

# Belief Space Planning Simplified: Trajectory-Optimized LQG (T-LQG) (Extended Report)

Mohammadhussein Rafieisakhaei, Suman Chakravorty, and P. R. Kumar  
Texas A&M University, College Station, Texas 77843, USA  
{mrafieis, schakrav, prk}@tamu.edu

## Abstract

Planning under motion and observation uncertainties requires solution of a stochastic control problem in the space of feedback policies. In this paper, we reduce the general  $(n^2 + n)$ -dimensional belief space planning problem to an  $(n)$ -dimensional problem by obtaining a Linear Quadratic Gaussian (LQG) design with the best nominal performance. Then, by taking the underlying trajectory of the LQG controller as the decision variable, we pose a coupled design of trajectory, estimator, and controller design through a Non-Linear Program (NLP) that can be solved by a general NLP solver. We prove that under a first-order approximation and a careful usage of the separation principle, our approximations are valid. We give an analysis on the existing major belief space planning methods and show that our algorithm has the lowest computational burden. Finally, we extend our solution to contain general state and control constraints. Our simulation results support our design.

## I. INTRODUCTION

Planning under process and measurement uncertainties is referred to as the belief space planning problem. In general, it requires the solution of Hamilton-Jacobi-Bellman (HJB) equations to obtain the optimal feedback policy [1], [2]. The Linear Quadratic Gaussian (LQG) methodology provides the optimal estimator and controller for linear systems with Gaussian noises [3]. However, an LQG planner requires a nominal trajectory to begin with. An approach utilizing this methodology by decoupling the two procedures provides a sequential design of the trajectory and the LQG policy, either by providing a-priori trajectories and comparing the LQG performance over each one [4], or by performing the decoupled procedure iteratively [5], [6]. An approach to a coupled design of trajectory and the policy in non-linear systems utilizing the Extended Kalman Filter (EKF) is based on the heuristic assumption of Most-Likely Observations (MLO) during planning [7]. Another approach considers general belief distributions, either by reformulating the problem in belief space as a Partially Observed Markov Decision Process (POMDP) [8] or by utilizing a Monte-Carlo representation of beliefs [9], [10], [11]. An exact approach to solve POMDPs for continuous action and observation spaces requires continuous (uncountable) branching in the decision tree of beliefs, which leads to intractable computations. The state-of-the-art POMDP solvers are posed in countable decision trees, reducing the search space to discretized belief nodes. Similar to POMDP solvers, particle filter-based methods suffer from the curse of dimensionality due to the depletion of particles in the update step of the estimation.

In this paper, we overcome this hurdle by providing a coupled design of trajectory and the policy using the underlying trajectory as the optimization variable, and simplifying the belief space planning as an optimization problem aiming for estimation performance that can be solved by a general Non-Linear Programming (NLP) solver, which we refer to as the Trajectory-optimized LQG (T-LQG). Essentially, we use the separation principle and the structure of the LQG method to pose an optimization problem on the sequence of control actions over LQG policies, rather than optimizing over the general policy space. This method reduces the dimension of the underlying state in the optimization problem from  $n + n^2$  (Gaussian belief dimension) to  $n$  (state dimension), breaking the computational burden of belief space planning problems. The computational complexity of our method is  $O(Kn^3)$ , where  $K$  is the planning horizon and  $n$  is the state dimension, which is lower than any other belief space planning method.

We utilize the separation principle to separate the design of estimator and the controller. Then—assuming the existence of a linear controller to stabilize the system around a given nominal trajectory—we prove that under a first-order approximation, the stochastic control objective is dominated by the nominal part of the cost function. Moreover, over the given nominal trajectory, the nominal performance of the estimator is given by the dynamic Riccati equations independent from the observations and controller form. The original problem is reformulated and reduced to a deterministic problem over the state space by choosing the underlying nominal trajectory as the optimization variable, aiming for the best estimation performance over that trajectory. The key observation is: fixing the feedback policy to an LQG policy for a linearized system around a nominal trajectory offers the solution of an optimal estimation and control performance along that specific trajectory.

For a fixed linearization trajectory, LQG gives the best estimator and controller to track that nominal trajectory. Our method uses the nominal trajectory itself as an optimization variable in order to obtain the best trajectory, and, subsequently, the best estimator and controller to follow that trajectory. The central difference between this method and the MLO method of [7] is the utilization of the separation principle, which maintains a separate design of the controller to keep the state around the

nominal trajectory. Otherwise, the state deviation from the nominal trajectory keeps growing intractably and the assumptions of the nominal trajectory (of control and subsequent state and observations) collapse, reducing the approach to a heuristic design as in [7]. On the contrary, in our approach, the controller keeps the state around the nominal trajectory, and therefore, the nominal estimation performance becomes valid. Also, using a high-dimensional controller such as a belief-LQR over an  $(n + n^2)$ -dimensional space as in [7]—or [5] and [6]—means avoiding a proper usage of the separation principle by coupling the controller design with the design of the estimator. Using the separation principle also enables us to pose the problem as a standard NLP, rather than using a dynamic programming mechanism (which involves tedious calculations in an  $(n + n^2)$ -dimensional space to solve the coupled equations of the belief estimation and the controller design, as in [5], [6]).

Finally, whenever the accumulated error of linearization (or other errors) increases from a tolerable threshold during the execution, replanning occurs. This is also another merit of posing the planning problem as a standard NLP with low dimension: replanning for a long horizon becomes possible in online applications. Moreover, it enables the utilization of various optimization literature's highly optimized softwares and tools.

Unlike point-based POMDP solvers [12], [13], [14], in T-LQG the time-horizon is a linear factor in the computational complexity, rather than a factor in the exponent—the curse of dimensionality (as in POMDPs and particle filter-based solvers) and the curse of history (as in POMDPs) disappears. This means T-LQG is capable of solving belief space problems on a considerably grander scale. Indeed, current point-based solvers are in short of solving problems in continuous action and observation spaces that are comparable to the problems that are dealt with in this paper.

In the next section, we define the general stochastic control problem to be tackled. We provide our method in section III, which includes the optimization problem, the execution and the replanning strategy. We provide a general method of dealing with non-convex path constraints in section IV, and adapt our optimization to address that. In section V, we provide a theoretical analysis on the validity of our method and provide the sufficient conditions under which our solution is a valid approximation of the original problem. After providing an extensive theoretical comparison between the state-of-the-art belief space planning methods in section VI, we provide our simulation results for several situations in section VII and conclude afterwards. Note, a shorter version of this report has been submitted to WAFR, 2016 [15].

## II. GENERAL PROBLEM

The general belief space planning problem is formulated as a stochastic control problem in the space of feedback policies. In this section, we define the basic elements of the problem, including system equations and belief dynamics.

*System equations:* We denote the state, control and observation vectors by  $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^{n_x}$ ,  $\mathbf{u} \in \mathbb{U} \subset \mathbb{R}^{n_u}$ , and  $\mathbf{z} \in \mathbb{Z} \subset \mathbb{R}^{n_z}$ , respectively. The motion  $f : \mathbb{X} \times \mathbb{U} \times \mathbb{R}^{n_x} \rightarrow \mathbb{X}$ , and observation  $h : \mathbb{X} \rightarrow \mathbb{Z}$  processes are defined as:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\omega}_t), \quad \boldsymbol{\omega}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}_t}) \quad (1a)$$

$$\mathbf{z}_t = h(\mathbf{x}_t, \boldsymbol{\nu}_t), \quad \boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\nu}_t}) \quad (1b)$$

where  $\{\boldsymbol{\omega}_t\}$  and  $\{\boldsymbol{\nu}_t\}$  are zero mean independent, identically distributed (i.i.d.) mutually independent random sequences.

*Belief (information state):* The conditional distribution of  $\mathbf{x}_t$  given the data history up to time  $t$ , is called the belief  $\mathbf{b}_t : \mathbb{X} \times \mathbb{Z}^t \times \mathbb{U}^t \rightarrow \mathbb{R}$ . It is defined as  $\mathbf{b}_t(\mathbf{x}, \mathbf{z}_{0:t}, \mathbf{u}_{0:t-1}, \mathbf{b}_0) := p_{\mathbf{x}_t | \mathbf{z}_{0:t}; \mathbf{u}_{0:t-1}}(\mathbf{x} | \mathbf{z}_{0:t}; \mathbf{u}_{0:t-1}; \mathbf{b}_0)$ , and denoted by  $\mathbf{b}_t$  in this paper. We use a Gaussian representation of belief in belief space,  $\mathbb{B}$ , [16]. The belief dynamics follow Bayesian update equations, summarized as a function  $\tau : \mathbb{B} \times \mathbb{U} \times \mathbb{Z} \rightarrow \mathbb{B}$ , where  $\mathbf{b}_{t+1} = \tau(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_{t+1})$  [1], [17].

**Problem 1. Stochastic Control Problem** Given an initial belief state  $\mathbf{b}_0$ , solve for the optimal policy as follows:

$$\begin{aligned} \min_{\pi} J^{\pi} &:= \mathbb{E} \left[ \sum_{t=0}^{K-1} c_t^{\pi}(\mathbf{b}_t, \mathbf{u}_t) + c_K^{\pi}(\mathbf{b}_K) \right] \\ \text{s.t. } \mathbf{b}_{t+1} &= \tau(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_{t+1}), \end{aligned} \quad (2)$$

where the optimization is over feasible policies, and  $\pi := \{\pi_0, \dots, \pi_t\}$  where  $\pi_t : \mathbb{B} \rightarrow \mathbb{U}$  specifies an action given the belief,  $\mathbf{u}_t = \pi_t(\mathbf{b}_t)$ . Moreover,  $c_t^{\pi}(\cdot, \cdot) : \mathbb{B} \times \mathbb{U} \rightarrow \mathbb{R}$  is the one-step cost function,  $c_K^{\pi}(\cdot) : \mathbb{B} \rightarrow \mathbb{R}$  denotes the terminal cost, and the expectation is taken over all randomness.

## III. BELIEF SPACE PLANNING METHOD: T-LQG

In this section, we provide the details of our design for the planning problem.

### A. Planning Problem

Here, we provide the details of our design for the planning problem.

*Parameterized Trajectories (p-traj):* Using the noiseless equation of (1a) we parametrize the possible feasible propagated trajectories of the initial estimate,  $\hat{\mathbf{x}}_0$ , given a set of unknown control inputs  $\{\mathbf{u}_t^p\}_{t=0}^{K-1}$ , as:

$$\mathbf{x}_{t+1}^p = f(\mathbf{x}_t^p, \mathbf{u}_t^p, 0), \quad 0 \leq t \leq K-1$$

where  $\mathbf{x}_0^p = \hat{\mathbf{x}}_0$ . It is important to note that the trajectory  $\{\mathbf{x}_t^p\}_{t=0}^K$  changes by changing the underlying control inputs  $\{\mathbf{u}_t^p\}_{t=0}^{K-1}$ . We will refer to the parametrized trajectory of  $\{\mathbf{x}_t^p\}_{t=0}^K$ ,  $\{\mathbf{u}_t^p\}_{t=0}^{K-1}$  as the  $p$ -*traj*.

*Linearization of the system equations:* We linearize the non-linear motion and observation models of equation (1) about the parametrized trajectory  $p$ -*traj*:

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{A}_t^p(\mathbf{x}_t^p, \mathbf{u}_t^p)\tilde{\mathbf{x}}_t + \mathbf{B}_t^p(\mathbf{x}_t^p, \mathbf{u}_t^p)\tilde{\mathbf{u}}_t + \mathbf{G}_t^p(\mathbf{x}_t^p, \mathbf{u}_t^p)\boldsymbol{\omega}_t \quad (3a)$$

$$\tilde{\mathbf{z}}_t = \mathbf{H}_t^p(\mathbf{x}_t^p)\tilde{\mathbf{x}}_t + \mathbf{M}_t^p(\mathbf{x}_t^p)\boldsymbol{\nu}_t, \quad (3b)$$

where  $\mathbf{A}_t^p(\mathbf{x}_t^p, \mathbf{u}_t^p) = \partial f(\mathbf{x}, \mathbf{u}, \boldsymbol{\omega}) / \partial \mathbf{x}|_{\mathbf{x}_t^p, \mathbf{u}_t^p, 0}$ ,  $\mathbf{B}_t^p(\mathbf{x}_t^p, \mathbf{u}_t^p) = \partial f(\mathbf{x}, \mathbf{u}, \boldsymbol{\omega}) / \partial \mathbf{u}|_{\mathbf{x}_t^p, \mathbf{u}_t^p, 0}$ ,  $\mathbf{G}_t^p(\mathbf{x}_t^p, \mathbf{u}_t^p) = \partial f(\mathbf{x}, \mathbf{u}, \boldsymbol{\omega}) / \partial \boldsymbol{\omega}|_{\mathbf{x}_t^p, \mathbf{u}_t^p, 0}$ ,  $\mathbf{H}_t^p(\mathbf{x}_t^p) = \partial h(\mathbf{x}, \boldsymbol{\nu}) / \partial \mathbf{x}|_{\mathbf{x}_t^p, 0}$  and  $\mathbf{M}_t^p(\mathbf{x}_t^p) = \partial h(\mathbf{x}, \boldsymbol{\nu}) / \partial \boldsymbol{\nu}|_{\mathbf{x}_t^p, 0}$ . Moreover  $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}_t^p$ ,  $\tilde{\mathbf{u}}_t := \mathbf{u}_t - \mathbf{u}_t^p$  and  $\tilde{\mathbf{z}}_t := \mathbf{z}_t - \mathbf{z}_t^p$  define the state, control and observation errors, respectively, and  $\mathbf{z}_t^p = h(\mathbf{x}_t^p)$ . Note that, as the control inputs change, the underlying trajectory they represent,  $\{\mathbf{u}_t^p\}_{t=0}^{K-1}$  and  $\{\mathbf{x}_t^p\}_{t=0}^K$ , also change, and therefore Jacobian matrices change.

*Expectation over policies:* In the cost function of problem 1, the expectation is over all types of policies in the feasible policy space. We assume the search is over linear feedback policies (LQG class), which is a valid assumption for locally controlling a linearized model around a nominal trajectory.

*Separation principle:* It can be shown that the minimization of the stochastic cost function for a linear Gaussian system in terms of state error  $\mathbf{x}_t - \hat{\mathbf{x}}_t^p$  is equivalent to two separate minimizations in terms of the estimation error  $\mathbf{x}_t - \hat{\mathbf{x}}_t$  and the controller error  $\hat{\mathbf{x}}_t - \mathbf{x}_t^p$ , where  $\mathbf{x}_t - \mathbf{x}_t^p = \mathbf{x}_t - \hat{\mathbf{x}}_t + \hat{\mathbf{x}}_t - \mathbf{x}_t^p$  [17]. As a result, the design of a stochastic controller with partially observed states reformulates to two separate designs of an estimator and a fully observed controller. In the LQG case, the estimator is a Kalman filter and the controller is an LQR controller.

*Estimation cost:* The optimization cost function,  $J$ , is a quadratic cost:

$$J := \mathbb{E} \left[ \sum_{t=1}^K \tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t + \tilde{\mathbf{u}}_{t-1}^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1} \right], \quad (4)$$

where  $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \hat{\mathbf{x}}_t$  is the estimator error (KF error), and  $\mathbf{W}_t^x, \mathbf{W}_t^u \succeq 0$  are two positive-definite weight matrices. Moreover, we assume  $\mathbf{W}_t^x$  is symmetric. Therefore, these matrices have square root. Note, this cost function can be rewritten as a function of the belief trajectory, which is shown in Appendix A. However, as we show in section V, under a first-order approximation,  $J$  is dominated by the part of the cost function that depends only on the underlying nominal trajectory, which is the  $p$ -*traj* in this case. Thus, we approximate  $J$  with  $J^p$  as follows:

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

where the first term is a function of the nominal belief which is discussed next.

*Nominal evolution of covariance:* Define  $\mathbf{P}_{\mathbf{b}_t^p}^+(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p) := \mathbb{E}[(\mathbf{x}_t - \mathbf{x}_t^p)(\mathbf{x}_t - \mathbf{x}_t^p)^T]$  to be the covariance of the nominal belief along the  $p$ -*traj*. Therefore, the first term of the cost function (5), can be rewritten as  $\sum_{t=1}^K \text{tr}(\mathbf{W}_t \mathbf{P}_{\mathbf{b}_t^p}^+ \mathbf{W}_t^T)$ , where  $\mathbf{W}_t^x = \mathbf{W}_t^T \mathbf{W}_t$  is the Cholesky decomposition of  $\mathbf{W}_t^x$  (see Appendix A). For a presumptive parametrized trajectory  $p$ -*traj*, the evolution of  $\mathbf{P}_{\mathbf{b}_t^p}^+$  is given by the recursive Riccati equations independent from the observations but as a function of the trajectory itself. Note, this also provides the nominal performance of the estimation along that trajectory. The dependencies on the  $p$ -*traj* inside the parenthesis is dropped for simplicity. The evolution of  $\mathbf{P}_{\mathbf{b}_t^p}^+$  is given as follows:

$$\begin{aligned} \mathbf{P}_{\mathbf{b}_t^p}^-(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p) &:= \mathbf{A}_t^p(\mathbf{x}_t^p, \mathbf{u}_t^p) \mathbf{P}_{\mathbf{b}_{t-1}^p}^+(\mathbf{x}_{0:t-1}^p, \mathbf{u}_{0:t-2}^p) (\mathbf{A}_t^p(\mathbf{x}_t^p, \mathbf{u}_t^p))^T + \mathbf{G}_t^p(\mathbf{x}_t^p, \mathbf{u}_t^p) \boldsymbol{\Sigma}_{\boldsymbol{\omega}_t} (\mathbf{G}_t^p(\mathbf{x}_t^p, \mathbf{u}_t^p))^T \\ \mathbf{S}_{\mathbf{b}_t^p}(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p) &:= \mathbf{H}_t^p(\mathbf{x}_t^p) \mathbf{P}_{\mathbf{b}_t^p}^-(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p) (\mathbf{H}_t^p(\mathbf{x}_t^p))^T + \mathbf{M}_t^p(\mathbf{x}_t^p) \boldsymbol{\Sigma}_{\boldsymbol{\nu}_t} (\mathbf{M}_t^p(\mathbf{x}_t^p))^T \\ \mathbf{K}_{\mathbf{b}_t^p}(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p) &:= \mathbf{P}_{\mathbf{b}_t^p}^-(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p) (\mathbf{H}_t^p(\mathbf{x}_t^p))^T (\mathbf{S}_{\mathbf{b}_t^p}(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p))^{-1} \\ \mathbf{P}_{\mathbf{b}_t^p}^+(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p) &= (\mathbf{I} - \mathbf{K}_{\mathbf{b}_t^p}(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p) \mathbf{H}_t^p(\mathbf{x}_t^p)) \mathbf{P}_{\mathbf{b}_t^p}^-(\mathbf{x}_{0:t}^p, \mathbf{u}_{0:t-1}^p) \end{aligned}$$

which is solvable with the initial condition  $\mathbf{P}_{\mathbf{b}_0^p}^+(\mathbf{x}_0^p) = \boldsymbol{\Sigma}_{\mathbf{x}_0}$ .

**Problem 2. Planning Problem** Given an initial belief  $\mathbf{b}_0 \in \mathbb{B}$ , a goal region represented as an  $\ell_2$ -norm ball,  $\mathbf{B}_{r_g}(\mathbf{x}_g)$ , of radius  $r_g$  around a goal state  $\mathbf{x}_g \in \mathbb{X}$ , and a planning horizon of  $K > 0$ , we define the following problem:

$$\min_{\mathbf{u}_{0:K-1}^p} \sum_{t=1}^K [\text{tr}(\mathbf{W}_t \mathbf{P}_{\mathbf{b}_t^p}^+ \mathbf{W}_t^T) + (\mathbf{u}_{t-1}^p)^T \mathbf{W}_t^u \mathbf{u}_{t-1}^p] \quad (6a)$$

$$s.t. \quad \mathbf{P}_{\mathbf{b}_t^p}^- = \mathbf{A}_t^p \mathbf{P}_{\mathbf{b}_{t-1}^p}^+ (\mathbf{A}_t^p)^T + \mathbf{G}_t^p \Sigma_{\omega_t} (\mathbf{G}_t^p)^T \quad (6a)$$

$$\mathbf{S}_{\mathbf{b}_t^p} = \mathbf{H}_t^p \mathbf{P}_{\mathbf{b}_t^p}^- (\mathbf{H}_t^p)^T + \mathbf{M}_t^p \Sigma_{\nu_t} (\mathbf{M}_t^p)^T \quad (6b)$$

$$\mathbf{K}_{\mathbf{b}_t^p} = \mathbf{P}_{\mathbf{b}_t^p}^- (\mathbf{H}_t^p)^T \mathbf{S}_{\mathbf{b}_t^p}^{-1} \quad (6c)$$

$$\mathbf{P}_{\mathbf{b}_t^p}^+ = (\mathbf{I} - \mathbf{K}_{\mathbf{b}_t^p} \mathbf{H}_t^p) \mathbf{P}_{\mathbf{b}_t^p}^- \quad (6d)$$

$$\mathbf{P}_{\mathbf{b}_0^p}^+ = \Sigma_{\mathbf{x}_0} \quad (6e)$$

$$\mathbf{x}_0^p = \mathbb{E}_{\mathbf{x}}[\mathbf{b}_0(\mathbf{x})] \quad (6f)$$

$$\mathbf{x}_{t+1}^p = f(\mathbf{x}_t^p, \mathbf{u}_t^p, 0) \quad 0 \leq t \leq K-1 \quad (6g)$$

$$\|\mathbf{x}_K^p - \mathbf{x}_g\|_2 < r_g \quad (6h)$$

$$\|\mathbf{u}_t^p\|_2 \leq r_u, \quad 1 \leq t \leq K, \quad (6i)$$

where equations (6a)-(6d) are regarded as one constraint at each time step, and are used to calculate first term of the objective at that time step, equations (6e) and (6f) represent the initial conditions, equation (6g) defines the state propagation (and relates the optimization variables to the state trajectory), equation (6h) constrains the terminal state within an  $\ell_2$ -norm ball of radius  $r_g > 0$  around the goal state, equation (6i) accounts for the saturation constraints for  $r_u > 0$ . Moreover, the first term of the objective aims for minimizing the estimation uncertainty, whereas the second term penalizes the control effort. Note, this problem should be viewed as an optimization in the space of control actions with the control actions as the variables and every other variable, such as the covariances, as a function of those controls.

*Optimized Trajectory, o-traj:* We will denote the resulting optimized  $p$ -traj of problem 2 with  $\{\mathbf{x}_t^o\}_{t=0}^K$ ,  $\{\mathbf{u}_t^o\}_{t=0}^{K-1}$  and refer to it as the  $o$ -traj.

*Feedback control:* The resulting trajectory from the optimization problem is optimized in terms of estimation performance. Now, using the separation principle, the LQR controller is designed to follow the  $o$ -traj. Therefore, the LQR cost is designed for the controller error  $\hat{\mathbf{x}}_t - \mathbf{x}_t^o$ . The resulting control policy is a linear feedback policy, and the evolution of  $\hat{\mathbf{x}}_t$  is obtained from the KF equations using the actual observations in the execution. The details of these equations, which are common to any LQG problem, are discussed next.

*Linearization of system equations:* Note that the linearization of the system equations still follow the same procedure as explained before in equations (3). This is because, the  $o$ -traj is a  $p$ -traj, as well. However, for simplicity, we would denote the Jacobian matrices and every other variable associated with the  $o$ -traj with an  $o$  superscript. Thus, for the rest of the paper, the system equations are linearized around the  $o$ -traj, with  $\mathbf{A}_t^o = \partial f(\mathbf{x}, \mathbf{u}, \omega) / \partial \mathbf{x}|_{\mathbf{x}_t^o, \mathbf{u}_t^o, 0}$ ,  $\mathbf{B}_t^o = \partial f(\mathbf{x}, \mathbf{u}, \omega) / \partial \mathbf{u}|_{\mathbf{x}_t^o, \mathbf{u}_t^o, 0}$ ,  $\mathbf{G}_t^o = \partial f(\mathbf{x}, \mathbf{u}, \omega) / \partial \omega|_{\mathbf{x}_t^o, \mathbf{u}_t^o, 0}$ ,  $\mathbf{H}_t^o = \partial h(\mathbf{x}, \nu) / \partial \mathbf{x}|_{\mathbf{x}_t^o, 0}$  and  $\mathbf{M}_t^o = \partial h(\mathbf{x}, \nu) / \partial \nu|_{\mathbf{x}_t^o, 0}$ . Moreover, define  $\mathbf{f}_t^o := f(\mathbf{x}_t^o, \mathbf{u}_t^o, 0) - \mathbf{A}_t^o \mathbf{x}_t^o - \mathbf{B}_t^o \mathbf{u}_t^o$ ,  $\mathbf{h}_t^o := h(\mathbf{x}_t^o, 0) - \mathbf{H}_t^o \mathbf{x}_t^o$ .

*LQR cost function:* The feedback controller cost function,  $J^f$  (with the superscript  $f$  accounting for the feedback controller), is a common quadratic cost, as follows:

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

where  $\tilde{\mathbf{u}}_t^o = \mathbf{u}_t - \mathbf{u}_t^o$  and  $\mathbf{W}_t^x, \mathbf{W}_t^u$  are as defined before.

*Control policy:* The resulting control policy is proved to be a linear feedback policy as follows:

$$\tilde{\mathbf{u}}_t^o = -\mathbf{L}_t^o (\hat{\mathbf{x}}_t - \mathbf{x}_t^o),$$

where the linear feedback gain  $\mathbf{L}_t^o$  is given as follows:

$$\mathbf{L}_t^o = (\mathbf{W}_t^u + (\mathbf{B}_t^o)^T \mathbf{P}_t^f \mathbf{B}_t^o)^{-1} (\mathbf{B}_t^o)^T \mathbf{P}_t^f \mathbf{A}_t^o,$$

and the matrix  $\mathbf{P}_t^f$  is the result of backward iterations of the dynamic Riccati equation as follows:

$$\mathbf{P}_{t-1}^f = (\mathbf{A}_t^o)^T \mathbf{P}_t^f \mathbf{A}_t^o - (\mathbf{A}_t^o)^T \mathbf{P}_t^f \mathbf{B}_t^o (\mathbf{W}_t^u + (\mathbf{B}_t^o)^T \mathbf{P}_t^f \mathbf{B}_t^o)^{-1} (\mathbf{B}_t^o)^T \mathbf{P}_t^f \mathbf{A}_t^o + \mathbf{W}_t^x,$$

which is solvable with a terminal condition  $\mathbf{P}_K^f = \mathbf{W}_K^x$ .

*Estimation Trajectory, e-traj:* The estimate of the state during the execution can be obtained using a Kalman Filter. This filtered trajectory is denoted by  $\{\hat{\mathbf{x}}_t\}_{t=0}^K$ , and called the  $e$ -traj. The equations governing its dynamics and relations to the corresponding control actions are derived next.

*Estimation with KF:* coupled with the LQR controller, in an LQG control strategy lies an Kalman filter to obtain the updated posterior distribution, whose mean update equations are as follows:

$$\hat{\mathbf{x}}_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1}^o \mathbf{H}_{t+1}^o) \mathbf{f}_t^o - \mathbf{K}_{t+1}^o \mathbf{h}_{t+1}^o + \mathbf{A}_t^o \hat{\mathbf{x}}_t + \mathbf{B}_t^o \mathbf{u}_t + \mathbf{K}_{t+1}^o (\mathbf{z}_{t+1} - \mathbf{H}_{t+1}^o (\mathbf{A}_t^o \hat{\mathbf{x}}_t + \mathbf{B}_t^o \mathbf{u}_t)) \quad (7)$$

with  $\hat{\mathbf{x}}_0 = \mathbb{E}(\mathbf{x}_0)$  (see Appendix B for derivation). Moreover,  $\mathbf{K}_t^o$  is the Kalman gain whose equation is as follows:

$$\mathbf{K}_t^o = \mathbf{A}_t^o \mathbf{P}_t^o (\mathbf{H}_t^o)^T (\mathbf{H}_t^o \mathbf{P}_t^o (\mathbf{H}_t^o)^T + \mathbf{M}_t^o \Sigma_{\nu_t} (\mathbf{M}_t^o)^T)^{-1},$$

and the matrix  $\mathbf{P}_t^o := \mathbb{E}[(\mathbf{x}_t - \mathbf{x}_t^o)(\mathbf{x}_t - \mathbf{x}_t^o)^T]$  is the covariance of the state estimation, whose governing equations are the forward Riccati equations as follows:

$$\mathbf{P}_{t+1}^o = \mathbf{A}_t^o (\mathbf{P}_t^o - \mathbf{P}_t^o (\mathbf{H}_t^o)^T (\mathbf{H}_t^o \mathbf{P}_t^o (\mathbf{H}_t^o)^T + \mathbf{M}_t^o \Sigma_{\nu_t} (\mathbf{M}_t^o)^T)^{-1} \mathbf{H}_t^o \mathbf{P}_t^o + \mathbf{G}_t^o \Sigma_{\omega_t} (\mathbf{G}_t^o)^T) (\mathbf{A}_t^o)^T, \quad (8)$$

which is solvable with the initial condition of  $\mathbf{P}_0^o := \mathbb{E}[(\mathbf{x}_0 - \mathbf{x}_0^o)(\mathbf{x}_0 - \mathbf{x}_0^o)^T]$ .

### B. Replanning During Execution

*Replanning during execution:* In a stochastic system, even with a closed loop control strategy, after a finite number of execution steps, the estimate deviates from the planned trajectory. This happens due to the accumulation of errors resulting from the un-modeled dynamics, noise, non-linearities, and unpredicted forces. In such a situation, the planned policy becomes irrelevant and a new policy is needed to drive the agent toward the predefined goals. In order to overcome the problem, we track the nominal belief. Then, based on the Kullback-Leibler divergence between the nominal and true belief, we define a symmetric distance (average of unsymmetrical KL-divergences). Once such a distance is greater than a predefined threshold  $d_{th} > 0$ , a deviation is detected, the planning module is initialized with the current belief, and all planning procedures are performed again. The details are discussed next.

*Kullback-Leibler (KL) divergence:* The KL divergence itself is not a symmetric distance function, however, a symmetric distance can be easily derived from that. If  $D_{KL}(Q_1 \parallel Q_2)$  denotes the KL divergence of  $Q_1$  and  $Q_2$ , where the latter are two probability distributions, then  $d(Q_1, Q_2) := (D_{KL}(Q_1 \parallel Q_2) + D_{KL}(Q_2 \parallel Q_1))/2$  denotes a distance between  $Q_1$  and  $Q_2$ , where

$$D_{KL}(Q_1 \parallel Q_2) = \int_{-\infty}^{\infty} q_1(\mathbf{x}) \log\left(\frac{q_1(\mathbf{x})}{q_2(\mathbf{x})}\right) d\mathbf{x},$$

with  $q_1(\mathbf{x})$  and  $q_2(\mathbf{x})$  denoting the densities of  $Q_1$  and  $Q_2$ .

*Detection of a deviation:* At any time instance, the estimation model reports the current belief as  $\mathbf{b}_t = \mathcal{N}(\hat{\mathbf{x}}_t, \mathbf{P}_t^o)$ , whereas the nominal belief at  $t$  is  $\mathbf{b}_t^o = \mathcal{N}(\mathbf{x}_t^o, \mathbf{P}_t^o)$ . Therefore, using the KL divergence formula for multivariate Gaussian distributions [18], the distance between  $\mathbf{b}_t$  and  $\mathbf{b}_t^o$  is as follows:

$$\begin{aligned} d(\mathbf{b}_t, \mathbf{b}_t^o) &= \frac{1}{4} [\log \frac{|\mathbf{P}_t^o|}{|\mathbf{P}_t|} - n_x + \text{tr}((\mathbf{P}_t^o)^{-1} \mathbf{P}_t) + (\mathbf{x}_t^o - \hat{\mathbf{x}}_t)^T (\mathbf{P}_t^o)^{-1} (\mathbf{x}_t^o - \hat{\mathbf{x}}_t)] \\ &\quad + \frac{1}{4} [\log \frac{|\mathbf{P}_t|}{|\mathbf{P}_t^o|} - n_x + \text{tr}((\mathbf{P}_t)^{-1} \mathbf{P}_t^o) + (\hat{\mathbf{x}}_t - \mathbf{x}_t^o)^T (\mathbf{P}_t)^{-1} (\hat{\mathbf{x}}_t - \mathbf{x}_t^o)] \\ &= \frac{1}{2} [n_x(n_x - 1) + (\hat{\mathbf{x}}_t - \mathbf{x}_t^o)^T (\mathbf{P}_t^o)^{-1} (\hat{\mathbf{x}}_t - \mathbf{x}_t^o)], \end{aligned} \quad (9)$$

where  $|\mathbf{P}|$  denotes the determinant of the matrix  $\mathbf{P}$ . Thus a deviation is detected if  $d(\mathbf{b}_t, \mathbf{b}_t^o) > d_{th}$ .

## IV. NON-CONVEX STATE CONSTRAINTS

Non-convex state constraints are handled with barrier functions.

*Polygonal obstacles approximated by ellipsoids:* Given a set of vertices that constitute a polygonal obstacle, we find the Minimum Volume Enclosing Ellipsoid (MVEE) and obtain its parameters [19]. Particularly, for the  $i$ th obstacle, the barrier function includes a Gaussian-like function, where the argument of the exponential is the MVEE, which can be disambiguated with its center  $\mathbf{c}^i$  and a positive definite matrix  $\mathbf{E}^i$  that determines the rotation and axes of the ellipsoid. Moreover, we add several number of inverse functions that tend to infinity along the major and minor axes of the ellipsoid. So, the overall function acts as a barrier to prevent the trajectory from entering the region enclosed by the ellipsoid. Note that for non-polygonal obstacles, one can find the MVEE, and the algorithm works independently from this fact. Therefore, given the ellipsoid parameters  $\mathcal{C} := (\mathbf{c}^1, \mathbf{c}^2, \dots, \mathbf{c}^{n_b}) \in \mathbb{R}^{n_x \times n_b}$  and  $\mathcal{E} := (\mathbf{E}^1, \dots, \mathbf{E}^{n_b}) \in \mathbb{R}^{n_x^2 \times n_b}$ , the Obstacle Barrier Function (OBF) is constructed as follows:

$$\Phi^{(\mathcal{E}, \mathcal{C})}(\mathbf{x}) := \sum_{i=1}^{n_b} \left[ M_1 \exp(-[(\mathbf{x} - \mathbf{c}^i)^T \mathbf{E}^i (\mathbf{x} - \mathbf{c}^i)]^q) + M_2 \sum_{\theta=0:\epsilon_m:1} \left[ \|\mathbf{x} - (\theta \zeta^{i,1} + (1-\theta) \zeta^{i,2})\|_2^{-2} + \|\mathbf{x} - (\theta \xi^{i,1} + (1-\theta) \xi^{i,2})\|_2^{-2} \right] \right],$$

where  $\epsilon_m = 1/m$ ,  $m \in \mathbb{Z}^+$ ,  $M_1, M_2 \geq 0$ ,  $q \in \mathbb{Z}^+$ , and  $\zeta^{i,1}$ ,  $\zeta^{i,2}$  and  $\xi^{i,1}$ ,  $\xi^{i,2}$  are the endpoints of the major and minor axes of the ellipsoid, respectively. Therefore, the second term in the sum places inverse function whose values tend to infinity along the axes of the ellipsoid at points formed by convex combination of the two endpoints of each axis. As  $\epsilon_m$  tends to zero, the entire axes of the ellipsoid become infinite, and, therefore, act as a barrier to any continuous trajectory of states. One can think of putting more infinity points inside the ellipsoid by forming the convex combination of the existing infinity points. Moreover, the first summand determines the territory of the ellipsoid and imposes an outwards gradient around the ellipsoid, acting as a penalty function pushing the trajectory out of the banned region. Hence, we define the cost of avoiding obstacles as:

$$\text{cost}_{\text{obst}}(\mathbf{x}_{t1}, \mathbf{x}_{t2}) := \int_{\mathbf{x}_{t1}}^{\mathbf{x}_{t2}} \Phi(\mathcal{E}, \mathcal{C})(\mathbf{x}') d\mathbf{x}', \quad (10)$$

which is the line integral of the OBF between two given points of the trajectory  $\mathbf{x}_{t1}$  and  $\mathbf{x}_{t2}$ . Therefore, the addition of this cost to the optimization objective, ensures the solver minimizes this cost and keeps the trajectory out of banned regions. However, for implementation purposes, the integral in equation (10) is approximated by a finite Riemann sum consisting of fewer points between  $\mathbf{x}_{t1}$  and  $\mathbf{x}_{t2}$ . Using this equation, we add the running obstacle cost of  $\text{cost}_{\text{obst}}(\mathbf{x}_{t-1}, \mathbf{x}_t)$  to the optimization objective and use the modified optimization problem to obtain locally optimal solutions in the inter-obstacle feasible space using gradient descent methods [20].

## V. THEORETICAL CONSIDERATIONS

In this section, we derive the conditions upon which our planning strategy is an acceptable approximation of the original problem. We perform a first-order approximation on the propagated errors of the state, control, observation, belief, and cost function around a nominal trajectory that stems from a nominal trajectory of the control actions. We show that under some sufficiency conditions, our planning strategy provides an acceptable approximation to the original problem. Note all proofs of the lemmas and the theorem are provided in Appendix C.

*Linearization of process and observation models:* Assuming there exists a nominal trajectory of control actions,  $\{\mathbf{u}_t^p\}_{t=0}^{K-1}$ , then there exists a corresponding nominal trajectory of states,  $\{\mathbf{x}_t^p\}_{t=0}^{K-1}$  and observations  $\{\mathbf{z}_t^p\}_{t=0}^{K-1}$ , where  $\mathbf{x}_{t+1}^p = f(\mathbf{x}_t^p, \mathbf{u}_t^p, \mathbf{0})$ ,  $\mathbf{z}_t^p = h(\mathbf{x}_t^p)$ , and the state, control, and observation error vectors are  $\tilde{\mathbf{u}}_t = \mathbf{u}_t - \mathbf{u}_t^p$ ,  $\tilde{\mathbf{x}}_t = \mathbf{x}_t - \mathbf{x}_t^p$  and  $\tilde{\mathbf{z}}_t = \mathbf{z}_t - \mathbf{z}_t^p$ , respectively. We linearize the state and observation dynamics around the nominal trajectories of state and control as  $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\omega}_t) \approx \mathbf{x}_{t+1}^p + \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \mathbf{G}_t \boldsymbol{\omega}_t$ , and  $\mathbf{z}_t = h(\mathbf{x}_t, \boldsymbol{\nu}_t) \approx \mathbf{z}_t^p + \mathbf{H}_t \tilde{\mathbf{x}}_t + \mathbf{M}_t \boldsymbol{\nu}_t$ , where  $\mathbf{A}_t = \partial f(\mathbf{x}, \mathbf{u}, \boldsymbol{\omega}) / \partial \mathbf{x}|_{\mathbf{x}_t^p, \mathbf{u}_t^p, \mathbf{0}}$ ,  $\mathbf{B}_t = \partial f(\mathbf{x}, \mathbf{u}, \boldsymbol{\omega}) / \partial \mathbf{u}|_{\mathbf{x}_t^p, \mathbf{u}_t^p, \mathbf{0}}$ ,  $\mathbf{G}_t = \partial f(\mathbf{x}, \mathbf{u}, \boldsymbol{\omega}) / \partial \boldsymbol{\omega}|_{\mathbf{x}_t^p, \mathbf{u}_t^p, \mathbf{0}}$ ,  $\mathbf{H}_t = \partial h(\mathbf{x}, \boldsymbol{\nu}) / \partial \mathbf{x}|_{\mathbf{x}_t^p, \mathbf{0}}$  and  $\mathbf{M}_t = \partial h(\mathbf{x}, \boldsymbol{\nu}) / \partial \boldsymbol{\nu}|_{\mathbf{x}_t^p, \mathbf{0}}$  are the corresponding Jacobian matrices. Note that,  $\tilde{\mathbf{z}}_t = \mathbf{H}_t \tilde{\mathbf{x}}_t + \mathbf{M}_t \boldsymbol{\nu}_t$  denotes the observation innovation, as well.

*Linearization of the belief dynamics:* Corresponding to the nominal trajectory of states, there exists a nominal (Gaussian) trajectory of beliefs,  $\{\mathbf{b}_t^p\}_{t=0}^K$ , where  $\mathbf{x}_t^p = \mathbb{E}[\mathbf{b}_t^p]$ , and the covariance evolution follows the Riccati equations from a KF. Likewise, we linearize the belief dynamics around the nominal trajectories of belief, control and observation as  $\mathbf{b}_{t+1} = \tau(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_{t+1}) \approx \mathbf{b}_{t+1}^p + \mathbf{T}_t^b \tilde{\mathbf{b}}_t + \mathbf{T}_t^u \tilde{\mathbf{u}}_t + \mathbf{T}_t^z \tilde{\mathbf{z}}_{t+1}$ , where  $\mathbf{b}_{t+1}^p = \tau(\mathbf{b}_t^p, \mathbf{u}_t^p, \mathbf{z}_{t+1}^p)$ , and  $\mathbf{T}_t^b = \partial \tau(\mathbf{b}, \mathbf{u}, \mathbf{z}) / \partial \mathbf{b}|_{\mathbf{b}_t^p, \mathbf{u}_t^p, \mathbf{z}_{t+1}^p}$ ,  $\mathbf{T}_t^u = \partial \tau(\mathbf{b}, \mathbf{u}, \mathbf{z}) / \partial \mathbf{u}|_{\mathbf{b}_t^p, \mathbf{u}_t^p, \mathbf{z}_{t+1}^p}$ , and  $\mathbf{T}_t^z = \partial \tau(\mathbf{b}, \mathbf{u}, \mathbf{z}) / \partial \mathbf{z}|_{\mathbf{b}_t^p, \mathbf{u}_t^p, \mathbf{z}_{t+1}^p}$ . Moreover,  $\tilde{\mathbf{b}}_t := \mathbf{b}_t - \mathbf{b}_t^p$  denotes the belief error.

*Linearization of the cost function:* Finally, we linearize the cost function around the nominal trajectories of belief and control actions as  $J \approx J^p + \mathbb{E}[\sum_{t=0}^{K-1} (\mathbf{C}_t^b \tilde{\mathbf{b}}_t + \mathbf{C}_t^u \tilde{\mathbf{u}}_t) + \mathbf{C}_K^b \tilde{\mathbf{b}}_K]$ , where  $J^p := \sum_{t=0}^{K-1} c_t(\mathbf{b}_t^p, \mathbf{u}_t^p) + c_K(\mathbf{b}_K^p)$ , and we assume continuity of the cost function, and  $\mathbf{C}_t^b = \partial c_t(\mathbf{b}, \mathbf{u}) / \partial \mathbf{b}|_{\mathbf{b}_t^p, \mathbf{u}_t^p}$ ,  $\mathbf{C}_t^u = \partial c_t(\mathbf{b}, \mathbf{u}) / \partial \mathbf{u}|_{\mathbf{b}_t^p, \mathbf{u}_t^p}$ , and  $\mathbf{C}_K^b = \partial c_K(\mathbf{b}) / \partial \mathbf{b}|_{\mathbf{b}_K^p}$ . Thus,  $\tilde{J} := J - J^p$  is the error occurred in cost function by our approximation scheme.

*Feedback controller:* As mentioned before, we assume the search is over linear feedback policies, which is a valid assumption for locally controlling a linearized model around a nominal trajectory. Our design, based on the separation principle, supposes the existence of an LQR controller to track and stabilize the trajectory of states around the nominal designed trajectory. Thus,  $\tilde{\mathbf{u}}_t = -\mathbf{L}_t(\tilde{\mathbf{x}}_t - \mathbf{x}_t^p)$ . Note, although we are working with the linearized system, the original system is not a linear system and certainty equivalence is only valid with strict limitations on the nonlinearities. Therefore, we cannot replace  $\tilde{\mathbf{x}}_t$  with  $\mathbf{x}_t$  in the control law without proofs and considerations.

**Assumption 1** (Assumption on calculation of summations). *We assume in the rest of this section for simplicity of our formulas in any sum such as  $\sum_{\tau=t_1}^{t_2} [\text{function}(\tau)]$  that  $t_1 \leq t_2$ , where function is a general function. In other words, function( $\tau$ ) is only evaluated for  $t_1 \leq \tau \leq t_2$ . Otherwise, it is not calculated, and the summation is zero.*

**Lemma 1. Estimation Error Propagation** *Define  $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \hat{\mathbf{x}}_t$  for  $t \geq 0$  to be the estimation error. Then, for  $t \geq -1$  the non-recursive estimation error propagation,  $\tilde{\mathbf{x}}_{t+1}$ , in terms of the independent variables, including process and measurement noises and the initial state error  $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}_0^p$  can be written as follows:*

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{F}}_{0:t} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t \tilde{\mathbf{F}}_{s+1:t} (\mathbf{U}_{s+1} \mathbf{G}_s \boldsymbol{\omega}_s - \mathbf{K}_{s+1} \mathbf{M}_{s+1} \boldsymbol{\nu}_{s+1}), \quad (11)$$

where  $\mathbf{U}_{t+1} := \mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}$ ,  $t \geq 0$ ,  $\mathbf{F}_t := \mathbf{U}_{t+1} \mathbf{A}_t$ ,  $t \geq 0$ , and  $\tilde{\mathbf{F}}_{t_1:t_2} := \Pi_{t=t_1}^{t_2} \mathbf{F}_t$ ,  $t_2 \geq t_1 \geq 0$  and otherwise, it is identity matrix.

**Lemma 2. State Error Propagation** Let state error be  $\tilde{\mathbf{x}}_t = \mathbf{x}_t - \mathbf{x}_t^p$  for  $t \geq 0$ . Then, for  $t \geq -1$  the non-recursive state error propagation,  $\tilde{\mathbf{x}}_{t+1}$ , in terms of the independent variables, including process and measurement noises and the initial state error  $\tilde{\mathbf{x}}_0$  can be written as follows:

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{D}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t \tilde{\mathbf{D}}_{s,t}^{\omega} \omega_s - \sum_{s=0}^{t-1} \tilde{\mathbf{D}}_{s+1,t}^{\nu} \nu_{s+1}, \quad (12)$$

where  $\tilde{\mathbf{F}}_{t+1}^{\mathbf{x}_0} := \mathbf{L}_{t+1} \tilde{\mathbf{F}}_{0:t}^{\mathbf{x}_0}$ ,  $t \geq -1$ ,  $\tilde{\mathbf{F}}_{s,t+1}^{\omega} := \mathbf{L}_{t+1} \tilde{\mathbf{F}}_{s+1:t}^{\omega} \mathbf{U}_{s+1} \mathbf{G}_s$ ,  $t \geq 0$ ,  $t \geq s \geq 0$ ,  $\tilde{\mathbf{F}}_{s+1,t+1}^{\nu} := \mathbf{L}_{t+1} \tilde{\mathbf{F}}_{s+1:t}^{\nu} \mathbf{K}_{s+1} \mathbf{M}_{s+1}$ ,  $t \geq 0$ ,  $t \geq s \geq 0$ ,  $\mathbf{D}_t := \mathbf{A}_t - \mathbf{B}_t \mathbf{L}_t$ ,  $t \geq 1$ ,  $\mathbf{D}_0 := \mathbf{A}_0$  and  $\tilde{\mathbf{D}}_{t_1:t_2} = \Pi_{t=t_1}^{t_2} \mathbf{D}_t$ ,  $t_2 \geq t_1 \geq 0$  otherwise, it is identity matrix. Moreover,  $\tilde{\mathbf{F}}_{s,r,t}^{\omega} := \tilde{\mathbf{D}}_{r+1:t} \mathbf{B}_r \tilde{\mathbf{F}}_{s,r}^{\omega}$ ,  $t-1 \geq s \geq 0$ ,  $t \geq r \geq s+1$ ,  $t \geq 1$ ,  $\tilde{\mathbf{F}}_{s+1,r,t}^{\nu} := \tilde{\mathbf{D}}_{r+1:t} \mathbf{B}_r \tilde{\mathbf{F}}_{s+1,r}^{\nu}$ ,  $t-1 \geq s \geq 0$ ,  $t \geq r \geq s+1$ ,  $t \geq 1$ ,  $\tilde{\mathbf{D}}_t^{\mathbf{x}_0} := \tilde{\mathbf{D}}_{0:t}$  and  $\sum_{r=1}^t \tilde{\mathbf{D}}_{r+1:t} \mathbf{B}_r \tilde{\mathbf{F}}_r^{\mathbf{x}_0}$ ,  $t \geq -1$ ,  $\tilde{\mathbf{D}}_{s,t}^{\omega} := \tilde{\mathbf{D}}_{s+1:t} \mathbf{G}_s + \sum_{r=s+1}^t \tilde{\mathbf{F}}_{s,r,t}^{\omega}$ ,  $1 \leq s \leq t-1$ ,  $t \geq 1$ ,  $\tilde{\mathbf{D}}_{t,t}^{\omega} := \tilde{\mathbf{D}}_{t+1:t} \mathbf{G}_t = \mathbf{G}_t$ ,  $t \geq 0$ ,  $\tilde{\mathbf{D}}_{0,t}^{\omega} := \sum_{r=1}^t \tilde{\mathbf{F}}_{0,r,t}^{\omega}$ ,  $t \geq 1$ , and  $\tilde{\mathbf{D}}_{s+1,t}^{\nu} := \sum_{r=s+1}^t \tilde{\mathbf{F}}_{s+1,r,t}^{\nu}$ ,  $0 \leq s \leq t-1$ ,  $t \geq 1$ .

**Lemma 3. Control Error Propagation** Let control error be  $\tilde{\mathbf{u}}_t = \mathbf{u}_t - \mathbf{u}_t^p$  for  $t \geq 0$ . Then, for  $t \geq -1$  the non-recursive control error propagation,  $\tilde{\mathbf{u}}_{t+1}$ , in terms of the independent variables, including process and measurement noises and the initial state error  $\tilde{\mathbf{x}}_0$  can be written as follows:

$$\tilde{\mathbf{u}}_{t+1} = -\mathbf{L}_{t+1}^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 - \sum_{s=0}^t \mathbf{L}_{s,t+1}^{\omega} \omega_s - \sum_{s=0}^t \mathbf{L}_{s,t+1}^{\nu} \nu_{s+1},$$

where  $\mathbf{L}_{t+1}^{\mathbf{x}_0} := \mathbf{L}_{t+1} \tilde{\mathbf{D}}_t^{\mathbf{x}_0} - \tilde{\mathbf{F}}_{t+1}^{\mathbf{x}_0}$ ,  $t \geq 1$ ,  $\mathbf{L}_{s,t+1}^{\omega} := \mathbf{L}_{t+1} \tilde{\mathbf{D}}_{s,t}^{\omega} - \tilde{\mathbf{F}}_{s,t+1}^{\omega}$ ,  $t \geq 1$ ,  $t \geq s \geq 0$ ,  $\mathbf{L}_{s,t+1}^{\nu} := \tilde{\mathbf{F}}_{s+1,t+1}^{\nu} - \mathbf{L}_{t+1} \tilde{\mathbf{D}}_{s+1,t}^{\nu}$ ,  $t \geq 1$ ,  $t-1 \geq s \geq 0$  and  $\mathbf{L}_{t,t+1}^{\nu} := \tilde{\mathbf{F}}_{t+1,t+1}^{\nu}$ .

**Lemma 4. Observation Error Propagation** Let observation error be  $\tilde{\mathbf{z}}_t = \mathbf{z}_t - \mathbf{z}_t^p$  for  $t \geq 1$ . Then, for  $t \geq 0$  the non-recursive observation error propagation,  $\tilde{\mathbf{z}}_{t+1}$ , in terms of the independent variables, including process and measurement noises and the initial state error  $\tilde{\mathbf{x}}_0$  can be written as follows:

$$\tilde{\mathbf{z}}_{t+1} = \tilde{\mathbf{H}}_{t+1}^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t \tilde{\mathbf{H}}_{s,t+1}^{\omega} \omega_s + \sum_{s=0}^t \tilde{\mathbf{H}}_{s+1,t+1}^{\nu} \nu_{s+1}, \quad (13)$$

where  $\tilde{\mathbf{H}}_{t+1}^{\mathbf{x}_0} := \mathbf{H}_{t+1} \tilde{\mathbf{D}}_t^{\mathbf{x}_0}$ ,  $t \geq 0$ ,  $\tilde{\mathbf{H}}_{s,t+1}^{\omega} := \mathbf{H}_{t+1} \tilde{\mathbf{D}}_{s,t}^{\omega}$ ,  $t \geq 0$ ,  $t \geq s \geq 0$ ,  $\tilde{\mathbf{H}}_{s+1,t+1}^{\nu} := -\mathbf{H}_{t+1} \tilde{\mathbf{D}}_{s+1,t}^{\nu}$ ,  $t \geq 1$ ,  $t-1 \geq s \geq 0$ , and  $\tilde{\mathbf{H}}_{t+1,t+1}^{\nu} := \mathbf{M}_{t+1}$ ,  $t \geq 0$ .

**Lemma 5. Belief Error Propagation** Let belief error be  $\tilde{\mathbf{b}}_t = \mathbf{b}_t - \mathbf{b}_t^p$  for  $t \geq 0$ . Then, for  $t \geq 0$  the non-recursive belief error propagation in terms of the independent variables, including process and measurement noises and the initial state error  $\tilde{\mathbf{x}}_0$  can be written as follows:

$$\tilde{\mathbf{b}}_t = \tilde{\mathbf{T}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-1} \tilde{\mathbf{T}}_{s,t}^{\omega} \omega_s + \sum_{s=1}^t \tilde{\mathbf{T}}_{s,t}^{\nu} \nu_s, \quad (14)$$

where  $\tilde{\mathbf{T}}_{t_1:t_2}^{\mathbf{b}} := \Pi_{t=t_1}^{t_2} \tilde{\mathbf{T}}_t^{\mathbf{b}}$ ,  $0 \leq t_1 \leq t_2$  otherwise, it is identity matrix,  $\tilde{\mathbf{T}}_t^{\mathbf{x}_0} := \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_t^{\mathbf{x}_0} + \sum_{s=0}^{t-2} (-\tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}} \mathbf{T}_{s+1}^{\mathbf{u}} \mathbf{L}_{s+1}^{\mathbf{x}_0} + \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}} \mathbf{T}_s^{\mathbf{z}} \tilde{\mathbf{H}}_{s+1}^{\mathbf{x}_0})$ ,  $t \geq 1$ ,  $\tilde{\mathbf{T}}_0^{\mathbf{x}_0} := \mathbf{0}$ ,  $\tilde{\mathbf{T}}_{s,t}^{\omega} := \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{s,t}^{\omega} + \sum_{r=s}^{t-2} (-\tilde{\mathbf{T}}_{r+2:t-1}^{\mathbf{b}} \mathbf{T}_{r+1}^{\mathbf{u}} \mathbf{L}_{s,r+1}^{\omega} + \tilde{\mathbf{T}}_{r+1:t-1}^{\mathbf{b}} \mathbf{T}_r^{\mathbf{z}} \tilde{\mathbf{H}}_{s,r+1}^{\omega})$ ,  $t \geq 2$ ,  $t-2 \geq s \geq 0$ ,  $\tilde{\mathbf{T}}_{t-1,t}^{\omega} := \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{t-1,t}^{\omega}$ ,  $t \geq 1$ ,  $\tilde{\mathbf{T}}_{s,t}^{\nu} := \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{s,t}^{\nu} + \sum_{r=s-1}^{t-2} (-\tilde{\mathbf{T}}_{r+2:t-1}^{\mathbf{b}} \mathbf{T}_{r+1}^{\mathbf{u}} \mathbf{L}_{s-1,r+1}^{\nu} + \tilde{\mathbf{T}}_{r+1:t-1}^{\mathbf{b}} \mathbf{T}_r^{\mathbf{z}} \tilde{\mathbf{H}}_{s,r+1}^{\nu})$ ,  $t \geq 2$ ,  $t-1 \geq s \geq 1$ , and  $\tilde{\mathbf{T}}_{t,t}^{\nu} := \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{t,t}^{\nu}$ ,  $t \geq 1$ .

**Theorem 1. Cost Function Error** Let cost function error be  $\tilde{J} = J - J^p$  for  $t \geq 0$ . Given that process and observation noises are zero mean i.i.d. and are mutually independent from each other and the initial belief, under a first-order approximation, the stochastic cost function is dominated by the nominal part of the cost function. Moreover the expected first-order error is zero, i.e.,  $\tilde{J} = 0$ .

*Discussion:* Theorem 1 shows that under the assumption that the error resulting from linearizations are small enough (i.e., the linearizations are valid), the error in our cost function is independent from the stochastic terms, and it is inconsequential. Therefore, under these conditions, the original cost function is dominated by the nominal part of the cost function. In practice, the time horizon is only chosen large enough to find feasible solutions, and the linearization error is negligible. Thus, choosing the underlying linearization trajectory (or equivalently, the control actions corresponding to the nominal trajectory) as the optimization variables, the optimal underlying trajectory can be calculated. Moreover, whenever the accumulated error of the

belief approximation under these assumption gets higher than a threshold, the problem restarts and replanning follows. It is important to note that the separation principle is the central idea behind the method, since it provides the mechanism to design the controller and estimator separate from each other. The T-LQG method, utilizes this theory and the dependence of the LQG on the underlying trajectory, and through a coupled design of trajectory with the provided LQG methodology and a standard optimization problem, finds the LQG controller with the best performance.

Hence, utilizing the separation theorem makes all these arguments possible, because, there is a feedback policy that changes the sensitivity matrices of the error propagations and stabilizes the system. Otherwise, the control error grows intractably and becomes an independent uncontrollable variable, thus, there is no guarantee for our statements. Moreover, since the LQR feedback gain is only dependent on the underlying trajectory (and is independent from the estimation by separation), the trajectory optimization is performed without the feedback, and the controller is designed on top of the policy.

## VI. COMPARISON OF METHODS

In this section, we provide a comparison between state-of-the-art belief space planning approaches from a methodology and computational complexity perspective. We make occasional references to the following methods: a) LQG-MP [4], b) iLQG-based method [5], c) SELQR [6] d) the method utilizing MLO [7], e) the non-Gaussian Receding Horizon Control (RHC)-based method [9], f) the non-Gaussian observation covariance reduction method [10], g) FIRM [21], and g) the point-based POMDP solvers [8], [12], [13], [14]. Table I summarizes the key differences between the methods.

As reflected in the table, a central difference between these methods is the way the system and observation equations are linearized. After linearization of the equations, the corresponding Jacobians become coupled with the trajectory. Therefore, if the underlying linearization trajectory is a sequence of fixed points, the Jacobians become constant matrices for the entire optimization, and the structure of the system models (on which depends many other properties of the system, such as sensitivity of the observations, controllability, reachability, etc.) essentially become fixed, untouchable, and, more importantly, un-exploitable for the optimization purposes. Table I summarizes the capability of methods on using this feature. As noted, our method fully exploits this property and finds the best linearization trajectory among the methods. Moreover, assumptions on observations in our method are inconsequential and the observation model is exploited with its best capacity. Most importantly, the computational complexity of T-LQG is the lowest among all.

Note: the computational complexity only reflects the calculations of the core problems for belief space planning in each method. For obstacle-avoidance, each method has a different approach, which is out of this discussion and can be further detailed in a pure motion-planning scope. The information in table I and the calculations regarding the computational complexity are estimated to the best of our knowledge.

Next, we provide a brief summary of the methods and afterwards, we elaborate more on the key methodological aspects and differences.

### A. An Overall Summary of the Methods

**a) LQG-MP [22]** In this method, several paths generated by RRT planner are taken as initial nominal trajectories, and the system equations are linearized around those trajectories. An LQG tracker is designed along each trajectory and the control sequences are compared based on an obstacle-avoidance performance measure. The trajectory with the best performance is selected as the nominal trajectory to track and the LQG tracker corresponding to that trajectory is chosen as the policy to implement.

**b) iLQG-based method [5]** In this method, the iteration begins with an initial guess trajectory that is obtained using a method such as RRT, around which the system equations, belief dynamics and value function are linearized. Then, the value function is evaluated by backward run along the nominal trajectory. Next, the noiseless belief dynamics is used to forward propagate the belief using the policy that was found in the backward propagation. This gives a new nominal trajectory for the next iteration of the algorithm. The iterations are coupled with an adaptive line search method and continue until convergence to a locally optimal policy.

**c) SELQR [6]** In these methods, the iteration idea of the iLQG-based methods is extended by a better choice of the underlying linearization trajectory. Starting with an initial guess, the forward and backward iterations are both done over that trajectory, then sum of the costs of forward and backward iterations at every time step is obtained. This defines a minimization problem whose result provides the nominal trajectory for linearization in the next iteration.

**d) MLO [7]** This method is also based on the LQG methodology. The mean update equation in the (extended) Kalman filtering equation requires an observation (or an assumption over the observations) to calculate the innovation term, whereas the covariance update equation only depends on an underlying trajectory (this trajectory can either come from the true mean update during the estimation, or can be a fixed nominal trajectory). Moreover, the mean update equations are tied to the covariance update, as well. In this method, in order to perform the mean update, the future observations are assumed to be the most-likely observations (which correspond to the noiseless observations predicted by the observation model). The system equations are linearized around such mean updates at each step. An optimization problem with a quadratic cost is defined to obtain the desired trajectory, and an LQR controller is used to reject the disturbances.



TABLE I  
COMPARISON OF BELIEF SPACE PLANNING METHODS ON IMPORTANT ISSUES.

	Planning as an Optimization	Linearization Trajectory (Exploitable for Optimization)	Planning Observations	Computational Complexity	Convergence Rate
LQG-MP [4]	None	RRT trajectories (No)	—	$O(n_r K n^3)$	—
iLQG-based [5]	DP	Fixed at each iteration (No)	Stochastic observations	$O(K n^6)$	Second order (line-search tuning)
SELQR [6]	DP	Fixed at each iteration (No)	MLO	$O(K n^6)$	Second order
MLO [7]	NLP	Predicted mean update (Yes)	MLO	$O(n_{tr}(K n^3 + k n^2))$ or $O(n_{tr}(K n^3 + k n^2) + K n^6)$	SQP rate
Non-Gaussian RHC-Based [9]	Convex	Linear propagation of initial estimate (Yes)	MLO	$O(NK(K n^3 + N n^2))$	$\Omega((N + K n) \log(\frac{1}{\epsilon}))$
Non-Gaussian Obs. Cov. Reduction [10]	Convex	Linear propagation of initial estimate (Yes)	Predicted ensemble of observation particles	$O(K n^3 + N n^2)$	$\Omega(K n \log(\frac{1}{\epsilon}))$
T-LQG	NLP	Non-linear propagation of initial estimate (Yes)	—	$O(K n^3)$	Second order

- We assume the size of vectors  $\mathbf{x}$ ,  $\mathbf{u}$  and  $\mathbf{z}$  are all  $O(n)$ , and  $K$  is planning horizon
- $n_r$  is the number of RRT paths generated in [4]
- For the method of [7],  $n_{tr}$  is the number of transcription steps in the direct transcription;  $k$  is the number of unit vectors pointing in the desired directions to minimize the covariance in; the second computational complexity is valid if the B-LQR is also used, otherwise, the first complexity is more accurate
- $N$  is the number of samples,  $\epsilon$  is the convergence error
- *Convergence Rate* is the number of calls needed to the oracle to converge using a method such as center of gravity
- *DP* is Dynamic Programming
- *Second order rate* is the general rate for Newton-like methods

**e) Non-Gaussian RHC-Based [9]** In this method, the most-likely observation method is adapted for a linear system and observation models with Gaussian noises, where the observation noise covariance is state-dependent. The representation of the belief is replaced with that of a particle filter, and the noise models are utilized to obtain the dynamics of the particle weights. An optimization problem is defined and convexified to obtain the optimal nominal trajectory. The policy is implemented with an RHC strategy closing the feedback loop in the execution.

**f) Non-Gaussian Observation Covariance Reduction [10]** In this method, system equations are linearized around an initial nominal trajectory, however, the observation model is linearized around the noiseless propagation of the initial estimate. The main contribution of this work is to exploit the observation uncertainty and define an optimization problem which is easy to solve, avoids performing the filtering equations and yields similar trajectories as the other belief space planning methods. Moreover, the belief has a particle filter representation where no assumptions on the noise distributions are assumed.

**g) FIRM [21]** Feedback Information RoadMap is an offline POMDP planner that solves an MDP over a graph with finite number of nodes in the belief space. Therefore, the solution over the graph is provided based on the dynamic programming. As mentioned before, in the point-based POMDP solvers where the probability of reaching to a belief node is zero and whence the solution is only valid for the initial belief. Unlike the point-based solvers, the key point in FIRM is stabilizing the belief over a belief node in the graph with high probability utilizing an stabilizer controller. This, also breaks the curse of history. Currently the abstracted algorithm of FIRM has been implemented utilizing the LQG methodology and is called the SLQG-FIRM.

**h) Point-Based POMDP Solvers [8], [12], [13], [14]** The POMDP problem was introduced in 1971 in [8], with an algorithm to obtain the exact optimal solution using the alpha-vectors. The algorithm then evolved into an anytime algorithm in 2003 in [12], introducing the point-based POMDP solvers. This method has been the foundation for the majority of research in the POMDP field [13]. There has been many successes in finding solving POMDP benchmark problems with low CPU-times. Even the latest advancements in the field, such as [14], suffer from multiple limitations. For instance, the scalability with time-horizon seems to be a fundamental limitation that might be difficult to just overcome. Ad-hoc solutions to reduce the planning time horizon to local planning (which are much lower than enough for reaching the goal region) and replanning every few steps is not a feasible a solution for practical problems. This is also much different from the Model Predictive Control (MPC) strategy, where the planning horizon is chosen to be large enough to reach to a goal region.

An issue of POMDP solvers is that the search over the belief space is reduced to a discrete set of belief nodes (either through a discretization of the underlying spaces or through random sampling of continuous spaces and building a decision tree over belief samples). In these methods, the probability of re-visiting any particular discrete belief node in the tree (other than the root) is equal to zero. Thus, the solution is *only* valid for the initial belief. A way of overcoming this limitation is to perform a continuous branching or an exact Monte-Carlo, where for every infinitesimal change in a higher level of the tree, there is an exponentially increased number of belief nodes in the next level, which brings back the original highly computational theoretical solution of POMDPs. It is only in such a case where the solution is comparable to methods such as ours, where the search occurs over a continuous set of beliefs—thus, the replanning does not need to happen every single step. For this reason, our solution is valid for a much longer horizon and for a belief space region far more considerable than results from point-based POMDP solvers.

Moreover, in T-LQG, by tracking the nominal and true belief during online implementation, whenever the optimality deviation is more than the tolerable threshold, replanning occurs, which is essentially impractical in true long-horizon POMDP solvers. FIRM [21] on the other hand, provides an offline approach to tackle the original POMDP problem by solving the dynamic programming over a graph in the belief space and breaking the curse of history; but, to get closer to optimality, more FIRM nodes need to be sampled in the offline design.

## B. Comparison on Important Issues

In this section we discuss more on the key differences between methods (a-f). Since, POMDPs were already discussed before, we avoid further discussions in here. Moreover, since FIRM is an offline planner, we do not compare with FIRM either. We explain how the linearization trajectory is different in these methods and how that leads to major differences in the algorithms. Moreover, we explain that a critical difference is the assumptions on the observation process during the planning stage. Note that, likewise methods (a-d), our current paper deals with Gaussian beliefs.

*Optimization problem:* In (a), the least-cost trajectory is chosen among a finitely generated initial trajectories, hence the underlying trajectory is not optimized or morphed. In (b), the underlying trajectory is morphed through an iteration mentioned as above coupled with tuning of a line-search method. Thus, the algorithm does not involve an explicit optimization problem that can be solved via an NLP solver. Rather, the whole method involves the inner mechanisms of an optimization problem. The method is essentially a dynamic-programming-based algorithm. Therefore, the merits of an explicit NLP problem cannot be exploited. In (c), the approach is similar to (b), with a difference that there is also an intermediary optimization problem in each back and forth iteration to find a better nominal trajectory for the next iteration. However, the whole algorithm is essentially similar in content to the method of (b) and the problem lacks a standard optimization problem. In (d) the trajectory optimization problem is posed as an optimization problem that can be solved using SQP. In (e) and (f), the problem is convexified and can be solved using any convex optimizer. Our method also presents the planning problem as an NLP program that can be

solved by a generic NLP solver. Presenting the problem as an standard optimization problem has the advantage that it can be solved using various tools and softwares in the optimization and control theory, increasing the efficiency of implementation and availing the usage of advanced techniques developed in those fields to obtain smoother solutions. Moreover, it does not require delving into the details of optimization problem solving.

*Linearization of the system equations:* As pointed above, this is a central difference between the methods. Essentially, an LQG planner with a form of Kalman filtering for estimation requires a nominal trajectory to linearize the system equations. As mentioned before, after linearization of the equations, the Jacobians correspond to the specific trajectory. Therefore, if the underlying linearization trajectory is not a variable of optimization, the Jacobians become constant matrices for the entire optimization and un-exploitable for the optimization purposes. This is what happens in methods (a), (b), and (c). In these methods, although, the underlying linearization changes during the whole algorithm; however, the linearization of the equations is decoupled from the manipulations and deformations of the underlying trajectory, and they happen sequentially with respect to each other. In (e), the model is linear to begin with. On the other hand, in (d) and (f), the linearization is coupled with the manipulation of the trajectory. However, methods are different; in (e), the linearization is done over the predicted mean of the belief (whose updates are possible based on most-likely observations assumption), but in (f), the underlying trajectory for the observation model is the parametrized possible trajectories obtained from the noiseless propagation of the initial estimate, and the trajectory for system equations is based on an initial guess. In this paper, the underlying linearization trajectory is the optimization variable.

*Assumptions on the observation during planning:* The observation distributions are calculated in the methods (a) and (b) based on the LQG methodology; however, in (a), the observations do not contribute to the designed trajectory. In (b), the stochasticity of the observations (distributed with a Gaussian density) is exploited in the dynamic programming equations. In (c), (d) and (e), the observations are most-likely observations. In (f), an ensemble of observation particles for the entire path is generated and their predicted covariance is reduced as an objective in the optimization problem. In the current work, any assumption on the observations is inconsequential and the planning is performed only utilizing the trajectory-dependent Jacobian of the observation model.

*Optimization problem time-complexity for obstacle-free case:* As mentioned, we provide the time complexity for methods (a), (c), (d) and (e) to the best of our knowledge. Let us assume for simplicity that the size of  $\mathbf{x}$ ,  $\mathbf{u}$ , and  $\mathbf{z}$  vectors are all  $O(n)$ . The computation time in method a is on finding as many RRT plans as possible, therefore, since this method is not constructing a path the quality of solution can be significantly poorer than the other methods. If  $n_r$  number of RRT paths are taken, then it would take  $O(n_r K n^3)$ , however, there is no issue of convergence in here. In (b) and (c), the computation complexity is  $O(K n^6)$  with a second-order convergence rate of Newton-like methods to a locally optimal solution. However, method (c), converges faster than (b), as stated in (c). Method (d), takes  $O(n_{tr}(K n^3 + k n^2))$ , where  $n_{tr}$  is the number of transcription steps in the direct transcription, and  $k$  is the number of unit vectors pointing in  $k$  directions to minimize the covariance in their algorithm. In method (e), utilizing a common method, such as center of gravity for convex optimization [23] to obtain a *globally optimal* solution with  $\epsilon$  confidence and  $N$  number of samples, the algorithm requires  $O(NK(K n^3 + N n^2))$  computations and the convergence needs  $\Omega((N + K n) \log(1/\epsilon))$  calls to the oracle. In method (f), the convex problem requires  $O(K n^3 + N n^2)$  computations and  $\Omega(K n \log(1/\epsilon))$  calls to the oracle [24]. Our current method requires  $O(K n^3)$  computations and the convergence rate is the rate for the particular gradient-descent method utilized. For instance, a Newton-like method converges at a second-order rate.

### C. Comparison on Other Issues

In this section, we point out some other differences between the methods that are of less importance than the previous points.

*Parametrization of the belief:* In a Gaussian model, it is assumed belief is fully parametrized by two parameters. In a non-Gaussian method this assumption is lifted and typically replaced by a number of samples taken from the belief. Methods (a-d) assume Gaussian beliefs and methods (e) and (f) assume a non-Gaussian representation of the belief. In (e), the particle weights become part of the optimization variables, whereas in (f), the samples or their weights are not variables and the optimization shows more scalability. The Gaussianity assumption can be a valid assumption in the vicinity of a nominal trajectory. Our current paper, deals with Gaussian beliefs. The Gaussianity assumption can be a valid assumption in the vicinity of a nominal trajectory. Therefore, a method that can better stabilize around a nominal trajectory can better exploit this feature. In particular, our method with a better promised path and coupled with feedback controller fully exploits this feature, making the Gaussianity assumption more valid.

*Form of the system equations:* In all methods except (e), the system and observation models are non-linear. In (e), both equations are linear. Moreover, in (d), the process noise is not included.

*Replanning policy:* In (a), (b), and (c), replanning is not discussed. In (d) it is based on the mean deviation from a predicted mean. In (e), a combination of KL divergence and RHC strategy is assumed, and in (f), at every stage replanning is performed. In our current method, a symmetric distance based on KL divergence is utilized.

*Initialization of the optimization problem:* The initial guess in (a), (b) and (c) is based on an RRT or a similar planner. However, in (a), essentially there is no construction of the path, whereas in the other methods, a path is constructed. In (b),

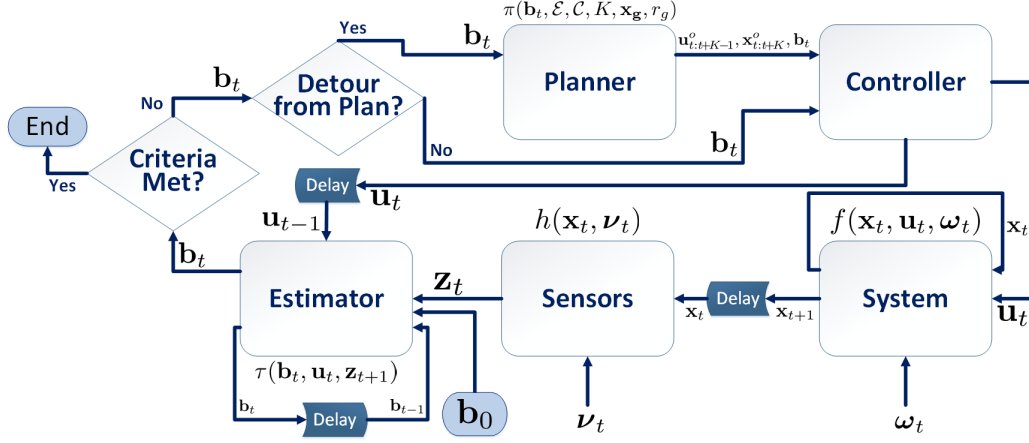


Fig. 1. The overall feedback control loop.

it requires an adaptive line-search and a feasible initial path to ensure convergence. In (d), the optimization yields a locally optimal solution. In (e) and (f) the convex planning problems require no initialization and the planning results are global in the sense of the defined optimization problem. In the current paper, the non-linear optimization requires initialization based on an RRT or a similar planner, and the result of the optimization is a locally optimal path.

*Non-convex constraints:* In (a), a performance measure based on obstacle avoidance is defined to compare the safety of the resulting policies. In (b), (c) and (f), a cost function is added to the optimization problem. In (d), obstacles are not considered. In (e), mixed integer programming and chance constraints are used to avoid constraints. In terms of the computation complexity, among methods (b-f), the methods (b), (c) and (f) have lower computation complexities. In the current paper, an extended version of the method in (f) is introduced, which provides safety based on the barrier functions.

## VII. SIMULATION RESULTS

In this section, we provide our simulation results to show the performance of T-LQG. Our simulations are performed in MATLAB 2016a with a 2.90 GHz CORE i7 machine with dual core technology and 8 GB of RAM. We use the MATLAB's fmincon solver to solve the NLP problem. First, we provide the overall algorithm and the overall control loop. Then, we investigate several situations in which the environment is obstacle-free. We perform 9 simulations for a KUKA youBot base model, with 9 different observation models including models adapted from the literature. Then, we perform a simulation in a complex environment with many obstacles. We conduct this scenario for two different initial trajectories and compare the results. In each scenario, we show the initial trajectory used to initialize the optimization problem along with the optimized output trajectory.

*Implementation:* The overall control loop is shown in Fig. 1, and the overall T-LQG algorithm is reflected in Algorithm 1. As it is seen in Fig. 1 and Alg. 1, the planning problem starts with the supply of an initial belief and ends whenever the

---

### Algorithm 1: T-LQG

---

**Input:** Initial belief  $\mathbf{b}_0$ , Goal region  $\mathbf{B}_{r_g}(\mathbf{x}_g)$ , Planning horizon  $K$ , Obstacle parameters  $(\mathcal{E}, \mathcal{C})$

```

1  $t \leftarrow 0$ ;
2 while  $\mathcal{P}(\mathbf{b}_t, r_g, \mathbf{x}_g) \leq p_g$  do
3   if  $d(\mathbf{b}_t, \mathbf{b}_t^o) > d_{th}$  or  $t == 0$  or  $t == K$  then
4     Optimal Trajectory:  $\{\mathbf{u}_{0:K-1}^o, \mathbf{x}_{0:K}^o\} \leftarrow \pi(\mathbf{b}_0, \mathcal{E}, \mathcal{C}, K, \mathbf{x}_g, r_g)$ ;
5      $t \leftarrow 0$ ;
6   end
7   else
8     Policy Function:  $\hat{\mathbf{x}}_t \leftarrow \mathbb{E}[\mathbf{b}_t]$ ,  $\mathbf{u}_t \leftarrow -\mathbf{L}_t(\hat{\mathbf{x}}_t - \mathbf{x}_t^o) + \mathbf{u}_t^o$ ;
9     Execution:  $\mathbf{x}_{t+1} \leftarrow f(\mathbf{x}_t, \mathbf{u}_t, \omega_t)$ ;
10    Perception:  $\mathbf{z}_{t+1} \leftarrow h(\mathbf{x}_{t+1}, \nu_{t+1})$ ;
11    Estimation:  $\mathbf{b}_{t+1} \leftarrow \tau(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_{t+1})$ ;
12     $t \leftarrow t + 1$ ;
13  end
14 end

```

---

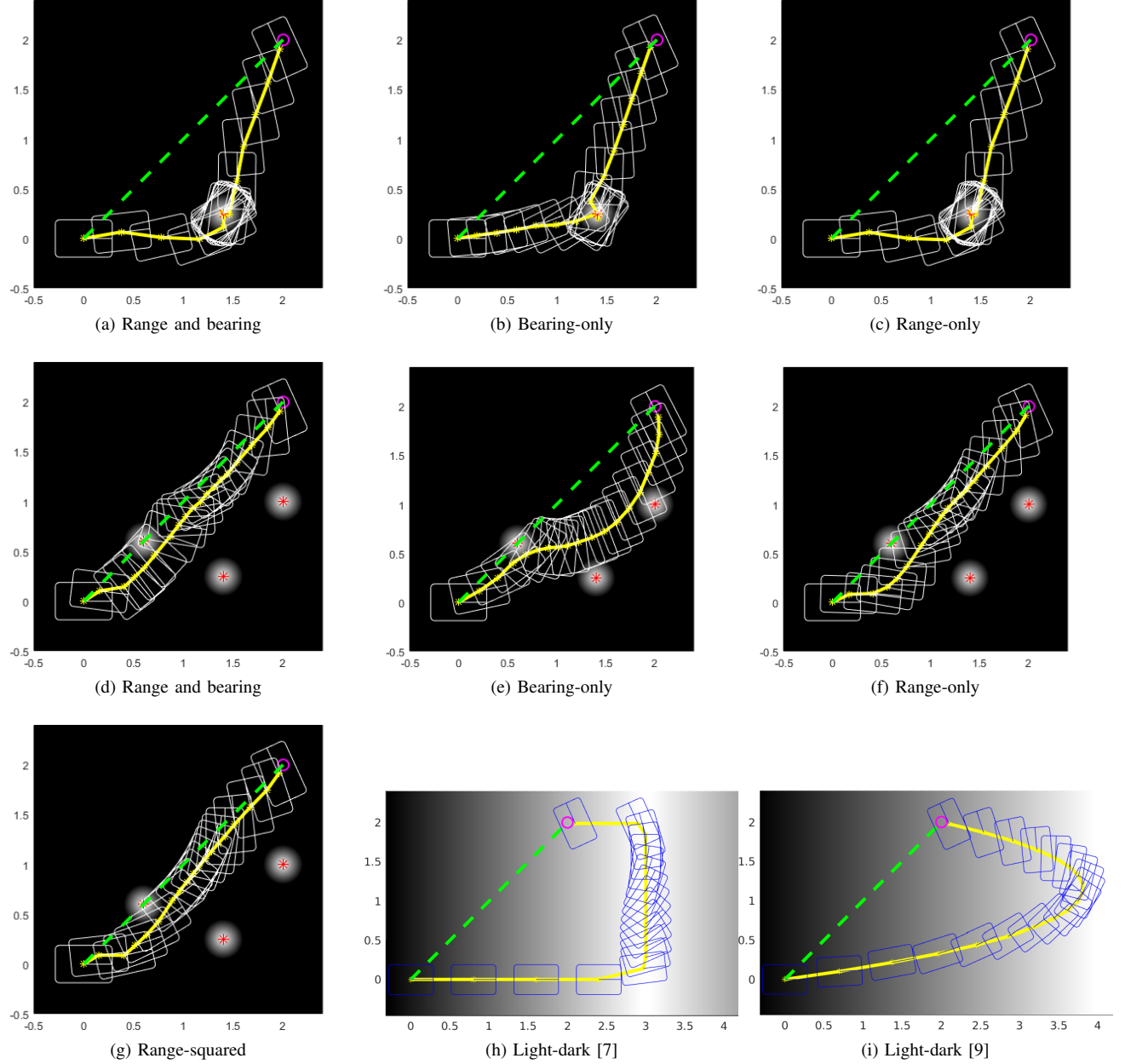


Fig. 2. Simulation results for obstacle-free situation with different observation models. The information is color-coded. Lighter means less noisy observations. The dashed green line represents the initial trajectory; the solid yellow line shows the optimized trajectory. In all cases,  $\hat{\mathbf{x}}_0 = (0, 0, 0)$ ,  $\mathbf{x}_g = (2, 2, 2)$ , and  $r_g = 0.1$ .

probability of reaching the goal region is greater than a predefined threshold  $p_g > 0$ . The planner  $\pi$  is fed the initial belief  $\mathbf{b}_0$ , the obstacle parameters  $(\mathcal{E}, \mathcal{C})$ , planning horizon  $K$ , a goal state  $\mathbf{x}_g$ , the goal region radius  $r_g$ , and other parameters, such as system equations. The resultant planned trajectory is provided to the controller, whose output is the policy function. The policy is executed, a new observation is perceived, and a new belief is obtained. If the distance between the updated belief and the nominal belief  $d(\mathbf{b}_t, \mathbf{b}_t^o) > d_{th}$  is greater than a threshold, or the policy execution is finished but the criteria is not met, the planning algorithm restarts.

*Obstacle-free environment:* Let us use the kinematics equations of KUKA youBot base as described in [25]. Particularly, the state vector can be denoted by a 3D vector,  $\mathbf{x} = [\mathbf{x}_x, \mathbf{x}_y, \mathbf{x}_\theta]^T$ , which describes the position and heading of the robot base, and  $\mathbf{x} \in SO(2)$ . The control consists of the velocities of the four wheels. It can be shown that the discrete motion model can be written as  $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \omega_t) = \mathbf{x}_t + \mathbf{B}\mathbf{u}_t dt + \mathbf{G}\omega_t \sqrt{dt}$ , where  $\mathbf{B}$  and  $\mathbf{G}$  are appropriate constant matrices whose elements depend on the dimensions of the robot as indicated in [26], and  $dt$  is the time-discretization period. The results depicted in Fig. 2 are for different observation models; including the range and bearing; bearing-only and range-only observation from one landmark in cases (a)-(c); and from three landmarks in cases (d)-(f), respectively. In case (g), the observation function is

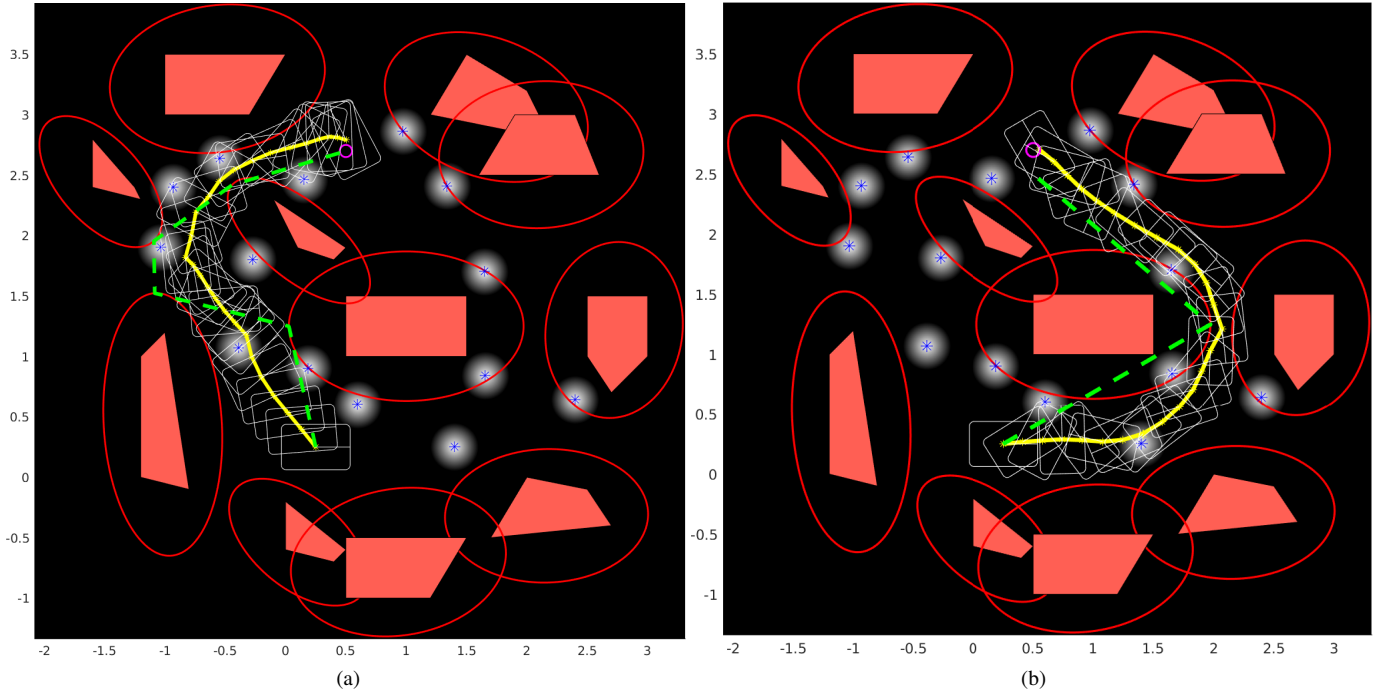


Fig. 3. Simulation results for two different initializations with obstacles. The obstacles are the red solid polygons; the ellipses show the inflated regions around them avoided by the configuration of points that represent the robot (they are also the argument of the Gaussian function in the obstacle cost). In all cases,  $\mathbf{x}_0 = (0.25, 0.25, 0)$ ,  $\mathbf{x}_g = (0.5, 2.7, 2)$ , and  $r_g = 0.1$ . The optimized trajectory in case (b) has a lower overall cost.

changed to the square of the range function. Finally, in cases (h) and (i), the light-dark models of the papers [7] and [9] are adopted. In both cases, the observation functions are linear, and the covariance of the observation noise is state dependent. It is a quadratic function with a minimum at 3 in case (h) and a hyperbolic function with a minimum at  $+\infty$  in (i). More details of these functions can be found in the corresponding papers. Finally as it is noticed in all cases, the optimization is initialized with the trivial straight-line, which is reflected in the figures with the dashed green line, whereas the optimal trajectories are depicted with solid lines.

*Complex environment:* Next, we perform a simulation in an environment full of obstacles for the youBot, with range and bearing observations from several landmarks. Inspired by [27], we model the robot with a configuration of a set of points that represent the balls' centers that cover the body of the robot. In our simulations, only two balls whose radii are proportional to the width of the robot suffice to cover the robot [28]. We find the MVEE of the polygons that are inflated from each vertex to the size of the radius and modify the cost of obstacles to keep the centers of the balls out of the new barriers. As it is seen in Fig. 3, we have initialed the optimization problem with two different initial trajectories obtained using a modified viability graph algorithm (shown with the green dashed lines). It should be noted that there is nothing particular about the initialization algorithm and methods—a planner such as RRT can be used as well, as long as the initialization trajectory is semi-feasible in that it does not pass through the infeasible local minima of the barrier functions. As it seen in this figure, the planning horizon is large (26 steps in case (a) and 25 steps in case (b)), which shows the scalability of T-LQG. The results show that the optimized trajectory (reflected with solid lines) avoids entering the banned regions bordered by the ellipsoids, so that the robot itself avoids colliding with the obstacles. Moreover, the locally optimal trajectory gets closer to the information sources and thereby obtains the best predicted estimation performance. In this scenario, by comparing the cost of the two optimize trajectories, the better of the two (trajectory in Fig. 3b) is chosen as the plan for execution.

### VIII. CONCLUSION

In this paper, we simplified the solution of the belief space planning problem by proposing a scalable method that is backed by theoretical analysis supported by the control literature. Particularly, we proposed a deterministic optimal control problem that can be solved by an NLP solver with  $O(Kn^3)$  computational complexity. The goal of Trajectory-optimized LQG is to find an LQG policy with the best nominal performance. T-LQG achieves this by finding the best underlying linearization trajectory for a non-linear system with a non-linear observation model, utilizing the trajectory-dependent covariance evolution of the Kalman filter given by the dynamic Riccati equations. We could do this by the proper usage of the separation principle that provided us with an LQR controller for a linearized system along that nominal trajectory. We proved that the accumulated error that is resulted by our calculations is deterministic under a first-order approximation and only depends on the linearization error. This can be overcome by either increasing the linearization points or by replanning whenever the deviation from the

planned trajectory is higher than a predefined tolerance. We also extended the method to non-convex environments by adding a cost function to avoid collision with the obstacles. Finally, we performed simulations for a common robotic system with several observation functions in obstacle-free environments, as well as complex narrow passages with obstacles.

In conclusion, while T-LQG and the MLO method of [7] address a similar optimization problem, their theoretical approach is vastly different:

- where MLO uses a heuristic approach, T-LQG uses the separation principle;
- MLO does not have a controller in the design, whereas T-LQG does;
- MLO uses assumptions on the observations to reach the optimization problem, while in T-LQG, assumptions on observations are inconsequential;
- MLO designs a belief-LQR, but T-LQG only requires an LQR on the state;
- MLO starts with an EKF design and linearizes the system equations around the mean update, while T-LQG starts with linearizing the system equations around a nominal trajectory and uses the KF and separation principle to obtain the nominal performance around that trajectory;
- MLO assumes from the beginning that process noise does not exist, and, ultimately, assumes observation noise does not exist either, but in T-LQG, neither of these assumptions exist; and
- while the computational complexity for MLO is  $O(n_{tr}(Kn^3 + kn^2))$  or  $O(n_{tr}(Kn^3 + kn^2) + Kn^6)$ , T-LQG minimizes the complexity to  $O(Kn^3)$ .

Our future works will extend the theory and test the validity of our results for more complex situations.

#### ACKNOWLEDGMENT

This material is based upon work partially supported by NSF under Contract Nos. CNS-1646449 and Science & Technology Center Grant CCF-0939370, the U.S. Army Research Office under Contract No. W911NF-15-1-0279, and NPRP grant NPRP 8-1531-2-651 from the Qatar National Research Fund, a member of Qatar Foundation.

#### REFERENCES

- [1] P. R. Kumar and P. P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1986.
- [2] D. Bertsekas, *Dynamic Programming and Stochastic Control*. Academic Press, 1976.
- [3] P. Kumar *et al.*, “Control: a perspective,” *Automatica*, vol. 50, no. 1, pp. 3–43, 2014.
- [4] J. Van Den Berg, P. Abbeel, and K. Goldberg, “Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.
- [5] J. Van Den Berg, S. Patil, and R. Alterovitz, “Motion planning under uncertainty using iterative local optimization in belief space,” *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.
- [6] W. Sun, J. van den Berg, and R. Alterovitz, “Stochastic extended lqr for optimization-based motion planning under uncertainty,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 437–447, 2016.
- [7] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, “Belief space planning assuming maximum likelihood observatoins,” in *Proceedings of Robotics: Science and Systems (RSS)*, June 2010.
- [8] E. J. Sondik, “The optimal control of partially observable markov processes,” *PhD thesis, Stanford University*, 1971.
- [9] R. Platt, “Convex receding horizon control in non-gaussian belief space,” in *Algorithmic Foundations of Robotics X*. Springer, 2013, pp. 443–458.
- [10] M. Rafieisakhaei, A. Tamjidi, S. Chakravorty, and P. Kumar, “Feedback motion planning under non-gaussian uncertainty and non-convex state constraints,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4238–4244.
- [11] M. Rafieisakhaei, A. Tamjidi, and S. Chakravorty, “On-line mpc-based stochastic planning in the non-gaussian belief space with non-convex constraints,” 2015.
- [12] J. Pineau, G. Gordon, and S. Thrun, “Point-based value iteration: An anytime algorithm for POMDPs,” in *International Joint Conference on Artificial Intelligence*, 2003, pp. 1025–1032.
- [13] G. Shani, J. Pineau, and R. Kaplow, “A survey of point-based pomdp solvers,” *Autonomous Agents and Multi-Agent Systems*, vol. 27, pp. 1–51, 2013.
- [14] K. M. Seiler, H. Kurniawati, and S. P. Singh, “An online and approximate solver for pomdps with continuous action space,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2290–2297.
- [15] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, “Belief space planning simplified: Trajectory-optimized lqg (t-lqg),” in *International Workshop on Algorithmic Foundations of Robotics*, 2016 (Submitted).
- [16] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [17] D. Bertsekas, *Dynamic Programming and Optimal Control: 3rd Ed.* Athena Scientific, 2007.
- [18] J. Duchi, “Derivations for linear algebra and optimization,” *Berkeley, California*, 2007.
- [19] N. Moshtagh, “Minimum volume enclosing ellipsoid,” *Convex Optimization*, vol. 111, p. 112, 2005.
- [20] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [21] A. Agha-mohammadi, S. Chakravorty, and N. Amato, “Firm: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements,” *International Journal of Robotics Research*, no. 2, 2014.
- [22] J. van den Berg, P. Abbeel, and K. Goldberg, “LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information,” *IJRR*, vol. 30, no. 7, pp. 895–913, 2011.
- [23] S. Bubeck, “Theory of convex optimization for machine learning,” *arXiv preprint arXiv:1405.4980*, 2014.
- [24] A. Nemirovsky, “Problem complexity and method efficiency in optimization.”
- [25] zakharov, “zakharov youbot model,” 2011. [Online]. Available: [https://github.com/zakharov/youbot\\_model/wiki/KUKA-youBot-kinematics,-dynamics-and-3D-model](https://github.com/zakharov/youbot_model/wiki/KUKA-youBot-kinematics,-dynamics-and-3D-model)
- [26] Youbot-store.com, “Youbot 3d model - youbot wiki,” 2016. [Online]. Available: [http://www.youbot-store.com/wiki/index.php/YouBot\\_3D\\_Model](http://www.youbot-store.com/wiki/index.php/YouBot_3D_Model)
- [27] 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,” *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.
- [28] 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.
- [29] D. S. Watkins, *Fundamentals of matrix computations*. John Wiley & Sons, 2004, vol. 64.

## APPENDIX A

### EXPRESSING THE COST FUNCTION AS A FUNCTION OF THE BELIEF

In this appendix, we show that the cost function in (4), expressed as a function of state and control, is indeed a function of belief and the control actions.

*Estimation cost:* The optimization cost function,  $J$ , can be rewritten as follows:

$$\begin{aligned}
J &= \mathbb{E} \left[ \sum_{t=1}^K \tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t + \tilde{\mathbf{u}}_{t-1}^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1} \right] \\
&= \mathbb{E} \left[ \sum_{t=1}^K (\mathbf{x}_t - \hat{\mathbf{x}}_t)^T \mathbf{W}_t^x (\mathbf{x}_t - \hat{\mathbf{x}}_t) + \tilde{\mathbf{u}}_{t-1}^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1} \right] \\
&= \mathbb{E} \left[ \sum_{t=1}^K (\mathbf{x}_t - \hat{\mathbf{x}}_t)^T \mathbf{W}_t^T \mathbf{W}_t (\mathbf{x}_t - \hat{\mathbf{x}}_t) + \tilde{\mathbf{u}}_{t-1}^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1} \right] \\
&= \mathbb{E} \left[ \sum_{t=1}^K (\mathbf{W}_t (\mathbf{x}_t - \hat{\mathbf{x}}_t))^T (\mathbf{W}_t (\mathbf{x}_t - \hat{\mathbf{x}}_t)) + \tilde{\mathbf{u}}_{t-1}^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1} \right] \\
&= \mathbb{E} \left[ \sum_{t=1}^K \text{tr}[(\mathbf{W}_t (\mathbf{x}_t - \hat{\mathbf{x}}_t))(\mathbf{W}_t (\mathbf{x}_t - \hat{\mathbf{x}}_t))^T] + \tilde{\mathbf{u}}_{t-1}^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1} \right] \\
&= \mathbb{E} \left[ \sum_{t=1}^K \text{tr}[\mathbf{W}_t (\mathbf{x}_t - \hat{\mathbf{x}}_t)(\mathbf{x}_t - \hat{\mathbf{x}}_t)^T \mathbf{W}_t^T] + \tilde{\mathbf{u}}_{t-1}^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1} \right] \\
&= \sum_{t=1}^K (\text{tr}[\mathbf{W}_t \mathbb{E}[(\mathbf{x}_t - \hat{\mathbf{x}}_t)(\mathbf{x}_t - \hat{\mathbf{x}}_t)^T] \mathbf{W}_t^T] + \mathbb{E}[\tilde{\mathbf{u}}_{t-1}^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1}]) \\
&= \sum_{t=1}^K (\text{tr}[\mathbf{W}_t \mathbf{P}_{\mathbf{b}_t} \mathbf{W}_t^T] + \mathbb{E}[\tilde{\mathbf{u}}_{t-1}^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1}]),
\end{aligned}$$

where we have used the fact that since  $\mathbf{W}_t^x$  is symmetric and positive semidefinite, there exists a (non-unique) Cholesky decomposition of  $\mathbf{W}_t^x = \mathbf{W}_t^T \mathbf{W}_t$ , where the diagonal entries of the real upper triangular matrix  $\mathbf{W}_t$  can be zero [29]. Note, the Cholesky decomposition is unique, if and only if the  $\mathbf{W}_t^x$  is symmetric and positive definite. In such a case, the diagonal entries of  $\mathbf{W}_t$  are only positive. Moreover,  $\mathbf{P}_{\mathbf{b}_t} := \mathbb{E}[(\mathbf{x}_t - \hat{\mathbf{x}}_t)(\mathbf{x}_t - \hat{\mathbf{x}}_t)^T]$  is the covariance of the belief during execution. Similarly, the nominal part of the cost function,  $J^p$ , can also be written as a function of the nominal belief and the control actions using the definition of  $J^p = \sum_{t=0}^{K-1} c_t(\mathbf{b}_t^p, \mathbf{u}_t^p) + c_K(\mathbf{b}_K^p)$  in the section V, as follows:

$$\begin{aligned}
J^p &= \sum_{t=1}^K (\mathbb{E}[\tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t] + (\mathbf{u}_{t-1}^p)^T \mathbf{W}_t^u \mathbf{u}_{t-1}^p) \\
&= \sum_{t=1}^K (\mathbb{E}[(\mathbf{x}_t - \mathbf{x}_t^p)^T \mathbf{W}_t^x (\mathbf{x}_t - \mathbf{x}_t^p)] + (\mathbf{u}_{t-1}^p)^T \mathbf{W}_t^u \mathbf{u}_{t-1}^p) \\
&= \sum_{t=1}^K (\text{tr}[\mathbf{W}_t \mathbb{E}[(\mathbf{x}_t - \mathbf{x}_t^p)(\mathbf{x}_t - \mathbf{x}_t^p)^T] \mathbf{W}_t^T] + (\mathbf{u}_{t-1}^p)^T \mathbf{W}_t^u \mathbf{u}_{t-1}^p) \\
&= \sum_{t=1}^K (\text{tr}(\mathbf{W}_t \mathbf{P}_{\mathbf{b}_t^p}^+ \mathbf{W}_t^T) + (\mathbf{u}_{t-1}^p)^T \mathbf{W}_t^u \mathbf{u}_{t-1}^p),
\end{aligned}$$

where  $\mathbf{P}_{\mathbf{b}_t^p}^+ = \mathbb{E}[(\mathbf{x}_t - \mathbf{x}_t^p)(\mathbf{x}_t - \mathbf{x}_t^p)^T]$  is the covariance of the nominal belief along the  $p$ -traj.

## APPENDIX B

### KALMAN FILTERING MEAN UPDATE

In this appendix, we provide more details on the formula given in equation (7).

*Mean update equations of KF:* Let us define  $\mathbf{e}_{\hat{\mathbf{x}}_t} := \hat{\mathbf{x}}_t - \mathbf{x}_t^p$  and  $\mathbf{e}_{\hat{\mathbf{x}}_t^-} := \hat{\mathbf{x}}_t^- - \mathbf{x}_t^p$  define predicted and updated errors of LQG estimation, respectively, where  $\hat{\mathbf{x}}_t^-$  is the predicted estimate of the system obtained in the prediction step of the KF. In a Kalman filter for a system with equation given in (3), in the prediction step,  $\mathbf{e}_{\hat{\mathbf{x}}_{t+1}^-}$  is calculated as follows [21]:

$$\mathbf{e}_{\hat{\mathbf{x}}_{t+1}^-} = \mathbf{A}_t^p \mathbf{e}_{\hat{\mathbf{x}}_t} + \mathbf{B}_t^p (\mathbf{u}_t - \mathbf{u}_t^p). \quad (15)$$



Moreover, in the update step of the KF,  $\mathbf{e}_{\hat{\mathbf{x}}_{t+1}^+}$  is obtained as follows [21]:

$$\mathbf{e}_{\hat{\mathbf{x}}_{t+1}} = \mathbf{e}_{\hat{\mathbf{x}}_{t+1}^-} + \mathbf{K}_{t+1}^p((\mathbf{z}_t - \mathbf{z}_t^p) - \mathbf{H}_{t+1}^p \mathbf{e}_{\hat{\mathbf{x}}_{t+1}^-}), \quad (16)$$

where  $\mathbf{K}_{t+1}^p$  is the Kalman gain as defined in equation (6c). Therefore, using equation (15), (16) can be rewritten as follows:

$$\begin{aligned} \mathbf{e}_{\hat{\mathbf{x}}_{t+1}} &= \mathbf{A}_t^p \mathbf{e}_{\hat{\mathbf{x}}_t^+} + \mathbf{B}_t^p (\mathbf{u}_t - \mathbf{u}_t^p) + \mathbf{K}_{t+1}^p((\mathbf{z}_t - \mathbf{z}_t^p) - \mathbf{H}_{t+1}^p (\mathbf{A}_t^p \mathbf{e}_{\hat{\mathbf{x}}_t^+} + \mathbf{B}_t^p (\mathbf{u}_t - \mathbf{u}_t^p))) \\ &= \mathbf{A}_t^p (\hat{\mathbf{x}}_t - \mathbf{x}_t^p) + \mathbf{B}_t^p (\mathbf{u}_t - \mathbf{u}_t^p) + \mathbf{K}_{t+1}^p((\mathbf{z}_t - \mathbf{z}_t^p) - \mathbf{H}_{t+1}^p (\mathbf{A}_t^p (\hat{\mathbf{x}}_t - \mathbf{x}_t^p) + \mathbf{B}_t^p (\mathbf{u}_t - \mathbf{u}_t^p))) \\ &= \mathbf{x}_{t+1}^p - \mathbf{x}_{t+1}^p + \mathbf{A}_t^p (\hat{\mathbf{x}}_t - \mathbf{x}_t^p) + \mathbf{B}_t^p (\mathbf{u}_t - \mathbf{u}_t^p) \\ &\quad + \mathbf{K}_{t+1}^p((\mathbf{z}_t - \mathbf{z}_t^p) - \mathbf{H}_{t+1}^p (\mathbf{x}_{t+1}^p - \mathbf{x}_{t+1}^p + \mathbf{A}_t^p (\hat{\mathbf{x}}_t - \mathbf{x}_t^p) + \mathbf{B}_t^p (\mathbf{u}_t - \mathbf{u}_t^p))) \\ &= -\mathbf{x}_{t+1}^p + \mathbf{x}_{t+1}^p - \mathbf{A}_t^p \mathbf{x}_t^p - \mathbf{B}_t^p \mathbf{u}_t^p + \mathbf{A}_t^p \hat{\mathbf{x}}_t + \mathbf{B}_t^p \mathbf{u}_t \\ &\quad + \mathbf{K}_{t+1}^p (\mathbf{z}_t - \mathbf{z}_t^p - \mathbf{H}_{t+1}^p (-\mathbf{x}_{t+1}^p + \mathbf{x}_{t+1}^p - \mathbf{A}_t^p \mathbf{x}_t^p - \mathbf{B}_t^p \mathbf{u}_t^p + \mathbf{A}_t^p \hat{\mathbf{x}}_t + \mathbf{B}_t^p \mathbf{u}_t)) \\ &= -\mathbf{x}_{t+1}^p + \mathbf{f}_t^p + \mathbf{A}_t^p \hat{\mathbf{x}}_t + \mathbf{B}_t^p \mathbf{u}_t + \mathbf{K}_{t+1}^p (\mathbf{z}_t - (\mathbf{z}_t^p - \mathbf{H}_{t+1}^p \mathbf{x}_{t+1}^p) - \mathbf{H}_{t+1}^p (\mathbf{f}_t^p + \mathbf{A}_t^p \hat{\mathbf{x}}_t + \mathbf{B}_t^p \mathbf{u}_t)) \\ &= -\mathbf{x}_{t+1}^p + \mathbf{f}_t^p - \mathbf{K}_{t+1}^p \mathbf{H}_{t+1}^p \mathbf{f}_t^p + \mathbf{A}_t^p \hat{\mathbf{x}}_t + \mathbf{B}_t^p \mathbf{u}_t + \mathbf{K}_{t+1}^p (\mathbf{z}_t - \mathbf{h}_t^p - \mathbf{H}_{t+1}^p (\mathbf{A}_t^p \hat{\mathbf{x}}_t + \mathbf{B}_t^p \mathbf{u}_t)) \\ &= -\mathbf{x}_{t+1}^p + (\mathbf{I} - \mathbf{K}_{t+1}^p \mathbf{H}_{t+1}^p) \mathbf{f}_t^p - \mathbf{K}_{t+1}^p \mathbf{h}_t^p + \mathbf{A}_t^p \hat{\mathbf{x}}_t + \mathbf{B}_t^p \mathbf{u}_t + \mathbf{K}_{t+1}^p (\mathbf{z}_t - \mathbf{H}_{t+1}^p (\mathbf{A}_t^p \hat{\mathbf{x}}_t + \mathbf{B}_t^p \mathbf{u}_t)), \end{aligned}$$

where  $\mathbf{f}_t^p := f(\mathbf{x}_t^p, \mathbf{u}_t^p, \mathbf{0}) - \mathbf{A}_t^p \mathbf{x}_t^p - \mathbf{B}_t^p \mathbf{u}_t^p$  and  $\mathbf{h}_t^p := h(\mathbf{x}, \mathbf{0}) - \mathbf{H}_t^p \mathbf{x}_t^p$ . Therefore, using the definition of the  $\mathbf{e}_{\hat{\mathbf{x}}_{t+1}}$ , the mean update equation for a linearized system around the *o-traj* (which is a particularly a *p-traj*) can be calculated as follows:

$$\hat{\mathbf{x}}_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1}^o \mathbf{H}_{t+1}^o) \mathbf{f}_t^o - \mathbf{K}_{t+1}^o \mathbf{h}_{t+1}^o + \mathbf{A}_t^o \hat{\mathbf{x}}_t + \mathbf{B}_t^o \mathbf{u}_t + \mathbf{K}_{t+1}^o (\mathbf{z}_{t+1} - \mathbf{H}_{t+1}^o (\mathbf{A}_t^o \hat{\mathbf{x}}_t + \mathbf{B}_t^o \mathbf{u}_t))$$

with  $\hat{\mathbf{x}}_0 = \mathbb{E}(\mathbf{x}_0)$ .

## APPENDIX C PROOFS OF LEMMAS AND THEOREMS

In this appendix, we provide the proof of Lemmas 1, 2, 3, 4, 5, and Theorem 1.

**Proof: Lemma 1, Estimation Error Propagation**

Let us calculate  $\tilde{\mathbf{x}}_{t+1}$  for  $0 \leq t$  as follows:

$$\begin{aligned} \tilde{\mathbf{x}}_{t+1} &= \mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1} \\ &= (\mathbf{x}_{t+1}^p + \tilde{\mathbf{x}}_{t+1} - \mathbf{A}_t \hat{\mathbf{x}}_t - \mathbf{B}_t \mathbf{u}_t - \mathbf{K}_{t+1} (\mathbf{z}_{t+1} - \mathbf{H}_{t+1} (\mathbf{A}_t \hat{\mathbf{x}}_t + \mathbf{B}_t \mathbf{u}_t))) - ((\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \mathbf{f}_t^p - \mathbf{K}_{t+1} \mathbf{h}_{t+1}^p) \\ &= \mathbf{x}_{t+1}^p + \tilde{\mathbf{x}}_{t+1} - \mathbf{A}_t \hat{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t+1} + \mathbf{A}_t \tilde{\mathbf{x}}_t - \mathbf{B}_t \mathbf{u}_t^p + \mathbf{G}_t \boldsymbol{\omega}_t - ((\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \mathbf{f}_t^p - \mathbf{K}_{t+1} \mathbf{h}_{t+1}^p) \\ &\quad - \mathbf{K}_{t+1} (\mathbf{z}_{t+1}^p + \mathbf{H}_{t+1} \tilde{\mathbf{x}}_{t+1} - \mathbf{H}_{t+1} (\mathbf{A}_t \hat{\mathbf{x}}_t + \tilde{\mathbf{x}}_{t+1} - \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \mathbf{u}_t^p - \mathbf{G}_t \boldsymbol{\omega}_t) + \mathbf{M}_{t+1} \boldsymbol{\nu}_{t+1}) \\ &= \mathbf{x}_{t+1}^p - \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{A}_t \tilde{\mathbf{x}}_t - \mathbf{B}_t \mathbf{u}_t^p + \mathbf{G}_t \boldsymbol{\omega}_t - ((\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \mathbf{f}_t^p - \mathbf{K}_{t+1} \mathbf{h}_{t+1}^p) \\ &\quad - \mathbf{K}_{t+1} (\mathbf{z}_{t+1}^p - \mathbf{H}_{t+1} (\mathbf{A}_t \tilde{\mathbf{x}}_t - \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \mathbf{u}_t^p - \mathbf{G}_t \boldsymbol{\omega}_t) + \mathbf{M}_{t+1} \boldsymbol{\nu}_{t+1}) \\ &= \mathbf{A}_t (\mathbf{x}_t - \hat{\mathbf{x}}_t) + \mathbf{x}_{t+1}^p - \mathbf{A}_t \mathbf{x}_t^p - \mathbf{B}_t \mathbf{u}_t^p + \mathbf{G}_t \boldsymbol{\omega}_t - ((\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \mathbf{f}_t^p - \mathbf{K}_{t+1} \mathbf{h}_{t+1}^p) \\ &\quad - \mathbf{K}_{t+1} (\mathbf{z}_{t+1}^p + \mathbf{H}_{t+1} (\mathbf{A}_t (\mathbf{x}_t - \hat{\mathbf{x}}_t) - \mathbf{A}_t \mathbf{x}_t^p - \mathbf{B}_t \mathbf{u}_t^p + \mathbf{G}_t \boldsymbol{\omega}_t) + \mathbf{M}_{t+1} \boldsymbol{\nu}_{t+1}) \\ &= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{f}_t^p + \mathbf{G}_t \boldsymbol{\omega}_t - \mathbf{K}_{t+1} (\mathbf{z}_{t+1}^p + \mathbf{H}_{t+1} (\mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{f}_t^p - \mathbf{x}_{t+1}^p + \mathbf{G}_t \boldsymbol{\omega}_t) + \mathbf{M}_{t+1} \boldsymbol{\nu}_{t+1}) \\ &\quad - ((\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \mathbf{f}_t^p - \mathbf{K}_{t+1} \mathbf{h}_{t+1}^p) \\ &= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{f}_t^p + \mathbf{G}_t \boldsymbol{\omega}_t - \mathbf{K}_{t+1} (\mathbf{h}_{t+1}^p + \mathbf{H}_{t+1} (\mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{f}_t^p + \mathbf{G}_t \boldsymbol{\omega}_t) + \mathbf{M}_{t+1} \boldsymbol{\nu}_{t+1}) - ((\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \mathbf{f}_t^p - \mathbf{K}_{t+1} \mathbf{h}_{t+1}^p) \\ &= (\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \mathbf{A}_t \tilde{\mathbf{x}}_t + (\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \mathbf{G}_t \boldsymbol{\omega}_t - \mathbf{K}_{t+1} \mathbf{M}_{t+1} \boldsymbol{\nu}_{t+1} \\ &\quad + ((\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \mathbf{f}_t^p - \mathbf{K}_{t+1} \mathbf{h}_{t+1}^p) - ((\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \mathbf{f}_t^p - \mathbf{K}_{t+1} \mathbf{h}_{t+1}^p) \\ &= : \mathbf{U}_{t+1} \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{U}_{t+1} \mathbf{G}_t \boldsymbol{\omega}_t - \mathbf{K}_{t+1} \mathbf{M}_{t+1} \boldsymbol{\nu}_{t+1} \\ &= : \mathbf{F}_t \tilde{\mathbf{x}}_t + \mathbf{U}_{t+1} \mathbf{G}_t \boldsymbol{\omega}_t - \mathbf{K}_{t+1} \mathbf{M}_{t+1} \boldsymbol{\nu}_{t+1} \\ &= : \tilde{\mathbf{F}}_{0:t} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t \tilde{\mathbf{F}}_{s+1:t} (\mathbf{U}_{s+1} \mathbf{G}_s \boldsymbol{\omega}_s - \mathbf{K}_{s+1} \mathbf{M}_{s+1} \boldsymbol{\nu}_{s+1}) \\ &= \tilde{\mathbf{F}}_{0:t} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t \tilde{\mathbf{F}}_{s+1:t} (\mathbf{U}_{s+1} \mathbf{G}_s \boldsymbol{\omega}_s - \mathbf{K}_{s+1} \mathbf{M}_{s+1} \boldsymbol{\nu}_{s+1}). \end{aligned}$$

Note  $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}_0^p = \tilde{\mathbf{x}}_0$ . Also note the last formula is also correct for  $t = -1$ ; since,  $\tilde{\mathbf{F}}_{0:-1} = \mathbf{I}$  from the definition, and the summation is zero based on Assumption 1. ■

**Proof: Lemma 2, State Error Propagation**

The calculations of  $\tilde{\mathbf{x}}_{t+1}$  for  $t \geq 1$  are as follows:

$$\begin{aligned}
\tilde{\mathbf{x}}_{t+1} &= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \mathbf{G}_t \boldsymbol{\omega}_t \\
&= (\mathbf{A}_t - \mathbf{B}_t \mathbf{L}_t) \tilde{\mathbf{x}}_t + \mathbf{G}_t \boldsymbol{\omega}_t + \mathbf{B}_t (\tilde{\mathbf{F}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-1} (\tilde{\mathbf{F}}_{s,t}^{\omega} \boldsymbol{\omega}_s - \tilde{\mathbf{F}}_{s+1,t}^{\nu} \boldsymbol{\nu}_{s+1})) \\
&= : \mathbf{D}_t \tilde{\mathbf{x}}_t + \mathbf{G}_t \boldsymbol{\omega}_t + \mathbf{B}_t (\tilde{\mathbf{F}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-1} (\tilde{\mathbf{F}}_{s,t}^{\omega} \boldsymbol{\omega}_s - \tilde{\mathbf{F}}_{s+1,t}^{\nu} \boldsymbol{\nu}_{s+1})) \\
&= : \tilde{\mathbf{D}}_{0:t} \tilde{\mathbf{x}}_0 + \sum_{r=1}^t \tilde{\mathbf{D}}_{r+1:t} [\mathbf{G}_r \boldsymbol{\omega}_r + \mathbf{B}_r (\tilde{\mathbf{F}}_r^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{r-1} (\tilde{\mathbf{F}}_{s,r}^{\omega} \boldsymbol{\omega}_s - \tilde{\mathbf{F}}_{s+1,r}^{\nu} \boldsymbol{\nu}_{s+1}))] \\
&= \tilde{\mathbf{D}}_{0:t} \tilde{\mathbf{x}}_0 + \sum_{r=1}^t \tilde{\mathbf{D}}_{r+1:t} \mathbf{G}_r \boldsymbol{\omega}_r + \sum_{r=1}^t \tilde{\mathbf{D}}_{r+1:t} \mathbf{B}_r \tilde{\mathbf{F}}_r^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{r=1}^t \sum_{s=0}^{r-1} (\tilde{\mathbf{D}}_{r+1:t} \mathbf{B}_r \tilde{\mathbf{F}}_{s,r}^{\omega} \boldsymbol{\omega}_s - \tilde{\mathbf{D}}_{r+1:t} \mathbf{B}_r \tilde{\mathbf{F}}_{s+1,r}^{\nu} \boldsymbol{\nu}_{s+1}) \\
&= : (\tilde{\mathbf{D}}_{0:t} + \sum_{r=1}^t \tilde{\mathbf{D}}_{r+1:t} \mathbf{B}_r \tilde{\mathbf{F}}_r^{\mathbf{x}_0}) \tilde{\mathbf{x}}_0 + \sum_{r=1}^t \tilde{\mathbf{D}}_{r+1:t} \mathbf{G}_r \boldsymbol{\omega}_r + \sum_{r=1}^t \sum_{s=0}^{r-1} (\tilde{\mathbf{F}}_{s,r,t}^{\omega} \boldsymbol{\omega}_s - \tilde{\mathbf{F}}_{s+1,r,t}^{\nu} \boldsymbol{\nu}_{s+1}) \\
&= : \tilde{\mathbf{D}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=1}^t \tilde{\mathbf{D}}_{s+1:t} \mathbf{G}_s \boldsymbol{\omega}_s + \sum_{s=0}^{t-1} [(\sum_{r=s+1}^t \tilde{\mathbf{F}}_{s,r,t}^{\omega}) \boldsymbol{\omega}_s - (\sum_{r=s+1}^t \tilde{\mathbf{F}}_{s+1,r,t}^{\nu}) \boldsymbol{\nu}_{s+1}] \\
&= \tilde{\mathbf{D}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=1}^t \tilde{\mathbf{D}}_{s+1:t} \mathbf{G}_s \boldsymbol{\omega}_s + \sum_{s=0}^{t-1} (\sum_{r=s+1}^t \tilde{\mathbf{F}}_{s,r,t}^{\omega}) \boldsymbol{\omega}_s - \sum_{s=0}^{t-1} (\sum_{r=s+1}^t \tilde{\mathbf{F}}_{s+1,r,t}^{\nu}) \boldsymbol{\nu}_{s+1} \\
&= : \tilde{\mathbf{D}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t \tilde{\mathbf{D}}_{s,t}^{\omega} \boldsymbol{\omega}_s - \sum_{s=0}^{t-1} \tilde{\mathbf{D}}_{s+1,t}^{\nu} \boldsymbol{\nu}_{s+1}.
\end{aligned}$$

Note using the definition of  $\tilde{\mathbf{x}}_t$ , the initial state error is  $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}_0^p$ . Likewise, the state error  $\tilde{\mathbf{x}}_1$  is  $\tilde{\mathbf{x}}_1 = \mathbf{A}_0 \tilde{\mathbf{x}}_0 + \mathbf{G}_0 \boldsymbol{\omega}_0$ . Moreover, these errors are consistent with the formula given in the lemma using the definitions provided and Assumption 1. ■

**Proof: Lemma 3, Control Error Propagation**

Replacing estimation error in the control law: Using equation (11), we can rewrite  $\tilde{\mathbf{u}}_{t+1}$  for  $t \geq -1$  as follows:

$$\begin{aligned}
\tilde{\mathbf{u}}_{t+1} &= -\mathbf{L}_{t+1} (\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}^p) \\
&= -\mathbf{L}_{t+1} (\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1} + \mathbf{x}_{t+1} - \mathbf{x}_{t+1}^p) \\
&= -\mathbf{L}_{t+1} \tilde{\mathbf{x}}_{t+1} - \mathbf{L}_{t+1} (\hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}) \\
&= -\mathbf{L}_{t+1} \tilde{\mathbf{x}}_{t+1} + \mathbf{L}_{t+1} \tilde{\mathbf{x}}_{t+1} \\
&= -\mathbf{L}_{t+1} \tilde{\mathbf{x}}_{t+1} + \mathbf{L}_{t+1} (\tilde{\mathbf{F}}_{0:t} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t \tilde{\mathbf{F}}_{s+1:t} (\mathbf{U}_{s+1} \mathbf{G}_s \boldsymbol{\omega}_s - \mathbf{K}_{s+1} \mathbf{M}_{s+1} \boldsymbol{\nu}_{s+1})) \\
&=: -\mathbf{L}_{t+1} \tilde{\mathbf{x}}_{t+1} + \tilde{\mathbf{F}}_{t+1}^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t (\tilde{\mathbf{F}}_{s,t+1}^{\omega} \boldsymbol{\omega}_s - \tilde{\mathbf{F}}_{s+1,t+1}^{\nu} \boldsymbol{\nu}_{s+1}). \tag{17}
\end{aligned}$$

Now, let us simplify the control error propagation using equations (17) and (12). Thus, we can write  $\tilde{\mathbf{u}}_{t+1}$  for  $t \geq -1$  as follows:

$$\begin{aligned}
\tilde{\mathbf{u}}_{t+1} &= -\mathbf{L}_{t+1} \tilde{\mathbf{x}}_{t+1} + \tilde{\mathbf{F}}_{t+1}^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t (\tilde{\mathbf{F}}_{s,t+1}^{\omega} \boldsymbol{\omega}_s - \tilde{\mathbf{F}}_{s+1,t+1}^{\nu} \boldsymbol{\nu}_{s+1}) \\
&= -\mathbf{L}_{t+1} (\tilde{\mathbf{D}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t \tilde{\mathbf{D}}_{s,t}^{\omega} \boldsymbol{\omega}_s - \sum_{s=0}^{t-1} \tilde{\mathbf{D}}_{s+1,t}^{\nu} \boldsymbol{\nu}_{s+1}) + \tilde{\mathbf{F}}_{t+1}^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^t (\tilde{\mathbf{F}}_{s,t+1}^{\omega} \boldsymbol{\omega}_s - \tilde{\mathbf{F}}_{s+1,t+1}^{\nu} \boldsymbol{\nu}_{s+1}) \\
&= -(\mathbf{L}_{t+1} \tilde{\mathbf{D}}_t^{\mathbf{x}_0} - \tilde{\mathbf{F}}_{t+1}^{\mathbf{x}_0}) \tilde{\mathbf{x}}_0 - \sum_{s=0}^t (\mathbf{L}_{t+1} \tilde{\mathbf{D}}_{s,t}^{\omega} - \tilde{\mathbf{F}}_{s,t+1}^{\omega}) \boldsymbol{\omega}_s - \sum_{s=0}^{t-1} (\tilde{\mathbf{F}}_{s+1,t+1}^{\nu} - \mathbf{L}_{t+1} \tilde{\mathbf{D}}_{s+1,t}^{\nu}) \boldsymbol{\nu}_{s+1} - \tilde{\mathbf{F}}_{t+1,t+1}^{\nu} \boldsymbol{\nu}_{t+1} \\
&=: -\mathbf{L}_{t+1}^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 - \sum_{s=0}^t \mathbf{L}_{s,t+1}^{\omega} \boldsymbol{\omega}_s - \sum_{s=0}^t \mathbf{L}_{s,t+1}^{\nu} \boldsymbol{\nu}_{s+1}.
\end{aligned}$$

Note  $\tilde{\mathbf{u}}_0 = \mathbf{0}$ , and the last formula is consistent with this error using the definitions provided in the lemma and Assumption 1. ■

**Proof: Lemma 4, Observation Error Propagation**

Let us calculate  $\tilde{\mathbf{z}}_{t+1}$  for  $t \geq 0$  as follows:

$$\begin{aligned}\tilde{\mathbf{z}}_{t+1} &= \mathbf{H}_{t+1}\tilde{\mathbf{x}}_{t+1} + \mathbf{M}_{t+1}\boldsymbol{\nu}_{t+1} \\ &= \mathbf{H}_{t+1}(\tilde{\mathbf{D}}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{s=0}^t \tilde{\mathbf{D}}_{s,t}^{\omega}\boldsymbol{\omega}_s - \sum_{s=0}^{t-1} \tilde{\mathbf{D}}_{s+1,t}^{\nu}\boldsymbol{\nu}_{s+1}) + \mathbf{M}_{t+1}\boldsymbol{\nu}_{t+1} \\ &=: \tilde{\mathbf{H}}_{t+1}^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{s=0}^t \tilde{\mathbf{H}}_{s,t+1}^{\omega}\boldsymbol{\omega}_s + \sum_{s=0}^t \tilde{\mathbf{H}}_{s+1,t+1}^{\nu}\boldsymbol{\nu}_{s+1}.\end{aligned}$$

■

**Proof: Lemma 5, Belief Error Propagation**

Utilizing the linearized equation of the belief dynamics, we can obtain the propagated error of the belief, as well. First note that, since  $\tilde{\mathbf{b}}_{t+1} = \mathbf{T}_t^{\mathbf{b}}\tilde{\mathbf{b}}_t + \mathbf{T}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t + \mathbf{T}_t^{\mathbf{z}}\tilde{\mathbf{z}}_{t+1}$ , we have  $\tilde{\mathbf{b}}_{t+1} = \sum_{s=0}^t \tilde{\mathbf{T}}_{s+1:t}^{\mathbf{b}}(\mathbf{T}_s^{\mathbf{u}}\tilde{\mathbf{u}}_s + \mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{z}}_{s+1})$ . Now, we can rewrite the belief error for  $t \geq 0$  as follows:

$$\begin{aligned}\tilde{\mathbf{b}}_t &= \sum_{s=0}^{t-1} \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}(\mathbf{T}_s^{\mathbf{u}}\tilde{\mathbf{u}}_s + \mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{z}}_{s+1}) \\ &= \tilde{\mathbf{T}}_{1:t-1}^{\mathbf{b}}\mathbf{T}_0^{\mathbf{u}}\tilde{\mathbf{u}}_0 + \sum_{s=1}^{t-1} \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{u}}\tilde{\mathbf{u}}_s + \sum_{s=0}^{t-1} \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{z}}_{s+1} \\ &= \sum_{s=0}^{t-2} \tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}\tilde{\mathbf{u}}_{s+1} + \sum_{s=0}^{t-2} \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{z}}_{s+1} + \tilde{\mathbf{T}}_{t-1}^{\mathbf{b}}\mathbf{T}_{t-1}^{\mathbf{z}}\tilde{\mathbf{z}}_t \\ &= \sum_{s=0}^{t-2} \tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}(-\mathbf{L}_{s+1}^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 - \sum_{r=0}^s \mathbf{L}_{r,s+1}^{\omega}\boldsymbol{\omega}_r - \sum_{r=0}^s \mathbf{L}_{r,s+1}^{\nu}\boldsymbol{\nu}_{r+1}) \\ &\quad + \sum_{s=0}^{t-2} \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}(\tilde{\mathbf{H}}_{s+1}^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{r=0}^s \tilde{\mathbf{H}}_{r,s+1}^{\omega}\boldsymbol{\omega}_r + \sum_{r=0}^s \tilde{\mathbf{H}}_{r+1,s+1}^{\nu}\boldsymbol{\nu}_{r+1}) \\ &\quad + \mathbf{T}_{t-1}^{\mathbf{z}}(\tilde{\mathbf{H}}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-1} \tilde{\mathbf{H}}_{s,t}^{\omega}\boldsymbol{\omega}_s + \sum_{s=0}^{t-1} \tilde{\mathbf{H}}_{s+1,t}^{\nu}\boldsymbol{\nu}_{s+1}) \\ &= \sum_{s=0}^{t-2} (-\tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}\mathbf{L}_{s+1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-2} \sum_{r=0}^s (-\tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}\mathbf{L}_{r,s+1}^{\omega})\boldsymbol{\omega}_r + \sum_{s=0}^{t-2} \sum_{r=0}^s (-\tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}\mathbf{L}_{r,s+1}^{\nu})\boldsymbol{\nu}_{r+1} \\ &\quad + \sum_{s=0}^{t-2} (\tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{H}}_{s+1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-2} \sum_{r=0}^s (\tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{H}}_{r,s+1}^{\omega})\boldsymbol{\omega}_r + \sum_{s=0}^{t-2} \sum_{r=0}^s (\tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{H}}_{r+1,s+1}^{\nu})\boldsymbol{\nu}_{r+1} \\ &\quad + \mathbf{T}_{t-1}^{\mathbf{z}}\tilde{\mathbf{H}}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-1} \mathbf{T}_{t-1}^{\mathbf{z}}\tilde{\mathbf{H}}_{s,t}^{\omega}\boldsymbol{\omega}_s + \sum_{s=0}^{t-1} \mathbf{T}_{t-1}^{\mathbf{z}}\tilde{\mathbf{H}}_{s+1,t}^{\nu}\boldsymbol{\nu}_{s+1} \\ &= \sum_{s=0}^{t-2} (-\tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}\mathbf{L}_{s+1}^{\mathbf{x}_0} + \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{H}}_{s+1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0 + \mathbf{T}_{t-1}^{\mathbf{z}}\tilde{\mathbf{H}}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 \\ &\quad + \sum_{s=0}^{t-2} \sum_{r=0}^s (-\tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}\mathbf{L}_{r,s+1}^{\omega} + \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{H}}_{r,s+1}^{\omega})\boldsymbol{\omega}_r + \sum_{s=0}^{t-1} \mathbf{T}_{t-1}^{\mathbf{z}}\tilde{\mathbf{H}}_{s,t}^{\omega}\boldsymbol{\omega}_s \\ &\quad + \sum_{s=0}^{t-2} \sum_{r=0}^s (-\tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}\mathbf{L}_{r,s+1}^{\nu} + \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{H}}_{r+1,s+1}^{\nu})\boldsymbol{\nu}_{r+1} + \sum_{s=0}^{t-1} \mathbf{T}_{t-1}^{\mathbf{z}}\tilde{\mathbf{H}}_{s+1,t}^{\nu}\boldsymbol{\nu}_{s+1} \\ &= [\mathbf{T}_{t-1}^{\mathbf{z}}\tilde{\mathbf{H}}_t^{\mathbf{x}_0} + \sum_{s=0}^{t-2} (-\tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}\mathbf{L}_{s+1}^{\mathbf{x}_0} + \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{H}}_{s+1}^{\mathbf{x}_0})]\tilde{\mathbf{x}}_0 \\ &\quad + \sum_{r=0}^{t-2} \sum_{s=r}^{t-2} (-\tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}\mathbf{L}_{r,s+1}^{\omega} + \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{H}}_{r,s+1}^{\omega})\boldsymbol{\omega}_r + \sum_{s=0}^{t-1} \mathbf{T}_{t-1}^{\mathbf{z}}\tilde{\mathbf{H}}_{s,t}^{\omega}\boldsymbol{\omega}_s \\ &\quad + \sum_{r=0}^{t-2} \sum_{s=r}^{t-2} (-\tilde{\mathbf{T}}_{s+2:t-1}^{\mathbf{b}}\mathbf{T}_{s+1}^{\mathbf{u}}\mathbf{L}_{r,s+1}^{\nu} + \tilde{\mathbf{T}}_{s+1:t-1}^{\mathbf{b}}\mathbf{T}_s^{\mathbf{z}}\tilde{\mathbf{H}}_{r+1,s+1}^{\nu})\boldsymbol{\nu}_{r+1} + \sum_{s=0}^{t-1} \mathbf{T}_{t-1}^{\mathbf{z}}\tilde{\mathbf{H}}_{s+1,t}^{\nu}\boldsymbol{\nu}_{s+1}\end{aligned}$$

$$\begin{aligned}
& = : \tilde{\mathbf{T}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-2} \sum_{r=s}^{t-2} (-\tilde{\mathbf{T}}_{r+2:t-1}^{\mathbf{b}} \mathbf{T}_{r+1}^{\mathbf{u}} \mathbf{L}_{s,r+1}^{\omega} + \tilde{\mathbf{T}}_{r+1:t-1}^{\mathbf{b}} \mathbf{T}_r^{\mathbf{z}} \tilde{\mathbf{H}}_{s,r+1}^{\omega}) \omega_s + \sum_{s=0}^{t-2} \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{s,t}^{\omega} \omega_s + \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{t-1,t}^{\omega} \omega_{t-1} \\
& \quad + \sum_{s=0}^{t-2} \sum_{r=s}^{t-2} (-\tilde{\mathbf{T}}_{r+2:t-1}^{\mathbf{b}} \mathbf{T}_{r+1}^{\mathbf{u}} \mathbf{L}_{s,r+1}^{\nu} + \tilde{\mathbf{T}}_{r+1:t-1}^{\mathbf{b}} \mathbf{T}_r^{\mathbf{z}} \tilde{\mathbf{H}}_{s+1,r+1}^{\nu}) \nu_{s+1} + \sum_{s=0}^{t-2} \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{s+1,t}^{\nu} \nu_{s+1} + \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{t,t}^{\nu} \nu_t \\
& = \tilde{\mathbf{T}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-2} [\mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{s,t}^{\omega} + \sum_{r=s}^{t-2} (-\tilde{\mathbf{T}}_{r+2:t-1}^{\mathbf{b}} \mathbf{T}_{r+1}^{\mathbf{u}} \mathbf{L}_{s,r+1}^{\omega} + \tilde{\mathbf{T}}_{r+1:t-1}^{\mathbf{b}} \mathbf{T}_r^{\mathbf{z}} \tilde{\mathbf{H}}_{s,r+1}^{\omega})] \omega_s + \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{t-1,t}^{\omega} \omega_{t-1} \\
& \quad + \sum_{s=0}^{t-2} [\mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{s+1,t}^{\nu} + \sum_{r=s}^{t-2} (-\tilde{\mathbf{T}}_{r+2:t-1}^{\mathbf{b}} \mathbf{T}_{r+1}^{\mathbf{u}} \mathbf{L}_{s,r+1}^{\nu} + \tilde{\mathbf{T}}_{r+1:t-1}^{\mathbf{b}} \mathbf{T}_r^{\mathbf{z}} \tilde{\mathbf{H}}_{s+1,r+1}^{\nu})] \nu_{s+1} + \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{t,t}^{\nu} \nu_t \\
& = : \tilde{\mathbf{T}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-1} \tilde{\mathbf{T}}_{s,t}^{\omega} \omega_s + \sum_{s=1}^{t-1} [\mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{s,t}^{\nu} + \sum_{r=s-1}^{t-2} (-\tilde{\mathbf{T}}_{r+2:t-1}^{\mathbf{b}} \mathbf{T}_{r+1}^{\mathbf{u}} \mathbf{L}_{s-1,r+1}^{\nu} + \tilde{\mathbf{T}}_{r+1:t-1}^{\mathbf{b}} \mathbf{T}_r^{\mathbf{z}} \tilde{\mathbf{H}}_{s,r+1}^{\nu})] \nu_s + \mathbf{T}_{t-1}^{\mathbf{z}} \tilde{\mathbf{H}}_{t,t}^{\nu} \nu_t \\
& = : \tilde{\mathbf{T}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-1} \tilde{\mathbf{T}}_{s,t}^{\omega} \omega_s + \sum_{s=1}^t \tilde{\mathbf{T}}_{s,t}^{\nu} \nu_s.
\end{aligned}$$

Note  $\tilde{\mathbf{b}}_0 = 0$ , and this is consistent with the last formula, using the definitions given in the lemma and Assumption 1.  $\blacksquare$

*Proof: Theorem 1, Cost Function Error*

Using the linearization process described previously, we can write the cost function error as  $\tilde{J} = \mathbb{E}[\sum_{t=0}^{K-1} (\mathbf{C}_t^{\mathbf{b}} \tilde{\mathbf{b}}_t + \mathbf{C}_t^{\mathbf{u}} \tilde{\mathbf{u}}_t) + \mathbf{C}_K^{\mathbf{b}} \tilde{\mathbf{b}}_K]$ . Utilizing the assumption that the process and observation noises are zero mean i.i.d.,  $\mathbb{E}[\omega_t] = \mathbb{E}[\nu_t] = 0$  for all  $t$ . Moreover,  $\mathbb{E}[\tilde{\mathbf{x}}_0] = 0$  which follows from the fact that  $\mathbb{E}[\mathbf{b}_0] = \mathbf{x}_0$ . Therefore, using the linearity of expectation operator and using Lemmas 3 and 5, we can rewrite  $\tilde{J}$  as follows:

$$\begin{aligned}
\tilde{J} & = \sum_{t=0}^{K-1} (\mathbf{C}_t^{\mathbf{b}} \mathbb{E}[\tilde{\mathbf{b}}_t] + \mathbf{C}_t^{\mathbf{u}} \mathbb{E}[\tilde{\mathbf{u}}_t]) + \mathbf{C}_K^{\mathbf{b}} \mathbb{E}[\tilde{\mathbf{b}}_K] \\
& = \sum_{t=0}^K \mathbf{C}_t^{\mathbf{b}} \mathbb{E}[\tilde{\mathbf{T}}_t^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-1} \tilde{\mathbf{T}}_{s,t}^{\omega} \omega_s + \sum_{s=1}^t \tilde{\mathbf{T}}_{s,t}^{\nu} \nu_s] + \sum_{t=0}^{K-1} \mathbf{C}_t^{\mathbf{u}} \mathbb{E}[-\mathbf{L}_{t+1}^{\mathbf{x}_0} \tilde{\mathbf{x}}_0 - \sum_{s=0}^t \mathbf{L}_{s,t+1}^{\omega} \omega_s - \sum_{s=0}^t \mathbf{L}_{s,t+1}^{\nu} \nu_{s+1}] \\
& = \sum_{t=0}^K \mathbf{C}_t^{\mathbf{b}} (\tilde{\mathbf{T}}_t^{\mathbf{x}_0} \mathbb{E}[\tilde{\mathbf{x}}_0] + \sum_{s=0}^{t-1} \tilde{\mathbf{T}}_{s,t}^{\omega} \mathbb{E}[\omega_s] + \sum_{s=1}^t \tilde{\mathbf{T}}_{s,t}^{\nu} \mathbb{E}[\nu_s]) + \sum_{t=0}^{K-1} \mathbf{C}_t^{\mathbf{u}} (-\mathbf{L}_{t+1}^{\mathbf{x}_0} \mathbb{E}[\tilde{\mathbf{x}}_0] - \sum_{s=0}^t \mathbf{L}_{s,t+1}^{\omega} \mathbb{E}[\omega_s] - \sum_{s=0}^t \mathbf{L}_{s,t+1}^{\nu} \mathbb{E}[\nu_{s+1}]) \\
& = \sum_{t=0}^K \mathbf{C}_t^{\mathbf{b}} (\tilde{\mathbf{T}}_t^{\mathbf{x}_0} \mathbf{0} + \sum_{s=0}^{t-1} \tilde{\mathbf{T}}_{s,t}^{\omega} \mathbf{0} + \sum_{s=1}^t \tilde{\mathbf{T}}_{s,t}^{\nu} \mathbf{0}) + \sum_{t=0}^{K-1} \mathbf{C}_t^{\mathbf{u}} (-\mathbf{L}_{t+1}^{\mathbf{x}_0} \mathbf{0} - \sum_{s=0}^t \mathbf{L}_{s,t+1}^{\omega} \mathbf{0} - \sum_{s=0}^t \mathbf{L}_{s,t+1}^{\nu} \mathbf{0}) \\
& = 0.
\end{aligned}$$

$\blacksquare$