satisfying problem-specific requirements such as the shortest path or time and low-energy consumption, is a critical part of task execution [9], [10]. In the past decades, as a fundamental research topic in robotics, many crucial breakthroughs in feasibility and optimality have made the solutions of motion planning algorithms largely practical for various tasks. Most motion planners assume that a robot's motion and observation are perfectly predictable and deterministic, without considering uncertainties resulting from noisy actua-

tors, sensors, and other unmodeled noise. Solutions based on these assumptions in real-world environments may result in collisions and task execution failures.

For underwater robots, uncertainties typically originate from three sources: (i) motion uncertainty arising from imprecise system modeling, noisy actuators and disturbance of water flow; (ii) sensing uncertainty arising from noisy sensor measurements; (iii) map uncertainty arising from uncertain or dynamic obstacles in the work space. In this study, we mainly handle the first two uncertainties. Considering the temporal-spatial variability of water flow, the motion uncertainty in this work is simplified as a spatially varying process noise. The sensing noise of underwater positioning sensors, such as USBL and cameras, also processes features that vary with spatial distribution, which depends on the deployment and distribution of transceivers and responders or the landmark populations. Therefore, in this work, we solve the problem of AUV motion planning with spatially varying motion and observation uncertainties in cluttered underwater environments.

In the presence of uncertainty, the state of the robot is indeterminant, as opposed to using a determinant value for decision making, and the state can be described by a probability distribution known as *belief* [11]. Planning under uncertainty generally can be formulated as *partially observable Markov decision processes* (POMDPs) [12]. The solution of planning problem is in a *belief space* rather than a configuration (or state) space. However, due to the computational complexity (the curse of history and dimensionality), state-of-the-art POMDP solvers can solve only small-scale discrete state problems and are intractable for continuous state-space problems. In recent years, to make the planning problem tractable, many studies have explored transforming deterministic space planning methods into belief space [13]–[15].

In this work, Based on the Linear Quadratic Gaussian (LQG) control framework, we propose a fast Gaussian belief space planning approach for AUVs under spatially varying motion and observation noise. As opposed to the decoupled design of trajectory and controller, to solve an optimal trajectory (with low localization uncertainty, being collision-free, etc.), we adopt a coupled optimization of the trajectory, localization and control based on the separation principle of LQG. Then the motion planning problem in belief space is formulated as a non-linear programming (NLP) problem, as shown in section IV-B. However, to obtain an optimal solution in a short time, there can be two issues.

The first issue is how to solve an optimal solution with NLP solvers such as sequential quadratic programming (SQP). The solution is sensitive to the initial value of the solver. To obtain an optimal solution of the NLP problem, we use RRT-connect [16] and the H-signature classification method [17] to find multiple initial trajectories from different homotopy classes and then select a near-optimal solution among all solutions based on different initials as the nominal trajectory of the feedback controller.

The second issue is how to reduce the computational

complexity of the NLP problem produced by matrix multiplication and inversion in Kalman filter (KF)-based belief propagation process. For uncertainty minimization in the NLP problem, general state estimation in LQG and its variants is based on the KF or extended Kalman filter (EKF), and beliefs are parameterized in terms of mean states and covariances [18]. However, discrete covariance or belief propagation accounts for a large amount of the computational cost for nominal trajectory optimization [18], [19]. In our method, to reduce this cost, we use the upper bound of the trace of the covariance matrix as the metric of state uncertainty, which is a continuous scalar function, as shown in section IV-A.

We evaluate our approaches on three types of underwater robots: a holonomic AUV [20], a torpedo-shaped non-holonomic Sparus II AUV shown in [21], and an I-AUV with 3-DOF manipulator [22]. Computational complexity analysis and numerical simulation results show that in cluttered environments with various spatially varying motion or observation noise, our method requires less computation than other approaches.

*Outline:* In the next section, we discuss the most relevant prior works on belief motion planning and its applications in AUVs. Section III provides the problem of belief space motion planning based on stochastic optimal control. Section IV describes the framework of our solution. Section V shows results and comparisons of numerical simulations. Finally, Section VI gives the conclusions of this paper and presents future work.

## II. RELATED WORK

In the research field of AUV task execution under uncertainties, many studies focus on robust control [23]–[25], while only a few works consider uncertainties in the motion planning phase [26]–[28]. However, a substantial body of work has addressed uncertainty in motion planning, which can be applied to AUV trajectory planning.

The first class includes general POMDP solvers, which provide a principled mathematical framework for planning under uncertainty. The advantages of these solvers are that they are model-free and distribution-free. They do not limit the forms of dynamics and sensing model and probability distributions of the robot states. However, they are known to be of extreme complexity, and can only be directly applied to problems with small and low-dimensional state spaces [12]. Point-based POMDP solvers, such as [29]–[31], have increased the size of problems that can be solved by POMDPs. However, they do not handle continuous state, control, and observation spaces. Based on the Gaussian distribution assumption, the approaches in [19], [32] provide locally optimal solution for continuous POMDPs in a Gaussian belief space. As presented in the authors' analyses in those works, those methods still have high computational complexity for complex systems or issues with high-dimensional state spaces.

The second class includes sampling-based methods. In systems without uncertainty, sampling-based motion plan-

ning methods, such as graph-based probabilistic roadmap methods (PRMs), tree-based planners rapidly-exploring randomized trees (RRTs), and their variants, are the most popular planners. The most intuitive method for planning under uncertainty is directly changing the sampling space into a belief space. However, there are two main challenges arising from uncertainty. The first challenge is how to ensure the reachability between the nodes in a graph. The second challenge is that the cost of each edge is not independent, which violates the independence assumption in sampling-based methods. Although the feedback-based information roadmap (FIRM) proposed in [13] provides a solution framework for belief space sampling, that method requires time-consuming offline graph construction and a large number of sample nodes to obtain an optimal solution.

A third class formulates the belief planning problem as an optimal control or a stochastic control problem. To simplify the POMDP problem, as used in this work, most approaches constrain the problem in the LQG framework. One intuitive approach is decoupling the trajectory optimization and control. As shown in [14], the trajectory search is in a deterministic space with an RRT-like planner and then selects an optimal trajectory with the best LQG control performance. Rather than generating random paths, the belief trees proposed in [33] solves for a globally optimal solution by utilizing an RRT\*-like pruning strategy and constructing a belief tree. Belief space iLQR [18] and B-LQR [34] solve a locally optimal solution based on the gradient descent optimization method by using the decoupled process in an iterative manner. As in our work, they also take state-dependent motion and observation noise into consideration. However, the computational complexity is higher because the optimization dimension ( $n + n^2$ ) includes the state and covariance, and the solution is locally optimal.

A fourth class is based on the coupled design of trajectory and control such as MLO [15] and T-LQG [35]. MLO solves for the optimal trajectory in the belief space with the maximum likelihood observation assumption and then tracks the trajectory with LQR control. T-LQG optimizes the trajectory in state space and solves for a locally optimal trajectory, which reduces the computational complexity compared with methods in a covariance augmented space. In these works, the cost of uncertainty is generally denoted by a function of the covariance, and Riccati equations are solved at every step. Our approach reduces this computational complexity by utilizing an upper bound function of the covariance.

### III. PROBLEM DEFINITION

#### A. CONTINUOUS-TIME SYSTEM MODEL

In this work, the general POMDP problem is formulated as a stochastic optimal control problem. Consider the following stochastic dynamic model:

$$\begin{aligned}\dot{\mathbf{x}}_t &= \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{G}(\mathbf{x}_t) \cdot \boldsymbol{\omega}_t \\ \mathbf{z}_t &= \mathbf{f}(\mathbf{x}_t) + \mathbf{F}(\mathbf{x}_t) \cdot \boldsymbol{\nu}_t,\end{aligned}\quad (1)$$

where  $\mathbf{x}_t$  denotes the system state, initialized from  $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$ , and  $\boldsymbol{\omega}_t$  and  $\boldsymbol{\nu}_t$  are stochastic disturbances (isotropic noise). For white-noise processes,  $E[\boldsymbol{\omega}_t] = 0$ ,  $E[\boldsymbol{\omega}_t \boldsymbol{\omega}^T(\tau)] = \mathbf{Q}_t \delta(t - \tau)$ ,  $E[\boldsymbol{\nu}_t] = 0$ ,  $E[\boldsymbol{\nu}_t \boldsymbol{\nu}^T(\tau)] = \mathbf{R}_t \delta(t - \tau)$ . The matrices  $\mathbf{Q}$  and  $\mathbf{R}$  represent the power spectral density (PSD matrices) and  $\delta$  denotes the Dirac delta function. We also assume that the noise vectors are mutually independent  $E[\boldsymbol{\omega}_t \boldsymbol{\nu}_\tau^T] = 0$ . The noise  $\mathbf{G}(\mathbf{x}_t) \cdot \boldsymbol{\omega}_t$  and  $\mathbf{F}(\mathbf{x}_t) \cdot \boldsymbol{\nu}_t$  denote the spatially varying motion and observation noise distributions as in [18]. We assume that the motion disturbance  $\mathbf{G}(\mathbf{x}_t) \cdot \boldsymbol{\omega}_t$  stands for a lumped uncertainty arising from model parameters uncertainty and external disturbance.

In our work, the state of the motion model depends on the specific AUV, which may include the position, attitude and configuration of the manipulators of the I-AUV. Detailed definitions of an AUV models are presented in section V. For the observation model, we assume that visual sensors, underwater-GPS (USBL), IMU, pressure sensors and angle encoders are mounted at the appropriate positions of the robot, which provide the measurement of the base position, attitude and manipulator configuration [5], [6].

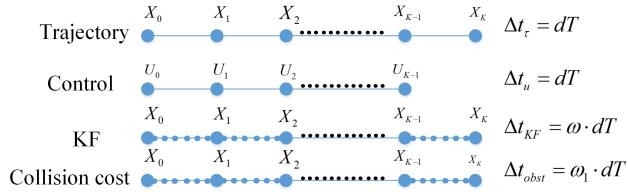
#### B. PLANNING BASED ON MULTIPLE RESOLUTION PROCESSES

The continuous-time system in Eq. (1) can be discretized into a discrete system with discrete time interval  $dT$ . The task of motion planning is to obtain a sequence of control actions  $\mathbf{u}_{0:K-1}$ . To solve the belief space planning problem efficiently, as shown in Fig. 1, there are four resolutions for different processes: trajectory and control are parameterized at the same resolution ( $\Delta t_\tau = \Delta t_u = dT$ ), and KF and collision checking are performed at a denser resolution ( $\Delta t_{KF} = \omega \cdot dT$ ,  $\Delta t_{obst} = \omega_1 \cdot dT$ , and  $\omega, \omega_1 < 1$ ).

For a fixed planning time  $T$ , the discrete planning horizon is  $K = T/dT$ . Smaller  $dT$  will increase the optimization dimension of the planning problem. Generally, an AUV is a slow dynamic system; therefore, the trajectory planning can be processed at a slow rate.

To estimate the belief of the state on a trajectory, discrete-time covariance update and continuous-time solvers (such as the Runge-Kutta-Merson algorithm) iteratively calculate the trace of covariance at each sampling time  $k\Delta t_{KF}(0 \leq k \leq \frac{T}{\Delta t_{KF}})$ . Considering the equivalence between continuous- and discrete-time KFs, the state and covariance propagation should proceed in a small time interval. The measurement frequency of the sensors (such as cameras) can be 10 Hz, 100 Hz or higher, which guarantees that the filtering process works at high-resolution. The equivalence transformation functions are given in Appendix -C, and in section V-A, we provide detailed simulations about the equivalence. Trajectory optimization is based on the gradient descent method. A small sample time for KF makes the cost function and optimized trajectory smoother.

To guarantee that the trajectory is collision free, a collision check should also be performed in a small discrete time  $\Delta t_{obst}$ .



**FIGURE 1:** Trajectory optimization based on multiple resolutions. The trajectory and control parameterize the resolution of the optimized trajectory. The KF and collision cost work at a higher resolution.

### C. DISCRETIZATION OF CONTINUOUS MODEL

We first linearize the AUV system and then formulate a linear quadratic Gaussian system. Let an nominal control sequence be denoted by  $\{\mathbf{u}_t^p\}_{t=0}^{K-1}$ , and  $\mathbf{x}_{t+1}^p = f(\mathbf{x}_t^p, \mathbf{u}_t^p, 0)$  for  $0 \leq t \leq K - 1$ , where  $\mathbf{x}_0^p = \hat{\mathbf{x}}_0$ . Linearization of the nonlinear motion models in Eq. (1) in discrete time with respect to the nominal control and state sequence yields

$$\begin{aligned}\tilde{\mathbf{x}}_{t+1} &= \mathbf{A}_t^p \tilde{\mathbf{x}}_t + \mathbf{B}_t^p \tilde{\mathbf{u}}_t + \mathbf{G}_t^p \boldsymbol{\omega}_t \\ \tilde{\mathbf{z}}_t &= \mathbf{H}_t^p \tilde{\mathbf{x}}_t + \mathbf{M}_t^p \boldsymbol{\nu}_t,\end{aligned}\quad (2)$$

where  $\tilde{\mathbf{x}}_t := \mathbf{x} - \mathbf{x}_t^p$ ,  $\tilde{\mathbf{u}}_t := \mathbf{u} - \mathbf{u}_t^p$ , and  $\tilde{\mathbf{z}}_t := \mathbf{z} - \mathbf{z}_t^p$  are the state, control and observation error, respectively. The matrices in the model are the corresponding Jacobian matrices around the nominal trajectory  $\{\mathbf{u}_{0:K-1}^p, \mathbf{x}_{0:K}^p\}$ .

### D. STOCHASTIC OPTIMAL CONTROL

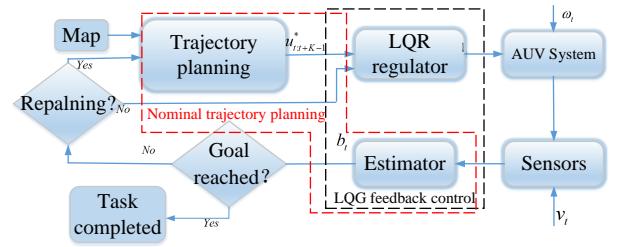
As shown in Eq. (1), due to the partial or noisy observations and lack of an accurate motion model, it might be difficult to obtain an accurate AUV state, resulting in localization uncertainty. Therefore, the trajectory planning problem is built in a belief space. The *Belief*  $b[\mathbf{x}_t] \in \mathbb{B}$  is the conditional distribution of  $\mathbf{x}_t$  and is represented by a Gaussian in a belief space  $\mathbb{B}$  [11]:

$$\begin{aligned}b[\mathbf{x}_t] &= p[\mathbf{x}_t | \mathbf{u}_0, \dots, \mathbf{u}_{t-1}; \mathbf{z}_0, \dots, \mathbf{z}_t] \\ b[\mathbf{x}_{t+1}] &= \tau(b[\mathbf{x}_t], \mathbf{u}_t, \mathbf{z}_t),\end{aligned}\quad (3)$$

where  $b[\mathbf{x}_{t+1}] : \mathbb{X} \times \mathbb{Z} \times \mathbb{U} \rightarrow \mathbb{R}$  and  $\tau : \mathbb{B} \times \mathbb{U} \times \mathbb{Z} \rightarrow \mathbb{B}$  is a belief propagation function. In this work, given the assumption of Gaussian noise, the belief can be denoted by the mean and covariance, and the propagation of belief can be based on the filtering method (such as the KF). In the remainder of this paper, let  $\mathbf{b}_t$  denote  $b[\mathbf{x}_t]$ .

The purpose of this work is to solve the following stochastic control problem:

**Problem 1. Stochastic optimal control:** Given the initial belief  $\mathbf{b}_0$ , the goal belief  $\mathbf{b}_K$ , and constraints on  $\mathbf{b}_t$  and  $\mathbf{u}_t$ , find the optimal control policy  $\pi_t : \mathbb{B} \rightarrow \mathbb{U}$  for all  $0 \leq t \leq K$



**FIGURE 2:** Closed-loop feedback control loop with motion planner and controller.

that minimizes the objective function  $J^\pi$ :

$$\begin{aligned}\min_\pi J^\pi &:= \mathbb{E} \left[ \sum_{t=0}^{K-1} c_t^\pi(\mathbf{b}_t, \mathbf{u}_t) + c_K^\pi(\mathbf{b}_K) \right] \\ \text{s.t. } \mathbf{b}_{t+1} &= \tau(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_{t+1}), \\ g_i(\mathbf{b}_f, \mathbf{u}_t) &\leq 0, \quad i = 1, \dots, m\end{aligned}\quad (4)$$

where  $c_t^\pi$  is the instant cost at timestep  $t$ , which could be a measure of the state uncertainty, control effort and/or obstacle cost;  $c_K^\pi$  denotes the terminal belief state cost;  $g_i(\mathbf{b}_f, \mathbf{u}_t)$  denotes constraints on the belief and control such as workspace and actuator force limits; and the index  $K$  is the terminal timestep and is fixed, but it could also be a variable to be optimized [36], [37].

### IV. GAUSSIAN BELIEF SPACE PLANNING AND CONTROL

As shown in Fig. 2, our framework includes mainly nominal trajectory planning, feedback control and replanning. We restrict the optimal control policy of **Problem 1** in the LQG framework, which provides an optimal control policy for linearized systems with Gaussian disturbance. For a partial observation system, LQG consists of a state estimator and LQR controller based on the separation principle. Therefore, the cost of the stochastic control problem is divided into estimation and control costs.

In the estimation stage, to reduce the estimator error and control effort, we define the quadratic form of the estimation cost:

$$J := \mathbb{E} \left[ \sum_{t=1}^K \tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t + \mathbf{u}_{t-1}^T \mathbf{W}_t^u \mathbf{u}_{t-1} \right], \quad (5)$$

where  $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$  is the estimator error, and  $\mathbf{W}_t^x$  and  $\mathbf{W}_t^u$  are positive-definite weight matrices. The estimated state  $\hat{\mathbf{x}}_t$  can be obtained by filtering methods. As shown in [35], the estimation cost can also be approximated by the nominal belief, which depends on the underlying nominal trajectory  $\{\mathbf{u}_{0:K-1}^p, \mathbf{x}_{0:K}^p\}$ ,

$$J^p := \sum_{t=1}^K (\mathbb{E}[\tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t] + (\mathbf{u}_{t-1}^p)^T \mathbf{W}_t^u \mathbf{u}_{t-1}^p). \quad (6)$$

Define  $\mathbf{P}_{b_t^p}^+(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p) := \mathbb{E}[(\mathbf{x} - \mathbf{x}_t^p)(\mathbf{x} - \mathbf{x}_t^p)^T]$  as the covariance of the nominal belief along the nominal trajectory. Then the first part of the cost in Eq. (6) can be rewritten as  $\sum_{t=1}^K \text{tr}(\mathbf{W}_t \mathbf{P}_{b_t^p}^+ \mathbf{W}_t^T)$ . In [35], the evolution of  $\mathbf{P}_{b_t^p}^+$  is given by recursive Riccati equations independent of the observations but as a function of the trajectory itself. Instead of using the trace of covariance matrix as the uncertainty metric, to reduce the computational complexity, we propose replacing it with its upper bound, and we reformulate the optimal state estimation problem as an NLP problem in section IV-B.

In the next subsections, section IV-A gives the covariance upper bound that we applied. Corresponding to the nominal trajectory cost in Eq. (6), section IV-B describes the new trajectory planning problem based on upper bound and multiple resolution processes. Section IV-C shows our NLP solver for the planning problem based on a homotopy class constrained initial search and SQP. Section IV-D defines the LQR-based feedback controller for the nominal trajectory following. Section IV-E describes how to guarantee the stability of closed-loop system. Section IV-F defines the replanning criterion. Finally, we provide a simple computational complexity analysis of our solution framework.

#### A. UPPER BOUND OF THE COVARIANCE MATRIX

For the continuous model in Eq. (1), the linearized notation is:

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}_t &= \mathbf{A}_t^c \tilde{\mathbf{x}}_t + \mathbf{B}_t^c \tilde{\mathbf{u}}_t + \mathbf{G}_t \boldsymbol{\omega}_t \\ \tilde{\mathbf{z}}_t &= \mathbf{H}_t^c \tilde{\mathbf{x}}_t + \mathbf{F}_t \boldsymbol{\nu}_t,\end{aligned}\quad (7)$$

where superscript  $c$  is used to distinguish the matrices of the discrete model in Eq. (2), and the matrices in this model are obtained by expanding the model of Eq. (1) in a Taylor series around the nominal control sequence.

For this linearized model, the Kalman-Bucy filter is used for the state estimation. We have the following *Riccati differential equation or algebraic Riccati equation* and the trace of the equation.

$$\begin{aligned}\partial_t \mathbf{P}_t &= \mathbf{A}_t^c \mathbf{P}_t + \mathbf{P}_t \mathbf{A}_t^{cT} + \mathbf{G}_t \mathbf{Q}_t \mathbf{G}_t^T - \mathbf{P}_t \mathbf{S}_t \mathbf{P}_t, \\ \partial_t \text{tr}(\mathbf{P}_t) &= \text{tr}(\mathbf{A}_t^c \mathbf{P}_t + \mathbf{P}_t \mathbf{A}_t^{cT}) \\ &\quad + \text{tr}(\mathbf{G}_t \mathbf{Q}_t \mathbf{G}_t^T) - \text{tr}(\mathbf{P}_t \mathbf{S}_t \mathbf{P}_t),\end{aligned}\quad (8)$$

where  $\mathbf{S}_t = \mathbf{H}_t^{cT} (\mathbf{F}_t \mathbf{R}_t \mathbf{F}_t^T)^{-1} \mathbf{H}_t^c$ .

Based on the theory of logarithmic norms (for matrix  $\mathbf{A}$ ,  $\mu(\mathbf{A}) = \frac{1}{2} \lambda_{\max}(A + A^T)$ ), eigenvalue inequalities (for matrices  $\mathbf{A}, \mathbf{B}$ ,  $\lambda_{\min}(\mathbf{A}) \text{tr}(\mathbf{B}) \leq \text{tr}(\mathbf{AB}) \leq \lambda_{\max}(\mathbf{A}) \text{tr}(\mathbf{B})$ ), and Jensen's inequality, we obtained the following inequation:

$$\begin{aligned}\partial_t \text{tr}(\mathbf{P}_t) &\leq 2M \text{tr}(\mathbf{P}_t) + \text{tr}(\mathbf{G}_t \mathbf{Q}_t \mathbf{G}_t^T) \\ &\quad + [\lambda_{\min}(\mathbf{S})/n] \text{tr}(\mathbf{P}_t)^2,\end{aligned}\quad (9)$$

where  $M$  is the largest eigenvalue of system matrix  $\mathbf{A}_t$ .

The proofs in [38] show that the inequalities of solutions of differential equations can be deduced from differential inequalities ( $\partial_t x_t \leq \partial_t y_t$  implies  $x_t \leq y_t$ ). Therefore, the solution of the right-hand side of Eq. (9) will be the upper bound of  $\text{tr}(\mathbf{P}_t)$ , denoted as  $\text{tr}(\mathbf{P}_t)_{UB}$ , where the subscript

UB stands for *upper bound*. We use the maximum values of  $\mathbf{G}_t$  and  $\mathbf{F}_t$  as invariant spatially varying noises in  $\Delta t_u$  time in Fig. 1, which are denoted by  $\mathbf{G}$  and  $\mathbf{F}$ . According to the scalar Riccati differential equations as shown in Appendix -A, the upper bound is

$$P_{UB}(t) \triangleq \text{tr}(\mathbf{P}_t)_{UB} = \frac{b + \alpha}{a} + \frac{2\alpha}{a} \frac{\beta e^{-2\alpha t}}{1 - \beta e^{-2\alpha t}}, \quad (10)$$

where

$$\begin{aligned}\alpha &= (a, b, c) = \alpha(\lambda_{\min}(\mathbf{S})/n, M(f), \text{tr}(\mathbf{G} \mathbf{Q} \mathbf{G}^T)), \\ \beta &= (a, b, c, x_0) \\ &= \beta(\lambda_{\min}(\mathbf{S})/n, M(f), \text{tr}(\mathbf{G} \mathbf{Q} \mathbf{G}^T), \text{tr}(\mathbf{P}_0)).\end{aligned}\quad (11)$$

The upper bound of the trace of the covariance defined in Eq. (10) is a continuous function. It can be easily discretized into any resolution and avoids recursively solving Riccati equations.

#### B. PLANNING PROBLEM BASED ON UPPER BOUND

Suppose that the time for a robot moving from the start position to the goal position is  $T$ . Based on the resolution settings in Fig. 1, the underlying optimal trajectory is parameterized by  $K = \frac{T}{\Delta T}$  segments. In each segment, the belief propagation rate is  $\delta = \frac{1}{\Delta t_{KF}}$ . We assume that the control in one segment is a constant value and that the trajectory in this period will be a straight line. Another form of the trajectory (spline curve and polynomial trajectory) can also be adapted to this problem formulation. Therefore, based on the upper bound and multi-resolution assumptions, we reformulate the cost in Eq. (6) and define the following planning problem:

**Problem 2. Trajectory planning:** Given an initial belief  $\mathbf{b}_0$ , goal state  $\mathbf{x}_g$ , its tolerance  $r_g$ , and obstacle definition  $\mathcal{O}_{obstacle}$ , we obtained the following NLP problem:

$$\begin{aligned}\min_{\mathbf{u}_0^p: K-1} \quad & \sum_{t=1}^K \sum_{i=t\delta}^{t\delta+\delta-1} [(P_{UB})_i] + (\mathbf{u}_{t-1}^p)^T \mathbf{W}_t^u \mathbf{u}_{t-1}^p + w_2 \mathcal{C}_{obs} \\ \text{s.t.} \quad & P_{UB}(t) = \frac{b + \alpha}{a} + \frac{2\alpha}{a} \frac{\beta e^{-2\alpha t}}{1 - \beta e^{-2\alpha t}} \\ & \mathbf{x}_0^p = \mathbb{E}[\mathbf{b}_0(\mathbf{x})] \\ & \mathbf{x}_{t+1}^p = f(\mathbf{x}_t^p, \mathbf{u}_t^p, 0), \quad 0 \leq t \leq K-1 \\ & \|\mathbf{x}_K^p - \mathbf{x}_g\|_2 < r_g \\ & \|\mathbf{u}_t^p\|_2 < r_u, \quad 0 < t < K,\end{aligned}\quad (12)$$

where  $w_1$ ,  $\mathbf{W}_t^u$  and  $w_2$  are positive-definite weight values. The first constraint in Eq. (12) is the upper bound of the trace of covariance matrix as defined in Eq. (10); the second and third equations are the initial condition and motion constraint; and the last two constraints are for confining the state and control to within their limits.

As opposed to using hard constraint on the collision avoidance, we apply a soft strategy by using an obstacle cost term  $\mathcal{C}_{obs}$  as shown in the objective function in equation (12). To define this cost, we use a sphere bounding strategy for the AUVs geometry description [39], [40]. Let  $S_j (j = 1, \dots, M)$

stand for the spheres we defined, the center and radius of the sphere are  $(x_j, y_j, z_j)$  and  $R_j$ , respectively. For each trajectory and state, we have the following definition of the collision cost functional  $\mathcal{C}_{obs}$  and cost function  $c(d)$ :

$$\mathcal{C}_{obs} = \sum_{t=0}^K \sum_{j=1}^M [c(d(\mathbf{x}_k, S_j))], \quad (13)$$

$$c(d) = \begin{cases} -d + \frac{1}{2}\varepsilon & \text{if } d < 0 \\ \frac{1}{2\varepsilon}(d - \varepsilon)^2\varepsilon & \text{if } 0 \leq d \leq \varepsilon \\ 0 & \text{if } d > \varepsilon \end{cases}, \quad (14)$$

where  $\varepsilon$  is the safety threshold and  $d$  is the distance from the center of  $S_j$  to the nearest obstacle surface.

In this work, the state of the robot has Gaussian uncertainty. To guarantee the safety of motion and avoid collision at the 99% confidence level, the distance  $d$  is amplified to  $d' = d + 2 \times 3.3682 \times \lambda_1(\Sigma)$ , where  $\lambda_1$  is the major axis of the 99% confidence ellipsoid [41].

### C. PLANNING NOMINAL TRAJECTORIES

#### 1) Homotopy constrained initial trajectory search

The solution of the NLP problem is sensitive to the initial values. To obtain an optimal solution, we propose to generate multiple initial trajectories belonging to different homotopy classes and then select the optimal trajectory with respect to the cost function defined in Eq. (12). For real-time system deployment, this process can be performed in a parallel manner using a multicore processor. The computational cost of separating the configuration space and solving one trajectory in every homotopy class is very high. Therefore, we use the fast random search method and then select multiple paths belonging to different classes. To find multiple initial trajectories, two steps are performed: 1. find a collision-free trajectory and 2. check whether it belongs to a new homotopy class.

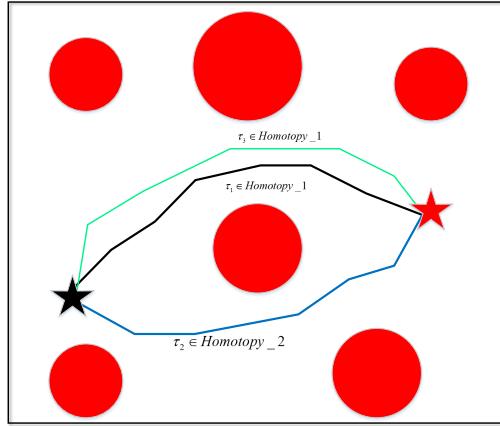
For the first step, we use RRT-connect as the collision-free path planner. By running multiple times, a number of random paths are generated. RRT-connect generates one path in a very short time without considering the optimality. We reconnected the RRT-connect path by reducing the length of the path. A path is then represented by the set of waypoints composing a rewired path.

For homotopy class checking, we use  $\mathcal{H}(\tau)$  proposed in [17] as the  $H$ -signature of a trajectory  $\tau$ .

$$\mathcal{H}(\tau) = \int_{\tau} \left[ \begin{array}{c} \mathbf{B}_1(l) \\ \mathbf{B}_2(l) \\ \vdots \\ \mathbf{B}_3(l) \end{array} \right] dl \quad (15)$$

$$\mathbf{B}_i(r) = \frac{1}{4\pi} \int_{S_i} \frac{(\mathbf{x} - \mathbf{r}) \times d\mathbf{x}}{\|\mathbf{x} - \mathbf{r}\|^3},$$

where  $S_i$  denotes the skeletons of the  $i$ -th obstacle,  $\mathbf{x}$  is the point on  $S_i$  and  $\mathbf{r}$  denotes a position on the path. As an example in Fig. 3, paths  $\tau_1$  and  $\tau_3$  belong to the same homotopy class, where  $\mathcal{H}(\tau_1) = \mathcal{H}(\tau_3) \neq \mathcal{H}(\tau_2)$ .



**FIGURE 3:** The initial trajectories in different homotopy classes. The trajectories  $\tau_1$ ,  $\tau_2$ , and  $\tau_3$  have the same start (black star) and goal position (red star).  $\tau_3$  can be continuously transformed into  $\tau_1$ ; therefore, they belong to the same homotopy class (*Homotopy\_1*).  $\tau_2$  belongs to another homotopy class *Homotopy\_2*.

For real-time system deployment, trajectory planning can be processed in a parallel manner with respect to the sampled initials.

#### 2) SQP for NLP problem

In this paper, we use SQP as the solver, which is one of the most effective approaches for constrained NLP problems [42] and provides a locally optimal solution. Homotopy class constrained initial trajectory search generate a set of trajectories  $\{\tau_1, \tau_2, \dots, \tau_N\}$ . Then, we solve the NLP problem defined in Eq. (12) with those initials. Finally, a near-optimal solution  $\{\mathbf{u}_{0:K-1}^*, \mathbf{x}_{0:K}^*\}$  is obtained by comparing the cost function values of the locally optimal solutions.

In our parameter settings, to guarantee safety, the weight of the collision cost must be higher than that of other costs. For faster and more optimal optimization, a dynamic weight scheduling process also can be applied in practical applications [39].

### D. LQR FEEDBACK CONTROLLER

Solution of **Problem 2** is a nominal trajectory with the consideration of estimation performance, control effort and collision avoidance. The feedback controller is designed for the trajectory tracking control. In this stage, based on the separation principle, we use LQG as the feedback controller, which has a control law similar to LQR. First, the system will be linearized around the optimized nominal trajectory  $\{\mathbf{u}_{0:K-1}^*, \mathbf{x}_{0:K}^*\}$  based on Eq. (2). Then we define the following control cost:

$$J^f := \sum_{t=1}^K \left[ (\hat{\mathbf{x}}_t - \mathbf{x}_t^*)^T \mathbf{W}_t^x (\hat{\mathbf{x}}_t - \mathbf{x}_t^*) + (\tilde{\mathbf{u}}_{t-1}^*)^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1}^* \right], \quad (16)$$

where  $\mathbf{W}_t^u$  and  $\mathbf{W}_t^x$  are the weight matrices and  $\hat{\mathbf{x}}_t$  is the estimated state obtained with the KF.

Then we obtain the optimal linear feedback control policy

$$\tilde{\mathbf{u}}_t^* = \mathbf{u}_t - \mathbf{u}_t^* = -\mathbf{L}_t (\hat{\mathbf{x}}_t - \mathbf{x}_t^*) \quad (17)$$

The feedback gain  $\mathbf{L}_t$  is defined by:

$$\mathbf{L}_t = \left( \mathbf{W}_t^u + (\mathbf{B}_t^*)^T \mathbf{P}_t^f \mathbf{B}_t^* \right)^{-1} (\mathbf{B}_t^*)^T \mathbf{P}_t^f \mathbf{A}_t^*, \quad (18)$$

where  $\mathbf{A}_t^*$  and  $\mathbf{B}_t^*$  are linearized system matrices, and  $\mathbf{P}_t^f$  is solved via backward iteration with the terminal condition  $\mathbf{P}_K^f = \mathbf{W}_t^x = w_1 \cdot \mathbf{I}$ :

$$\begin{aligned} \mathbf{P}_{t-1}^f &= (\mathbf{A}_t^*)^T \mathbf{P}_t^f \mathbf{A}_t^* - (\mathbf{A}_t^*)^T \mathbf{P}_t^f \mathbf{B}_t^* \left( \mathbf{W}_t^u + (\mathbf{B}_t^*)^T \mathbf{P}_t^f \right)^{-1} \\ &\quad \cdot (\mathbf{B}_t^*)^T \mathbf{P}_t^f \mathbf{A}_t^* + \mathbf{W}_t^x. \end{aligned} \quad (19)$$

### E. CLOSED LOOP SYSTEM STABILITY ANALYSIS

Although the limitation of nominal control  $\mathbf{u}_t^*$  is considered in planning stage (the last constraint in Eq. (12)), the feedback control  $\mathbf{u}_t$  in Eq. (17) may move in and out of control saturation. Therefore, the stability of the closed-loop system must be guaranteed under control saturation. When control saturation happens, we use

$$\mathbf{u}_t' = sat(\mathbf{u}_t) = \begin{cases} \mathbf{u}_{\max} & \text{if } \mathbf{u}_t \geq \mathbf{u}_{\max} \\ \mathbf{u}_t & \text{if } \mathbf{u}_{\min} < \mathbf{u}_t < \mathbf{u}_{\max} \\ \mathbf{u}_{\min} & \text{if } \mathbf{u}_t \leq \mathbf{u}_{\min} \end{cases}, \quad (20)$$

as the feedback control, which means that the control will equal to saturation value when  $\mathbf{u}_t$  in Eq. (17) moves in or out of saturation.

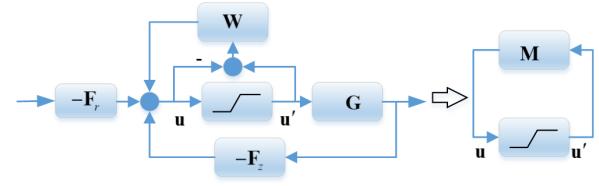
For linear time-invariant closed-loop LQG control system with system parameters  $\mathbf{A}, \mathbf{B}, \mathbf{G}, \mathbf{H}, \mathbf{F}$  (Eq. (7)), state estimation gain  $\mathbf{K}$  (Eq. (29)) and feedback control gain  $\mathbf{L}$  (Eq. (17)), the closed-loop system model can be rewrite as :

$$\begin{aligned} \dot{\hat{\mathbf{x}}}_t &= \mathbf{A}\hat{\mathbf{x}}_t + \mathbf{B}\mathbf{u}_t + \mathbf{K}(\mathbf{z}_t - \hat{\mathbf{z}}_t) = \\ &= (\mathbf{A} - \mathbf{KH})\hat{\mathbf{x}}_t + \mathbf{B}(\mathbf{u}_t + \mathbf{u}'_t - \mathbf{u}_t) + \mathbf{K}(\mathbf{z}_t - \hat{\mathbf{z}}_t) \\ &= (\mathbf{A} - \mathbf{BL} - \mathbf{KH})\hat{\mathbf{x}}_t + \mathbf{K}\mathbf{z}_t + \mathbf{B}(\mathbf{u}'_t - \mathbf{u}_t). \end{aligned}$$

Taking Laplace transformation, we obtain

$$\begin{aligned} \mathbf{U}(s) &= -\mathbf{L}(s\mathbf{I} - \mathbf{A} + \mathbf{BL} + \mathbf{KH})^{-1} \mathbf{KZ}(s) \\ &\quad - \mathbf{L}(s\mathbf{I} - \mathbf{A} + \mathbf{BL} + \mathbf{KH})^{-1} \mathbf{B}(\mathbf{U}'(s) - \mathbf{U}(s)) \\ &= -\mathbf{C}(s)\mathbf{Z}(s) + \mathbf{W}(s)(\mathbf{U}'(s) - \mathbf{U}(s)). \end{aligned}$$

This equation can be denoted by the block diagram in Fig. 4. Based on the small-gain theorem [43], if the  $H_\infty$  norm  $\|\mathbf{M}\|_\infty < 1$ , the stability of the close-loop system can be ensured, where  $\mathbf{M} = \mathbf{W} - \mathbf{GC}$ , and  $\mathbf{G}$  is the system transfer function of linear system (Eq. (7)) in frequency domain. This criteria is a guidance of the controller parameters setting in Eq. (16).



**FIGURE 4:** Block diagram for small-gain analysis

### F. REPLANNING PROCESS

In this work, we assume a linear stochastic system with Gaussian disturbance. We first obtain an trajectory with optimal estimation performance, and then use an optimal controller for the trajectory tracking. However, the trajectory tracking result based on this controller may deviate from the nominal trajectory during real-time trajectory tracking due to the linearization error and un-modeled noise (non-Gaussian disturbance, system model parameters error, unpredicted or unknown external disturbances or forces). There are many robust trajectory tracking control methods for underwater robots to guarantee the robustness of the trajectory tracking control under unknown disturbance, such as disturbance-observer-based methods [44]–[46]. In our future work, we will consider how to integrate those robust controllers into the nominal trajectory planning to improve the performance of control.

In our approach, replanning will be performed when the distance between the current belief ( $\mathbf{b}_t = \mathcal{N}(\mathbf{x}_t, \mathbf{P}_t)$ ) and the nominal belief ( $\mathbf{b}_t^* = \mathcal{N}(\mathbf{x}_t^*, \mathbf{P}_t^*)$ ) exceeds the defined threshold ( $d_{th}$ ) as shown in Fig. 8(b). There are many methods for measuring the distance of two probability distributions such as the Bhattacharyya distance, the Kullback–Leibler divergence and the Hellinger distance. In this work, we select the symmetric Bhattacharyya distance as the deviation measurement [47]

$$D_{\mathbf{b}_t^*, \mathbf{b}_t} = \frac{1}{8} (\mathbf{x}_t^* - \mathbf{x}_t)^T \mathbf{P} (\mathbf{x}_t^* - \mathbf{x}_t) + \frac{1}{2} \ln \left( \frac{\det \mathbf{P}}{\sqrt{\det \mathbf{P}_t^* \det \mathbf{P}_t}} \right).$$

The complete solution framework is shown in Algorithm 1. The first part solves the optimal nominal trajectory and feedback control gains. During task execution, if the deviation exceeds threshold ( $D_{\mathbf{b}_t^*, \mathbf{b}_t} > d_{th}$ ) or time has run out, replanning will be performed.

### G. COMPUTATIONAL COMPLEXITY ANALYSIS

In the motion planning process, the obstacle cost calculation has higher computational complexity than other parts. Therefore, we consider only the computational complexity of the obstacle-free case. Based on the complexity analysis in [18], [19], the cost mainly results from the non-linear covariance matrix update. There is a detailed comparison of computational complexity of state-of-the-art belief space motion planners in [35]. The approach proposed in [35] has less computational complexity ( $\mathcal{O}(Kn^3)$ ) than others. Compared with this method, our current formulation reduces the updating process by retrieving the upper bound of the trace

---

**Algorithm 1** Closed-loop AUV motion planning approach

**Input:** Initial belief  $\mathbf{b}_0$ , goal state  $\mathbf{x}_g$  and tolerance  $r_g$ , planning time horizon  $K$ , map  $\mathcal{M}$

**Output:** Control policy  $\pi$  for AUV task execution

$$(\mathbf{u}_{0:K-1}^*, \mathbf{b}_{0:K}^*) \leftarrow \text{solveOptimalTraj}(\mathbf{J}, \mathbf{b}_0, \mathbf{x}_g, K, \mathcal{M})$$

$$\mathbf{L}_{0:K-1} \leftarrow \text{solveLQR}(\mathbf{J}^f, \mathbf{u}_{0:K-1}^*, \mathbf{b}_{0:K}^*)$$

**while**  $\|\mathbf{x}_t - \mathbf{x}_g\|_2 > r_g$  **do**

- if**  $t < K$  **&&**  $d(\mathbf{b}_t, \mathbf{b}_t^*) < d_{th}$  **then**

  - $\hat{\mathbf{x}}_t \leftarrow \mathbb{E}[\mathbf{b}_t]$ ,  $\mathbf{u}_t \leftarrow -\mathbf{L}_t(\hat{\mathbf{x}}_t - \mathbf{x}_t^*) + \mathbf{u}_t^*$ ;
  - Task Execution:  $\mathbf{x}_{t+1} \leftarrow f(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\omega}_t)$
  - State Observation:  $\mathbf{z}_{t+1} \leftarrow h(\mathbf{x}_{t+1}, \boldsymbol{\nu}_{t+1})$ ;
  - Belief Propagation:  $\mathbf{b}_{t+1} \leftarrow \tau(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_{t+1})$ ;
  - $t \leftarrow t + 1$ ;

**end**

*/\*Replaning process\*/*

**if**  $t \leq K$  **&&**  $d(\mathbf{b}_t, \mathbf{b}_t^*) > d_{th}$  **then**

- if**  $t == K$  **then**

  - $| t \leftarrow 0; K' \leftarrow K;$

**end**

**else**

- $| \mathbf{b}_0 \leftarrow \mathbf{b}_t, K' \leftarrow K - t;$

**end**

$(\mathbf{u}_{t:K-1}^*, \mathbf{b}_{t:K}^*) \leftarrow$   
 $\text{solveOptimalTrajectory}(\mathbf{J}, \mathbf{b}_0, \mathbf{x}_g, K', \mathcal{M})$

$$\mathbf{L}_{t:K-1} \leftarrow \text{solveLQR}(\mathbf{J}^f, \mathbf{u}_{0:K'-1}^*, \mathbf{b}_{0:K}^*)$$

$$\hat{\mathbf{x}}_t \leftarrow \mathbb{E}[\mathbf{b}_t]$$
,  $\mathbf{u}_t \leftarrow -\mathbf{L}_t(\hat{\mathbf{x}}_t - \mathbf{x}_t^*) + \mathbf{u}_t^*$ ;

Task Execution:  $\mathbf{x}_{t+1} \leftarrow f(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\omega}_t)$

State Observation:  $\mathbf{z}_{t+1} \leftarrow h(\mathbf{x}_{t+1}, \boldsymbol{\nu}_{t+1})$ ;

Belief Propagation:  $\mathbf{b}_{t+1} \leftarrow \tau(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_{t+1})$ ;

$t \leftarrow t + 1$ ;

**end**

**end**

---

**Algorithm 2** *solveOptimalTraj* compute optimal nominal trajectory planning

**Input:** Initial belief  $\mathbf{b}_0$ , goal state  $\mathbf{x}_g$  and tolerance  $r_g$ , planning time horizon  $K$ , map  $\mathcal{M}$

**Output:** Optimal nominal trajectory with low uncertainty  
 $(\mathbf{u}_{1:K}^*, \mathbf{b}_{1:K}^*)$

1. Generate  $N$  initial trajectories belong to different homotopy class:
$$(\mathbf{u}_{1:K}^N, \mathbf{b}_{1:K}^N) \leftarrow \text{RRT-connect}(\mathbf{x}_0, \mathbf{x}_g, \mathcal{M})$$
  2. Nominal trajectory planning (**problem 1**):
$$(\mathbf{u}_{1:K}^*, \mathbf{b}_{1:K}^*) \leftarrow \text{solveSQP}(\mathbf{J}, \mathbf{b}_0, \mathbf{x}_g, K, \mathbf{u}_{1:K}^N, \mathcal{M})$$
- 

of the covariance from the continuous-time function, and we also reduce the optimization dimension of the NLP problem via multiple resolution framework. Therefore our method has lower computational complexity (less than  $\mathcal{O}(Kn^3)$ ). The convergence rate of the non-linear programming problem defined in (12) is equal to the convergence rate of the optimization method applied. For Newton's method, the converge rate is second-order.

## V. SIMULATIONS AND RESULTS

In this section, we test our approach on three types of AUVs, including holonomic AUVs with linear dynamics, torpedo-shaped non-holonomic AUVs with non-linear dynamics, and high-dimensional I-AUV navigation. The motion planning scenarios involve motion and observation with state-dependent noise and spatially varying sensing capabilities. The goal of the motion planning task is to obtain a collision-free trajectory with low localization uncertainty in cluttered underwater environments.

In the next subsections, we first show the simulation of the relationship between the covariance and the upper bound. Then we demonstrate and analyze the planning and task execution results (trajectories and computational cost) of three categories of underwater robots in different scenarios. All simulations are conducted in MATLAB2016b on a computer with a 2.8 GHz Intel Core i7-7700HQ CPU.

### A. COMPARISON BETWEEN THE COVARIANCE AND UPPER BOUND

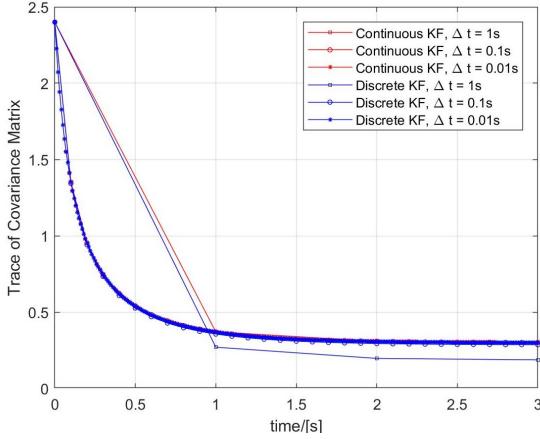
We first show the equivalence between trace from discrete covariance updating and the continuous-time upper bound. Consider the following simple three-dimensional linear velocity process model and full state observation model:  $\dot{\mathbf{x}}_t = \mathbf{v}_t + \mathbf{G}_t \boldsymbol{\omega}_t$ ,  $\mathbf{z}_t = \mathbf{x}_t + \mathbf{F}_t \boldsymbol{\nu}_t$ , the system can be discretized with timestep  $\Delta t$ . As shown in Fig. 5, we compared the value of the trace of covariance obtained by discrete updating and upper bound updating and showed that the difference is smaller with a small discrete timestep. In this case, the noise is isotropic Gaussian noise, and the solution of the upper bound function is the same as the solution of the differential Riccati equation.

The result in Fig. 6 shows that for non-isotropic noise, the upper bound of the trace covariance is always greater than that of the normal trace; however, they have the same variation trend. To obtain a smooth uncertainty cost, the discrete timestep should be small. However, this will introduce a high of computational cost due to the updating of the Riccati equations.

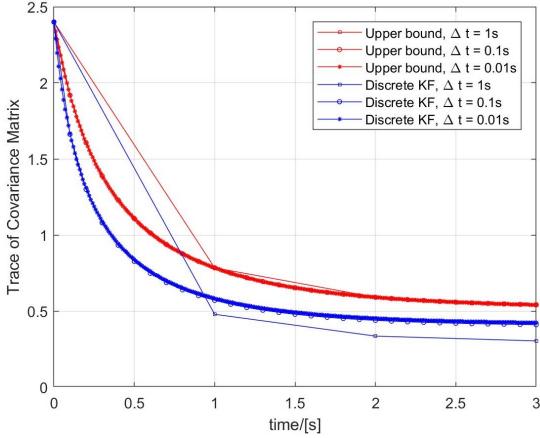
### B. PLANNING UNDER VARIOUS SPATIALLY VARYING NOISE

#### 1) Holonomic AUVs

For general holonomic AUV robots, the kinematics equations can be written as  $\dot{\mathbf{x}}_t = \mathbf{v}_t + \mathbf{G}_t \boldsymbol{\omega}_t$ ,  $\mathbf{z}_t = \mathbf{x}_t + \mathbf{F}_t \boldsymbol{\nu}_t$ . The state can be described by the configuration of the vehicle,  $\mathbf{x} = [x, y, z]^T$ , which denotes the position of the robot. For the multiple resolution setting in Fig. 1, we set  $\Delta t_\tau = \Delta t_u = 1$ ,  $\Delta t_{KF} = 0.01$ , and  $\Delta t_{obst} = 0.5$ . In underwater environments, the process noise distribution is related to the water flow, and the observation noise is related to the localization sensors. As shown in Fig. 7 and Fig. 8, the proposed planning method is tested in an obstacle populated environment under various spatially varying processes or observation noise (cases 1, 2, and 3). The noise weights are a function of one ( $y$ ) or two dimensions ( $x, y$ ) of the pose of



**FIGURE 5:** Comparison of the trace of covariance matrix solved from the discrete KF and upper bound of the trace of covariance matrix. *Parameters:* In Eq. (7), the noise weight matrix  $\mathbf{G}_t = \mathbf{F}_t = \mathbf{I}$ . To discretize the system,  $\Delta t_\tau = \Delta t_u = \Delta t_{KF} = \Delta T$ ,  $\Delta T = \{1, 0.1, 0.01\}$ . The noise covariance  $\mathbf{Q}_t = \mathbf{R}_t = diag([0.1; 0.1; 0.1])$ .



**FIGURE 6:** Comparison of the trace of covariance matrix solved using the discrete KF and the upper bound of the trace of the covariance matrix. *Parameters:* In Eq. (7), noise weight matrix  $\mathbf{F}_t = \mathbf{I}$ . To discretize the system,  $\Delta t = \Delta T$ ,  $\Delta T = \{1, 0.1, 0.01\}$ . The noise covariance  $\mathbf{Q}_t = diag([0.1; 0.1; 0.1])$ ,  $\mathbf{R}_t = diag([0.1; 0.3; 0.2])$ .

the robot. For all scenarios, the task and parameter settings are shown in the caption of the figures. As clearly shown, the robot always attempts to spend more time in a low-noise area to obtain a better control accuracy or sensor measurement and reduce the state uncertainty.

The approach proposed in this work is based on the separation principle of LQG. However, when we consider the uncertainty of model parameters, the separation principle will be invalid [48]. Therefore, there is no theoretical analysis of the effect of model parameters uncertainty and this uncertainty is not considered in the planning stage. To test the robustness of our approach, in trajectory tracking control

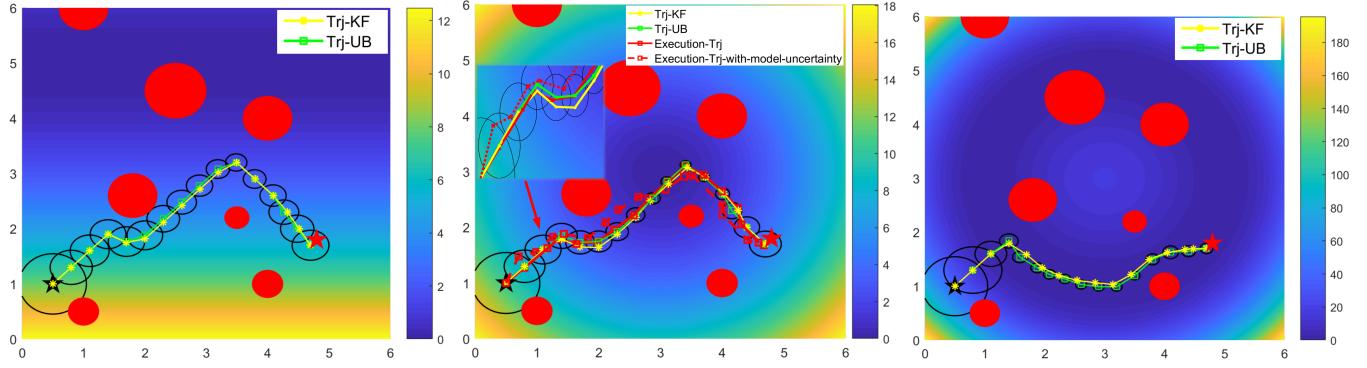
stage, we use 10% nominal values of the system model parameters  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{H}$  in Eq. (2) as the added model parameters uncertainty. As an example shown in Fig. 7(b), although the tracking result (Execution-Trj-with-model-uncertainty) is not perfect compared with the result without considering model uncertainties (Execution-Trj), the nominal optimal trajectory still can be tracked.

In Fig. 8(a), the two initial trajectories (Initi-Trj-1 and Initi-Trj-2) solved by RRT-connect belong to two homotopy classes. The optimized trajectories (Trj-UB-1 and Trj-UB-2) are local minimum solutions based on the initials. In task execution, we will select one solution with the minimum cost (Trj-UB-1).

In Fig. 8(b), Trj-UB illustrates the nominal optimal trajectory. When  $t = 2s$ , the Bhattacharyya distance between the current and nominal state exceeds the threshold due to the effect of unmodeled external disturbance. The green dashed curve (Replanned-Trj-UB) denotes the replanned trajectory, and the red dotted curve is the execution trajectory obtained by tracking the replanned trajectory with the feedback control defined in section IV-D.

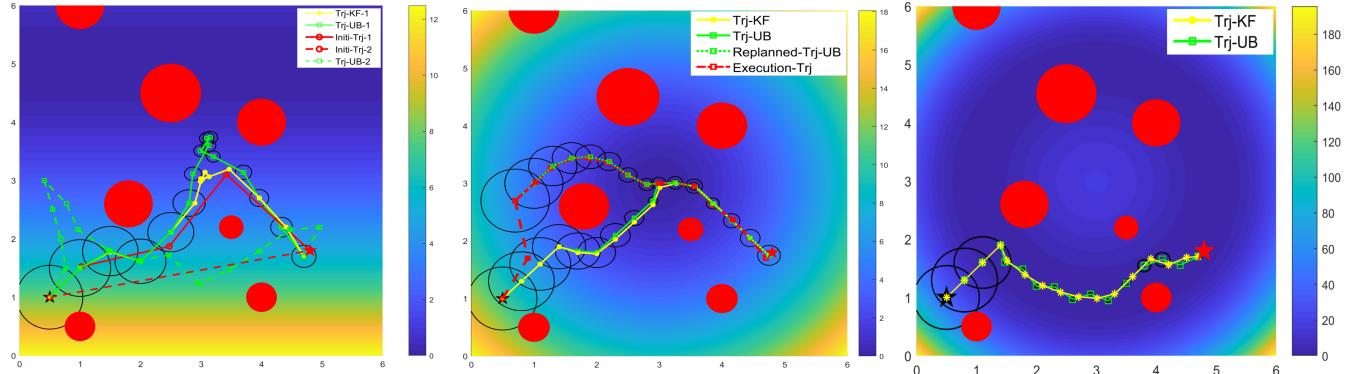
The average computational time for one nominal trajectory optimization (not including initial trajectories search) and the change in the objective function value are reported in Table 1. We compared the computational time with the method proposed in [35], which use a KF to propagate the state belief in the planning stage. In this table, UB denotes the method that we proposed;  $\mathbf{G}_t$  and  $\mathbf{F}_t$  denote whether the system has spatially varying process noise or observation noise. The noise distribution functions are below the figures. Test results show that SQP solver can converge after 30 iterations in most cases. Moreover, the number of iterations of the upper bound-based optimization is less than 30 in most cases. Therefore, for a fair comparison, we restrict the maximum SQP iteration times to 30. As reported in Table 1, the filtering-based optimization is approximately five times faster than the upper bound-based optimization.

As shown in Fig. 6, the uncertainty value denoted by the upper bound is higher than the trace of covariance. To compare the performance of optimized trajectory, the uncertainty costs of initial and optimized trajectory in Table 1 are transform into the trace of covariance. The initial cost is the objective function value of a straight-line initial (like the "Initi-Trj-2" in Fig. 8(a)). In all cases, the optimized cost function values are very close, which is consistent with the results shown in Fig. 7 and Fig. 8 (Trj-UB and Trj-KF are very close in each figure). Therefore, we clearly see that the framework proposed in this work is more efficient.



(a) Case 1: The optimized low uncertainty trajectories under spatially-varying process noise  $0.5 * (x - 3)^2 + (y - 3)^2 + 0.01$ .  
(b) Case 2: The optimized low uncertainty trajectories under spatially-varying process noise  $((x - 3)^2 + (y - 3)^2 + 0.1)$ .  
(c) Case 3: The optimized low uncertainty trajectories under spatially-varying process noise  $((x - 3)^2 + (y - 3)^2 - 4)^2$ .

**FIGURE 7:** Optimized trajectory under different spatially-varying *process noise*. The black and red pentagram  $\star$  denote the start ( $[0.5, 1, 2]^T$ ) and goal position ( $[4.8, 1.8, 2]^T$ ). In all figures, the green trajectories are planned based on upper bound theory, and the yellow trajectories are solved based on KF. The black circles show the beliefs of the nodes on the optimized trajectory. In (b), the two red curves denote the execution trajectories with (Execution-Trj with model uncertainty) or without (Execution-Trj) model uncertainties ( $\Delta\mathbf{A}, \Delta\mathbf{B}, \Delta\mathbf{C}$ ). Noise covariance  $\mathbf{Q}_t = \text{diag}([0.01; 0.01; 0.01])$ ,  $\mathbf{R}_t = \text{diag}([0.03; 0.02; 0.01])$ . The



(a) Case 1: The optimized low uncertainty trajectories under spatially-varying observation noise  $0.5 * (5 - y)^2 + 0.01$ .  
(b) Case 2: The optimized low uncertainty trajectories under spatially-varying observation noise  $((x - 3)^2 + (y - 3)^2 + 0.1)$ .  
(c) Case 3: The optimized low uncertainty trajectories under spatially-varying observation noise  $((x - 3)^2 + (y - 3)^2 - 4)^2$ .

**FIGURE 8:** Optimized trajectory under different spatially varying *observation noise*. The noise covariance  $\mathbf{Q}_t = \text{diag}([0.01; 0.01; 0.01])$ ,  $\mathbf{R}_t = \text{diag}([0.03; 0.02; 0.01])$ .

## 2) Non-holonomic AUVs

The simplified kinematics model of the non-holonomic torpedo-shaped AUV is defined by

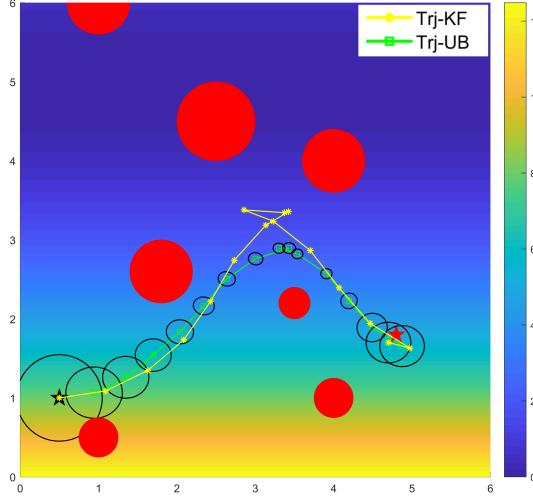
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v \cdot \cos(\psi) \\ v \cdot \sin(\psi) \\ v_z \\ w \end{bmatrix} + \mathbf{G}_t[\mathbf{p}] \cdot \boldsymbol{\omega}_t \quad (21)$$

$$\mathbf{z}_t = \mathbf{x}_t + \mathbf{F}_t[\mathbf{p}] \cdot \boldsymbol{\nu}_t,$$

where the robot state  $\mathbf{x} = [\mathbf{p}, \psi]^T$ ,  $\mathbf{p} = [x, y, z]$  denotes the position and  $\psi$  is the heading angle. The control input includes the forward and vertical velocity and steering angle

velocity  $\mathbf{u} = [v, v_z, \omega]^T$ .

Under the same uncertainty distribution cases shown in Fig. 7 and 8, Fig. 9 shows the result of one case. The robot spends a substantial amount of time at a location with low noise to reduce the state uncertainty before moving to the goal position. The two trajectories obtained from the KF (yellow curve) and the upper bound-based (green curve) optimization have similar uncertainties. However, as reported in Table 1, the computational time for planning method based on the KF is almost five times greater than our method for all cases, and the cost function value change shows that the performance of the results is similar.



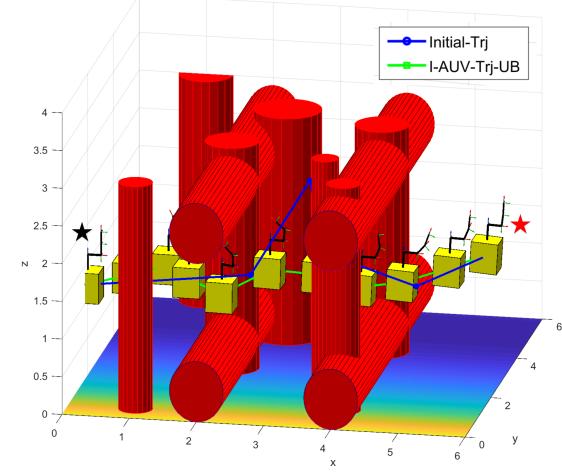
**FIGURE 9:** The optimized low uncertainty trajectory for non-holonomic AUV under spatially-varying process noise  $0.5 * (5 - y)^2 + 0.01$ .

### 3) I-AUV

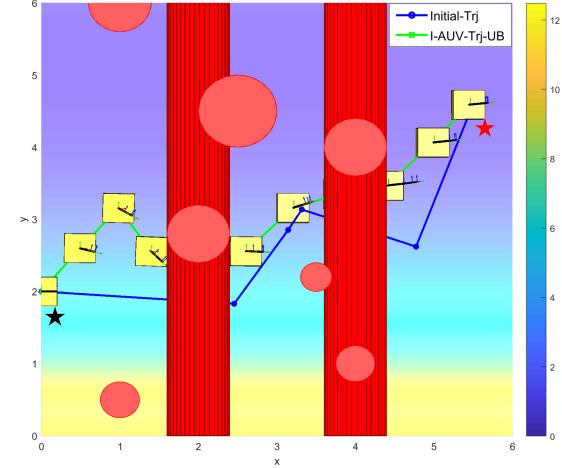
We consider the case of high-dimensional I-AUV systems moving in cluttered underwater environment. As described in our earlier work [22], the simulated I-AUV is illustrated in Fig. 10. It has a 3-link manipulator at the top of the base. The I-AUV generalized coordinates are defined as  $\mathbf{x} = [\mathbf{P}^B{}^T, \Psi^B{}^T, \boldsymbol{\theta}^M{}^T]^T$ , where  $\mathbf{P}^B = [x, y, z]^T$  denotes the position, and  $\Psi^B = [\theta_{pitch}, \gamma_{roll}, \psi_{yaw}]^T$  is the attitude. Its generalized velocities are denoted as  $\dot{\mathbf{x}} = [\mathbf{v}^B{}^T, \omega^B{}^T, \omega^M{}^T]^T$ ,  $\boldsymbol{\theta}^M = [\theta_1, \theta_2, \theta_3]^T$ ,  $\boldsymbol{\omega}^M = [\omega_{\theta_1}, \omega_{\theta_2}, \omega_{\theta_3}]^T$ . The control input is the desired velocities  $\mathbf{u} = \dot{\mathbf{x}}$ . Specifically, the kinematic equations of motion are:  $\mathbf{x}_{k+1} = \mathbf{x}_k + \tau \mathbf{v}_k + M[\mathbf{P}_k] \cdot \boldsymbol{\omega}_t$ ,  $\mathbf{z}_k = \mathbf{x}_k + N[\mathbf{P}_k] \cdot \boldsymbol{\nu}_t$ , where  $\tau$  is the size of the discretized timestep, and  $M[\mathbf{P}_k]$  and  $N[\mathbf{P}_k]$  scale the motion and observation noise, respectively, being proportional to the robot's state.

Compared with the framework that we proposed in [22] and according to the multiple resolution setting, we reduced the dimensions of the optimization problem with a low control resolution ( $\Delta t_u = 1s$ ). The upper bound function also avoids the Riccati equation updating in  $\Delta t_u$  time. Figs. 10 and 11 show an optimal trajectory of one case. The initial trajectories of the I-AUV are obtained by the methods described in our earlier work [22]. The computational time in Table 1 shows that optimization based on upper bound achieves a lower computational time. Under the same problem settings, method in [22] takes approximately 600s to solve the NLP problem in Fig. 10. To guarantee a collision-free trajectory for the manipulator of an I-AUV, we set  $\Delta t_{obst} = 0.01s$  in Fig. 1; then, the collision check accounts for most of the computational cost [49]. Therefore, the difference in the computational time for the I-AUV is not obvious; however, it is still 50% faster on average when the upper bound-based optimization is employed. As described in Table 1, we also

simulated other cases, and the advantages of the upper bound-based planning remain.



**FIGURE 10:** I-AUV 3D planning trajectory in cluttered environment under spatially varying observation noise  $0.5 * (5 - y)^2 + 0.01$ . In this case, the start state:  $\mathbf{x}_0 = [0; 2; 1.2; 0; 0; 0; -\pi/2; 0]$ , and goal state is  $\mathbf{x}_g = [5.5; 4.6; 1.2; 0; 0; 0; -\pi/2; 0]$ . A supplementary video clearly shows the motion trajectory of I-AUV: <https://youtu.be/Z1yBgDJgGl8>.



**FIGURE 11:** I-AUV 3D trajectory planning in cluttered environment. XoY view of Fig. 10.

## VI. CONCLUSIONS AND FUTURE WORK

This paper presented a fast motion planning approach in belief space for underwater robots in cluttered environments. Based on multiple resolution processes and the upper bound of trace of the covariance matrix, we proposed a trajectory planning approach under the LQG framework, which solves a low uncertainty and collision-free trajectory in environments with motion and observation noises. Numerical simulations on three types of underwater robots showed that in various noise scenarios the proposed trajectory optimization

**TABLE 1:** Comparison of computational time and changes in the objective function value.

			Computation time (sec.)			Initial cost ( $\times 10^2$ )			Optimized cost ( $\times 10^2$ )		
			Case 1	Case 2	Case 3	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
Holonomic AUVs	$G_t$	UB	<b>4.52</b>	<b>6.35</b>	<b>6.35</b>	4.35	2.92	4.19	2.60	1.72	1.71
		KF	23.75	27.50	34.81				2.63	1.75	1.69
	$F_t$	UB	<b>7.32</b>	<b>4.24</b>	<b>7.22</b>	7.01	7.61	11.8	3.99	3.12	1.97
		KF	28.71	27.41	33.65				3.95	3.14	1.99
Non-holonomic AUVs	$G_t$	UB	<b>9.26</b>	<b>4.95</b>	<b>8.49</b>	4.96	21.1	6.62	2.6	1.98	1.49
		KF	35.20	21.16	34.98				2.59	1.95	1.47
	$F_t$	UB	<b>10.42</b>	<b>6.40</b>	<b>9.87</b>	7.09	15.2	8.74	5.79	2.97	1.83
		KF	43.27	44.37	43.17				5.8	2.95	1.81
I-AUVs	$G_t$	UB	<b>173.24</b>	<b>171.27</b>	<b>136.2</b>	1.29	2.65	4.57	0.96	1.11	1.59
		KF	245.45	241.58	165.51				0.94	1.06	1.66
	$F_t$	UB	<b>193.76</b>	<b>187.54</b>	<b>128.13</b>	1.35	2.65	4.5	0.78	0.64	7.97
		KF	286.20	280.13	131.16				0.72	0.56	8.03

approach has lower computational complexity than general LQG-based belief space planning approaches.

In our simulations, we considered only spatially varying Gaussian noise. However, in some cases, it is difficult to directly determine the noise distribution. A more complex process or observation model, such as the feature-based observation model in active simultaneous localization and mapping [50], will be analyzed in our future work.

Although the nonlinear programming problem defined in our approach can be solved by general solvers, this approach is time consuming compared with convex optimization. In future work, we will attempt to transform the planning problem into a convex problem, which can produce a globally optimal solution.

Our solution speeds up the belief planning in static environments. However, there are many dynamic obstacles in some actual application environments [51]. Extending our approach to dynamic environments will be a future work.

## APPENDIX

### A. SOLUTION OF SCALAR RICCATI DIFFERENTIAL EQUATIONS

For scalar Riccati equations, we give the following solution [52]:

*Lemma 1:* Let  $a, c > 0, b$  and  $x_0 \geq 0$  be constants and define

$$\alpha(a, b, c) = \alpha = \sqrt{ac + b^2} \quad (22)$$

$$\beta(a, b, c, x_0) = \beta = \frac{ax_0 - \alpha - b}{ax_0 + \alpha - b}.$$

Consider the scalar Riccati differential equation

$$\dot{x}_t = -ax_t^2 + 2bx_t + c \quad (23)$$

with initial condition  $x_0$ .

*Proof 1:* Two solutions for the quadratic equation  $-ax_t^2 + 2bx_t + c = 0$ .

$$x^+ = \frac{b + \alpha}{a} > 0 \text{ and } x^- = \frac{b - \alpha}{a} < 0. \quad (24)$$

Suppose that  $\varepsilon_t = x_t - x^+$  and  $z_t = \frac{1}{\varepsilon_t}$  lead to the (affine) differential equation

$$\partial_t z_t = 2\alpha z_t + a, \quad (25)$$

the solution to which is

$$z_t = -\frac{a}{2\alpha} + \frac{a}{2\alpha\beta} e^{2\alpha t}, \quad (26)$$

and then

$$x_t = \frac{1}{z_t} + x^+ = x^+ + \frac{2\alpha}{a} \frac{\beta e^{-2\alpha t}}{1 - \beta e^{-2\alpha t}}. \quad (27)$$

### B. DISCRETE TIME RICCATI EQUATION

For the discrete system:

$$\begin{aligned} \tilde{x}_{t+1} &= \mathbf{A}_t^p \tilde{x}_t + \mathbf{B}_t^p \tilde{u}_t + \mathbf{G}_t^p \omega_t \\ \tilde{z}_t &= \mathbf{H}_t^p \tilde{x}_t + \mathbf{M}_t^p \nu_t. \end{aligned} \quad (28)$$

The covariance evolution is

$$\begin{aligned} \mathbf{P}_t^- &= \mathbf{A}_t \mathbf{P}_{t-1}^+ \mathbf{A}_t^T + \mathbf{G}_t \mathbf{Q} \mathbf{G}_t^T \\ \mathbf{S}_t &= \mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{M}_t \mathbf{R} \mathbf{M}_t^T \\ \mathbf{K}_t &= \mathbf{P}_t^- (\mathbf{H}_t^T \mathbf{S}_t)^{-1} \\ \mathbf{P}_t^+ &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_t^-. \end{aligned} \quad (29)$$

### C. EQUIVALENCE OF CONTINUOUS AND DISCRETE KALMAN FILTER

1) Theoretical derivations of the equivalence

The continuous model is

$$\begin{aligned} \dot{\mathbf{x}}_t &= \mathbf{A}_t \mathbf{x}_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{F}_t \omega_t \\ \mathbf{z}_t &= \mathbf{H}_t \mathbf{x}_t + \mathbf{M}_t \nu_t. \end{aligned} \quad (30)$$

For white noise processes,  $E[\omega_t] = 0$ ,  $E[\omega_t \omega_t^T(\tau)] = \mathbf{Q}_t \delta(t - \tau)$ , and  $E[\nu_t] = 0$ ,  $E[\nu_t \nu_t^T(\tau)] = \mathbf{R}_t \delta(t - \tau)$ .

The matrices  $\mathbf{Q}_t$  and  $\mathbf{R}_t$  represent the power spectral density (PSD matrices) of the two white noise terms  $\boldsymbol{\omega}_t$  and  $\boldsymbol{\nu}_t$ , respectively. We also assume that the noise vectors are mutually independent  $E[\boldsymbol{\omega}_t \boldsymbol{\nu}_t^T] = 0$ .

The discrete model is

$$\begin{aligned}\mathbf{x}_{t+1} &= \Phi_{t+1,t} \mathbf{x}_t + \Psi_{t+1,t} \mathbf{u}_t + \mathbf{G}_{t+1} \mathbf{W}_t \\ \mathbf{Z}_{t+1} &= \mathbf{H}_{t+1} \mathbf{x}(t+1) + \mathbf{M}_{t+1} \mathbf{V}(t+1)\end{aligned}\quad (31)$$

where  $\mathbf{W}_k$  and  $\mathbf{V}_k$  are white noise sequences.

$$\begin{aligned}E[\mathbf{W}_t] &= E[\mathbf{V}_t] = 0, \quad E[\mathbf{W}_t \mathbf{W}_j^T] = \mathbf{Q}_{t,k} \delta_{kj}, \\ E[\mathbf{V}_t \mathbf{V}_j^T] &= \mathbf{R}_{t,k} \delta_{t,j}, \quad E[\mathbf{W}_t \mathbf{V}_j^T] = 0,\end{aligned}\quad (32)$$

The equivalence between continuous and discrete system models is now shown. Assume the sample interval  $\Delta t = t_{k+1} - t_k$ ,

$$\begin{aligned}\Phi(t_{k+1}, t_k) &= \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{A}(\tau) d\tau\right), \\ \Psi(t_{k+1}, t_k) &= \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) \mathbf{B}(\tau) d\tau, \\ \mathbf{G}(t_{k+1}, t_k) &= \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) \mathbf{F}(\tau) d\tau,\end{aligned}\quad (33)$$

when  $\Delta t \rightarrow 0$ ,

$$\mathbf{Q}_{t_k} = \frac{\mathbf{Q}(t_k)}{\Delta t}, \quad \mathbf{R}_{t_k} = \frac{\mathbf{R}(t_k)}{\Delta t}. \quad (34)$$

## REFERENCES

- [1] G. Antonelli, Underwater Robots. Springer Tracts in Advanced Robotics, 2014.
- [2] F. Bonin-Font, A. Burguera, and J.-L. Lisani, "Visual discrimination and large area mapping of posidonia oceanica using a lightweight AUV," IEEE Access, vol. 5, pp. 24 479–24 494, 2017.
- [3] P. Norgren and R. Skjetne, "A multibeam-based SLAM algorithm for iceberg mapping using AUVs," IEEE Access, vol. 6, pp. 26 318–26 337, 2018.
- [4] M. Martin-Abadal, E. Guerrero-Font, F. Bonin-Font, and Y. Gonzalez-Cid, "Deep semantic segmentation in an AUV for online posidonia oceanica meadows identification," IEEE Access, vol. 6, pp. 60 956–60 967, 2018.
- [5] T. Jinwu, X. Xiaosu, Z. Tao, Z. Liang, and L. Yao, "Study on installation error analysis and calibration of acoustic transceiver array based on SINS/USBL integrated system," IEEE Access, vol. 6, pp. 66 923–66 939, 2018.
- [6] T. Zhang, L. Chen, and Y. Yan, "Underwater positioning algorithm based on SINS/LBL integrated system," IEEE Access, vol. 6, pp. 7157–7163, 2018.
- [7] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions AUVs," Ocean Engineering, vol. 36, no. 1, pp. 15–23, 2009.
- [8] P. Sanz, P. Ridao, G. Oliver, C. Melchiorri, G. Casalino, C. Silvestre, Y. Petillot, and A. Turetta, "TRIDENT: A framework for autonomous underwater intervention missions with dexterous manipulation capabilities," IFAC Proc., vol. 43, no. 16, pp. 187 – 192, 2010.
- [9] D. Li, P. Wang, and L. Du, "Path planning technologies for autonomous underwater vehicles-a review," IEEE Access, vol. 7, pp. 9745–9768, 2018.
- [10] L. Zhang, L. Zhang, S. Liu, J. Zhou, and C. Papavassiliou, "Three-dimensional underwater path planning based on modified wolf pack algorithm," IEEE Access, vol. 5, pp. 22 783–22 795, 2017.
- [11] S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics. MIT press, 2005.
- [12] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," Artif. Intell., vol. 101, no. 1-2, pp. 99–134, 1998.
- [13] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," Int J Rob Res., vol. 33, no. 2, pp. 268–304, 2014.
- [14] J. Van Den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," Int J Rob Res., vol. 30, no. 7, pp. 895–913, 2011.
- [15] R. Platt Jr, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in Proceedings of Robotics: Science and Systems (RSS), 2010.
- [16] J. J. Kuffner Jr and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in Proc. of the IEEE Intl. Conf. on Robot. and Autom., San Francisco, CA, USA, 2000, pp. 995–1001.
- [17] H. Durrant-Whyte, N. Roy, and P. Abbeel, "Identification and representation of homotopy classes of trajectories for search-based path planning in 3D," in Proceedings of Robotics: Science and Systems (RSS), 2011.
- [18] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," Int J Rob Res., vol. 31, no. 11, pp. 1263–1278, 2012.
- [19] ———, "Motion planning under uncertainty using differential dynamic programming in belief space," in Robotics Research. Springer, 2017, pp. 473–490.
- [20] J. Woolfrey, D. Liu, and M. Carmichael, "Kinematic control of an autonomous underwater vehicle-manipulator system (auvms) using autoregressive prediction of vehicle motion and model predictive control," in Proc. of the IEEE Intl. Conf. on Robot. and Autom., Stockholm, SE, 2016, pp. 4591–4596.
- [21] J. D. Hernández, E. Vidal, M. Moll, N. Palomeras, M. Carreras, and L. E. Kavraki, "Online motion planning for unexplored underwater environments using autonomous underwater vehicles," J Field Robot, vol. 36, no. 2, pp. 370–396, 2019.
- [22] H. Yu, W. Lu, and D. Liu, "A unified closed-loop motion planning approach for an I-AUV in cluttered environment with localization uncertainty," in the IEEE Intl. Conf. on Robot. and Autom., Montreal, CA, 2019.
- [23] F. Rezaazadeh, K. Shojai, F. Sheikholeslam, and A. Chatraei, "A novel approach to 6-dof adaptive trajectory tracking control of an auv in the presence of parameter uncertainties," Ocean Engineering, vol. 107, pp. 246–258, 2015.
- [24] X. Xiang, C. Yu, and Q. Zhang, "Robust fuzzy 3D path following for autonomous underwater vehicle subject to uncertainties," COMPUT OPER RES., vol. 84, pp. 165–177, 2017.
- [25] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of auvs with control input nonlinearities using reinforcement learning," IEEE Trans. Syst., Man, Cybern., vol. 47, no. 6, pp. 1019–1029, 2017.
- [26] É. Pairet, J. D. Hernández, M. Lahijanian, and M. Carreras, "Uncertainty-based online mapping and motion planning for marine robotics guidance," in Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, 2018, pp. 2367–2374.
- [27] V. T. Huynh, M. Dubabin, and R. N. Smith, "Predictive motion planning for AUVs subject to strong time-varying currents and forecasting uncertainties," in Proc. of the IEEE Intl. Conf. on Robot. and Autom., 2015, pp. 1144–1151.
- [28] S. MahmoudZadeh, D. Powers, K. Sammut, and A. Yazdani, "Differential evolution for efficient AUV path planning in time variant uncertain underwater environment," Robotics (cs. RO), 2016.
- [29] J. M. Porta, N. Vlassis, M. T. Spaan, and P. Poupart, "Point-based value iteration for continuous pomdps," J Mach Learn Res., vol. 7, no. Nov, pp. 2329–2367, 2006.
- [30] H. Kurniawati, D. Hsu, and W. S. Lee, "Sarsop: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in Robotics: Science and systems, vol. 2008. Zurich, Switzerland., 2008.
- [31] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Planning under uncertainty for robotic tasks with mixed observability," Int J Rob Res., vol. 29, no. 8, pp. 1053–1068, 2010.
- [32] J. Berg, S. Patil, and R. Alterovitz, "Efficient approximate value iteration for continuous Gaussian POMDPs," in Proc. AAAI Conf. on Artificial Intelligence, 2012.
- [33] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in Proc. of the IEEE Intl. Conf. on Robot. and Autom., 2011, pp. 723–730.
- [34] W. Sun, J. van den Berg, and R. Alterovitz, "Stochastic extended lqr for optimization-based motion planning under uncertainty," IEEE Trans. Autom. Sci. Eng., vol. 13, no. 2, pp. 437–447, 2016.

- [35] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, "T-LQG: Closed-loop belief space planning via trajectory-optimized lqg," in Proc. of the IEEE Intl. Conf. on Robot. and Autom., Brisbane, AUS, 2017, pp. 649–656.
- [36] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in Robotics Research. Springer, 2016, pp. 649–666.
- [37] F. Gao, W. Wu, Y. Lin, and S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," in Proc. of the IEEE Intl. Conf. on Robot. and Autom., Brisbane, AU, 2018.
- [38] W. J. Terrell, Stability and stabilization: An introduction. Princeton University Press, 2009.
- [39] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "CHOMP: Covariant hamiltonian optimization for motion planning," Int J Rob Res., vol. 32, no. 9-10, pp. 1164–1193, 2013.
- [40] J. Dong, M. Mukadam, F. Dellaert, and B. Boots, "Motion planning as probabilistic inference using gaussian processes and factor graphs," in Robotics: Science and Systems, vol. 12, 2016.
- [41] B. Wang, W. Shi, and Z. Miao, "Confidence analysis of standard deviational ellipse and its extension into higher dimensional euclidean space," PloS one, vol. 10, no. 3, p. e0118537, 2015.
- [42] J. Nocedal and S. Wright, Numerical optimization. Springer Science & Business Media, 2006.
- [43] S. Joshi and A. Kelkar, "Design of norm-bounded and sector-bounded LQG controllers for uncertain systems," J OPTIMIZ THEORY APP, vol. 113, no. 2, pp. 269–282, 2002.
- [44] H. Qin, Z. Wu, Y. Sun, and H. Chen, "Disturbance-observer-based prescribed performance fault-tolerant trajectory tracking control for ocean bottom flying node," IEEE Access, vol. 7, pp. 49 004–49 013, 2019.
- [45] H. Qin, Z. Wu, Y. Sun, C. Zhang, and C. Lin, "Fault-tolerant prescribed performance control algorithm for underwater acoustic sensor network nodes with thruster saturation," IEEE Access, 2019.
- [46] Y. Wang, S. Jiang, B. Chen, and H. Wu, "Trajectory tracking control of underwater vehicle-manipulator system using discrete time delay estimation," IEEE Access, vol. 5, pp. 7435–7443, 2017.
- [47] S. A. Janse, "Inference using bhattacharyya distance to model interaction effects when the number of predictors far exceeds the sample size," 2017.
- [48] J. Luo and A. Johnson, "Stability robustness of the discrete-time lqg system under plant perturbation and noise uncertainty," INT J CONTROL, vol. 55, no. 6, pp. 1491–1502, 1992.
- [49] M. Mukadam, J. Dong, X. Yan, F. Dellaert, and B. Boots, "Continuous-time gaussian process motion planning via probabilistic inference," Int J Rob Res., vol. 37, no. 11, pp. 1319–1340, 2018.
- [50] 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 Rob Res., vol. 34, no. 7, pp. 849–882, 2015.
- [51] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, "Non-gaussian slap: Simultaneous localization and planning under non-gaussian uncertainty in static and dynamic environments," arXiv preprint arXiv:1605.01776, 2016.
- [52] E. M. T. Karvonen, S. Bonnabel, "Bounds on the covariance matrix of a class of kalman–bucy filters for systems with non-linear dynamics," in the 57th IEEE Conference on Decision and Control, Miami Beach, FL, USA, 2018.



HUAN YU received his B.S. in Electronic and Information Engineering from Hunan University, Changsha, China in 2014. He then received his M.S. in Control Engineering from Beijing Institute of Technology, Beijing, China in 2016, where he is currently pursuing the Ph.D. degree. He is a two-years visiting research Ph.D. student in Centre for Autonomous Systems, University of Technology Sydney, Australia. His research interests include motion planning, localization and SLAM for autonomous robots.



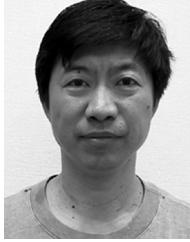
WENJIE LU received his B.S. in Mechatronic Engineering from Zhejiang University, Hangzhou, China in 2009. He then received his M.S. and Ph.D. in Mechanical Engineering from Duke University, Durham, NC, USA in 2011 and 2014 respectively. His research interests include the learning of efficient motion planners and controllers for mobile robots that are subject to uncertainties in 3D complex environment. He has been developing such algorithms via integrating machine learning, approximate dynamic programming, hybrid systems, information theories, and nonparametric Bayesian models.



DIKAI LIU received his Ph.D. degree in 1997. His main research interest is robotics including exploration, motion planning, robot teams, and physical human-robot interaction. He has been developing novel methods and algorithms that enable robots to operate in unstructured and complex 3D environments autonomously or collaboratively with human users. Example robotic systems he developed and practically deployed include autonomous grit-blasting robots for steel bridge maintenance, assistive robots for human strength augmentation in industrial applications.



YONGQIANG HAN received his M.S. and Ph.D. degree in Control Science and Engineering from Beijing Institute of Technology, Beijing, China in 2008 and 2012 respectively. Now he is a lecturer in Automation School of Beijing Institute of Technology. His research interests include robot navigation, localization and control.



QINGHE WU (M'90) received the B.Eng. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1982, the post-graduate diploma and the Dr.Tech.Sci. degrees from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, in 1984 and 1990, respectively. From 1986 to 1994 he was Assistant and Oberassistant with the Institute of Automatic Control, ETH. Since 1995, he has been with the Beijing Institute of Technology, Beijing, China, where he has been Professor since 1997. He was a Visiting Research Fellow from July 2002 to June 2003 in Akita Prefectural University, Akita, Japan. His research interests include H-infinity control and robust control theory. (Based on document published on 17 September 2007).