# A DECOUPLING PRINCIPLE FOR SIMULTANEOUS LOCALIZATION AND PLANNING UNDER UNCERTAINTY IN MULTI-AGENT DYNAMIC ENVIRONMENTS

A Dissertation

by

MOHAMMADHUSSEIN RAFIEISAKHAEI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

| | |
|---|---|
| Chair of Committee, | P. R. Kumar |
| Co-Chair of Committee, | Suman Chakravorty |
| Committee Members, | Srinivas Shakkottai |
| | Aniruddha Datta |
| Head of Department, | Miroslav M. Begovic |

December 2017

Major Subject: Electrical Engineering

# ABSTRACT

Simultaneous localization and planning for nonlinear stochastic systems under process and measurement uncertainties is a challenging problem. In its most general form, it is formulated as a stochastic optimal control problem in the space of feedback policies. The Hamilton-Jacobi-Bellman equation provides the theoretical solution of the optimal problem; but, as is typical of almost all nonlinear stochastic systems, optimally solving the problem is intractable. Moreover, even if an optimal solution was obtained, it would require centralized control, while multi-agent mobile robotic systems under dynamic environments require decentralized solutions.

In this study, we aim for a theoretically sound solution for various modes of this problem, including the single-agent and multi-agent variations with perfect and imperfect state information, where the underlying state, control and observation spaces are continuous with discrete-time models. We introduce a decoupling principle for planning and control of multi-agent nonlinear stochastic systems based on a small noise asymptotics. Through this decoupling principle, under small noise, the design of the real-time feedback law can be decoupled from the off-line design of the nominal trajectory of the system. Further, for a multi-agent problem, the design of the feedback laws for different agents can be decoupled from each other, reducing the centralized problem to a decentralized problem requiring no communication during execution. The resulting solution is quantifiably near-optimal.

We establish this result for all the above-mentioned variations, which results in the following variants: Trajectory-optimized Linear Quadratic Regulator (T-LQR), Multi-agent T-LQR (MT-LQR), Trajectory-optimized Linear Quadratic Gaussian (T-LQG), and Multi-agent T-LQG (MT-LQG). The decoupling principle provides the conditions under which a decentralized linear Gaussian system with a quadratic approximation of the cost, obtained by linearization around an optimally designed nominal trajectory can be utilized to control the nonlinear system. The resulting decentralized feedback solution at runtime, being decoupled with respect to the mobile agents, requires no communication between the agents during the execution phase. Moreover, the complexity of the solution vis-a-vis the computation of the nominal trajectory as well as the closed-loop gains is tractable with low polynomial orders of computation. Experimental implementation of the solution shows that the results hold for moderate levels of noise with high probability.

Further optimizing the performance of this approach we show how to design a special cost function for the problem with imperfect state measurement that takes advantage of the fact that the estimation covariance of a linear Gaussian system is deterministic and not dependent on the observations. This design, which corresponds in our overall design to “belief space planning”, incorporates the consequently deterministic cost of the stochastic feedback system into the deterministic design of the nominal trajectory to obtain an optimal nominal trajectory with the best estimation performance. Then, it utilizes the T-LQG approach to design an optimal feedback law to track the designed nominal trajectory. This iterative approach can be used to further tune both the open loop as well as the decentralized feedback gain portions of the overall design. We also provide the multi-agent variant of this approach based on the MT-LQG method.

Based on the near-optimality guarantees of the decoupling principle and the T-LQG approach, we analyze the performance and correctness of a well-known heuristic in robotic path planning. We show that optimizing measures of the observability Gramian as a surrogate for estimation performance may provide irrelevant or misleading trajectories for planning under observation uncertainty.

We then consider systems with non-Gaussian perturbations. An alternative heuristic method is proposed that aims for fast planning in belief space under non-Gaussian uncertainty. We provide a special design approach based on particle filters that results in a convex planning problem implemented via a model predictive control strategy in convex environments, and a locally convex problem in non-convex environments. The environment here refers to the complement of the region in Euclidean space that contains the obstacles or “no fly zones”.

For non-convex dynamic environments, where the no-go regions change dynamically with time, we design a special form of an obstacle penalty function that incorporates non-convex time-varying constraints into the cost function, so that the decoupling principle still applies to these problems. However, similar to any constrained problem, the quality of the optimal nominal trajectory is dependent on the quality of the solution obtainable for the nonlinear optimization problem.

We simulate our algorithms for each of the problems on various challenging situations, including for several nonlinear robotic models and common measurement models. In particular, we consider 2D and 3D dynamic environments for heterogeneous holonomic and non-holonomic robots, and range and bearing sensing models. Future research can potentially extend the results to more general situations including continuous-time models.

# DEDICATION

*To my mother*

*whose passionate effort is the reason behind lots of my achievements.*

# ACKNOWLEDGEMENTS

I would like to express my great respect and gratitude to my advisor Prof. P. R. Kumar, and my co-advisor Prof. Suman Chakravorty. Over the course of my PhD, Dr. Kumar has taught me the importance of mathematical rigor in theory and Dr. Chakravorty has taught me the importance of intuition behind the practical aspects of the problems. Both were very supportive and provided resources whenever I needed them. In particular, they created an environment that was very open for new ideas and trained me for independent research. I also enjoyed long constructive brainstorming discussions along with mentorship relation with Suman and exact problem analysis meetings with Dr. Kumar. I would like to thank my committee members Dr. Srinivas Shakkottai and Dr. Aniruddha Datta for their helpful comments and advice. I also want to express my gratitude to Dr. Behbood B. Zoghi who helped me a lot in successful transitioning in the beginning of my studies.

During my studies, I have enjoyed the support of my friends who helped me along my stay in College Station. I express my sincere thanks for Babak Barazandeh for being a great support and a helpful friend, Arash Torkan for being always there to reach out when I needed and ask for help, and Maricruz Salinas for being an invaluable friend, help and support during the work that I did for my research. I was privileged to work with Dr. Chakravorty's group in Estimation, Decision and Planning Laboratory (EDPLab) and Dr. Kumar's group in Cyber-Physical Systems Lab. I would like to thank all my friends and colleagues Amir Hossein Tamjidi, Saurav Agarwal, Dan Yu, Abhishek Halder, Simon Yao, Jonnathon Ponnniah, Woo-Hyun Ko, Bharadwaj Satchidanandan, and my office-mate Ping-Chun Hsieh with whom I had a lot of constructive discussions. I also thank Mostafa Ghoreyshi and

Reza Manshouri for their great friendship. I would extend my gratitude for Ali-Akbar Agha-Mohammadi who introduced this field of study to me and was always supportive during my study.

Last but not least, I cannot thank enough from my Mom and Dad and my Sisters for their passionate support. They are an invaluable source of love for me, whom I can always trust and reach out.

# CONTRIBUTORS AND FUNDING SOURCES

## Contributors

Amir Hossein Tamjidi has contributed in the simulations and work of Chapter 11 that were also illustrated in [1].

## Funding Sources

Graduate study was supported by Graduate Assistantships and ISFAA grants from Texas A&M University. The material on this dissertation is based upon work partially supported by the NSF under Contract Nos. CNS-1302182, CNS-1646449, and Science & Technology Center Grant CCF-0939370, the AFOSR Contract No. FA9550-13-1-0008, the U.S. Army Research Office under Contract No. W911NF-15-1-0279, and NPRP grants NPRP 6-784-2-329 and NPRP 8-1531-2-651 from the Qatar National Research Fund, a member of Qatar Foundation.

# NOMENCLATURE

| | |
|---|---|
| SLAP | Simultaneous Localization and Planning |
| EKF | Extended Kalman Filter |
| OG | Observability Gramian |
| T-LQG | Trajectory-optimized Linear Quadratic Gaussian |
| T-LQR | Trajectory-optimized Linear Quadratic Regulator |
| MT-LQG | Multi-agent Trajectory-optimized Linear Quadratic Gaussian |
| MT-LQR | Multi-agent Trajectory-optimized Linear Quadratic Regulator |
| MPC | Model Predictive Control |
| RHC | Receding Horizon Control |
| MDP | Markov Decision Process |
| POMDP | Partially Observed Markov Decision Process |
| Dec-POMDP | Decentralized Partially Observed Markov Decision Process |

# TABLE OF CONTENTS

Page

# LIST OF FIGURES

FIGURE Page

# LIST OF TABLES

TABLE Page

# I. THE DECOUPLING PRINCIPLE

# 1. INTRODUCTION AND LITERATURE REVIEW

In this chapter, we introduce the problem that is discussed in this research, provide a literature review of previous related work, lay out a brief overview of the organization of the dissertation, and discuss the contributions of this work as well as possible future developments.

## 1.1 Introduction

Planning under uncertainty is a challenging problem for stochastic systems. Many problems in robotics fall in this category since there is inherent uncertainty in the measurements obtained from sensors in addition to the uncertainty in the robot's motion. The uncertainty can result from several causes such as unpredicted forces, e.g., wind forces that occur in aerial vehicles or sudden unexpected interactions of a robot with its environment. It may also arise in medical robotics while steering a needle in a tiny environment surrounded with soft tissues of a live organ. The actions prescribed by the controller of a ground robot might not be performed well due to unmodeled friction. Many robotic systems are equipped with noisy actuators that require feedback compensation or planning ahead and a policy that accounts for the random perturbations even in perfect environments. Simply ignoring the noise and planning for the unperturbed equivalent of the stochastic system can result in crucial errors, leading to failure in reaching the end-goal, or cause the system to fall into unsafe states.

The main challenge in this category of problems is that the controller's knowledge about the true state of the system is limited to the conditional probability distribution of the state given the past data history of actions and observations, which is an information state that is a sufficient statistic for the problem [2]. We will just refer to it as the “information state” in the general case, and as the “belief” for the specific case of a linear Gaussian system where the conditional distribution is Gaussian. The controller needs to plan over the space of all possible probability distributions over the state, referred to as the information (belief) space, which is infinite-dimensional in practical problems where the underlying state space is a finite-dimensional vector space. In the special case of a linear Gaussian system where the belief can be represented by just a vector of mean and covariance, the belief space is a finite-dimensional vector space. The most general case of the problem can be formulated as a stochastic optimal control problems in the space of policies, or equivalently is framed as a Partially Observed Markov Decision Process (POMDP) [3], [4], [5], whose solution involves iteratively solving a set of Dynamic Programming (DP) equations over the information (belief) space. This is generally difficult to solve. The major concerns vis-a-vis solutions of the multi-agent problem are tractability of the solution, as well as the amount of communication between the agents required during the execution of the policies.

In this work, we address the nonlinear stochastic control problem for a multi-agent system and propose an architecture under which, first, the centralized design of feedback policies for different agents can be decoupled near-optimally to a decentralized solution; and second, the design of an optimal open-loop control sequence and a feedback policy to track that trajectory can be near-optimally decoupled. We term this overall result as a decoupling principle, and state and prove it rigorously for single-agent and multi-agent systems with perfect or imperfect information. In particular, we show that under a small noise assumption, the decoupling of the nominal trajectory design and a feedback control law to track the nominal trajectory holds for a nonlinear stochastic system. For the multi-agent situation, this leads to a near-optimal decoupled design of the feedback policies for different agents.

We quantify the first-order stochastic error for small-noise levels based on large-deviations theory [6], and show that the expected first-order deviation of the cost function is zero. That is, the first-order approximation of the expected stochastic cost function is dominated by the nominal cost, independent of the linear feedback gain. We thereby arrive at Trajectory-optimized Linear Quadratic Regulator (T-LQR) or Trajectory-optimized LQG (T-LQG) designs for a single-agent fully-observed or partially-observed nonlinear stochastic systems, respectively, under Gaussian small-noise perturbations. Then, we extend the results to a multi-agent setting and obtain the Multi-agent T-LQR and Multi-agent T-LQG designs.

In short, for a single-agent problem, the design can be broken into two parts: *i)* an open-loop optimal trajectory planning problem that designs the nominal trajectory of the LQR/LQG controller, which respects the nonlinearities; *ii)* the design of an LQR/LQG policy to track the optimized nominal trajectory. For the multi-agent setting, we assume that the dynamics of different agents are independent and they are only coupled with respect to a cost function that needs to be optimized. This leads to the near-optimal policy design, which involves first solving the joint nominal trajectory optimization problem followed by the design of feedback laws (and estimators) for each agent independently from the other agents.

The quadratic cost of the LQG design can be chosen such that it results in a decoupled feedback law where the agents do not need to estimate or employ each other's states. This sheds light into the circumstances under which a centralized multi-agent stochastic optimal control problem can be reduced near-optimally into a factored decentralized problem and then near-optimally solved. Importantly, all these methods require only a polynomial order of computations. Therefore, this reduces both the communication requirements as well as the computational burden of those classes of multi-agent nonlinear stochastic problems, while still resulting in a near-optimal solution.

To substantiate the decoupling principle, we determine conditions under which the general nonlinear stochastic system with additive Gaussian perturbations can be controlled via a surrogate linear Gaussian system around a nominal solution. This system is constructed via linearizing the nonlinear models around the optimized nominal trajectory. Due to the nonlinearity, the original system's distributions remain non-Gaussian, whereas the linear surrogate system's conditional distribution is Gaussian. We analyze the validity of these approximations and characterize the probabilistic bounds precisely. Then, we utilize the well-defined characteristics of the belief evolution of the linear surrogate system to define a specific form of cost function in terms of the belief to obtain nominal trajectories that aim for better estimation performance as well as resulting in a decoupling of the control law. We refer to this form of the problem as the belief space planning. We utilize the T-LQG and the MT-LQG framework to obtain the decoupled problems for belief space planning as well.

Next, based on the theoretical guarantees of the T-LQG method, we analyze the usage of the Observability Gramian (OG) in robotic path planning problems. We analyze the limitations and the practical usage of designs based on the OG.

Last, we consider systems with non-Gaussian additive uncertainty and design a heuristic trajectory design approach for tackling problems under non-Gaussian uncertainty using particle filters. The optimization problem that is solved in this approach is a convex program for common nonlinear observation models in convex environments. Moreover, the resulting approach is implemented via a model predictive control that provides feedback.

Finally, we present simulation results and analyze the performance aspects of our method, such as the dependence of the performance on the tuning parameters for various models and environments.

## 1.2 Literature Review

In this section, we review the related literature in the category of the problems relevant to the current work. After providing a general background, we review the related methods in multi-agent literature, small noise theory, Point-based POMDP solvers, LQG-based methods and MPC-based methods.

### *1.2.1 General Background*

In a stochastic environment, the general problem of sequential decision-making is formulated as a Markov Decision Process (MDP) [2, 7]. The optimal solution of the stochastic control problem can be obtained iteratively by value or policy iteration methods to solve the Hamilton-Jacobi-Bellman equation [7]. Except in special cases, such as in a linear Gaussian environment, this involves discretization of the underlying spaces [8]; an approach whose scalability faces the curse of dimensionality [9]. As a result, they require a computation time that is provably exponential in the state dimension, in a real number based model of complexity, without any assumption that $P \neq NP$ [10].

In a situation with imperfect state information where the sensing data is contaminated with noise, the problem can be formulated as a Partially Observed MDP (POMDP) [11]. In this setting the notion of "information state" or "belief state" of the system, which encompasses the entire data history of the problem as a conditional distribution of the state given the past observation, controls and the prior distribution, is a sufficient statistic for analysis [2, 12, 13, 14]. The stochastic optimal problem to be solved in this setting can be formulated as a search for a policy in the high-dimensional information (belief) space [2, 13, 14, 15]. Attempts to optimally solve this problem through Dynamic Programming (DP) [7] face the curse of history, i.e., the exponential growth of number of possible policies with time-horizon [16].

Many approaches have been proposed based on their tractability. Point-based POMDP solvers, which are the forward search-based variants of solving the HJB equation, have had successes during recent years in scaling to larger problems [16, 17]. However, these methods suffer both curses of dimensionality and horizon due to the exponential growth of the number of policies [16, 18, 19, 20, 21, 22, 17, 23, 24, 25, 26]. Model Predictive Control (MPC)-based methods [27, 28], robust formulations [29, 30], and other designs that relate to the Pontryagin's Maximum Principle [31], are some of the methods that have been successfully used as surrogate design approaches.

Another popular approach is utilizing Differential Dynamic Programing (DDP) [32] and DDP-based variations, such as the Stochastic DDP [33], iLQR and iLQG [34] and iLQG-based methods [35, 36]. These methods rely on local second-order linearizations of the cost function and second order (in DDP) or first-order (iLQG) approximation of the dynamics and propose iterative methods based on policy and value iterations. Heuristically, they attempt to find "locally-optimal" solutions in a tube (uncharacterized in properties) around a nominal trajectory [34]. These methods couple the design of the nominal trajectory and the feedback policy via iterative incremental local updates of the policy and run into relatively high-dimensional optimization problems with high order of complexity. Last, similar other methods such as [37] also have attempted to provide local linear quadratic approximations of the stochastic optimal problem by providing iterative methods.

### *1.2.2 Decentralized POMDPs*

In a multi-agent setting, optimally solving the problem can be formulated as a stochastic optimal control problem in the space of joint policies. Many variations of this problem have been characterized and successfully tackled based on the level of observability, in/dependence of the dynamics, cost functions and communications [38, 39, 40]. This has resulted in a variety of solutions from fully-centralized [41] to fully-decentralized approaches with many different subclasses [42, 43]. Most of the body of literature in the multi-agent belief space planning problem utilizes the general framework of the Decentralized-POMDPs (Dec-POMDPs) [44, 42, 45, 46]. While the single-agent finite-horizon POMDP problem is proven to be PSPACE-complete [47, 48, 49], the Dec-POMDP problem is in the NEXP class [46].

The major concerns of the multi-agent problem are tractability of the solution and the magnitude and frequency of communication required during the execution of the policies. While the broader knowledge assumed by the planner in a centralized approach as in Multi-agent MDP (MMDP) and Multi-agent POMDP (MPOMDP) can increase the computational tractability [50, 51, 52], it can also load the planner with a high-dimensional problem, meanwhile assuming full connectivity with a central authority during the execution. Factored Dec-POMDPs on the other hand assume some structure of independence either in observations, actions or rewards, and require less communication burdens [53, 54, 55]. A fully-decentralized approach ideally provides the lowest communication burden; however, determining the optimal policy is a more daunting task. Recent success in extending Dec-POMDP solutions have provided important computational improvements. While naive extension of POMDPs to multi-agent problems can provide poor performance, methods such as [56] plan macro-actions centrally for agents, and implement local plans for each agent in a distributed manner.

In a single-agent POMDP setting, [57, 58, 15, 59] utilize Linear Quadratic Gaussian (LQG)-based policies. [57] and [58] utilize the Most-Likely Observation (MLO) heuristic to predict the estimation covariance of the Extended Kalman Filter (EKF), and [15] takes into account all possible observations. Naive extension of the single-agent methods such as [35, 36, 60] to a multi-agent setting with $m$ agents and a state dimension of $n$, increases the dimension of the belief space from $((n)^2 + n)$ to $((mn)^2 + mn)$ leading to un-scalable complexity of the optimization problems. Extension of MPC-based methods such as [61, 62], which utilize Monte-Carlo representation of beliefs, into a multi-agent setting, require planning at every time step, and more importantly, require coordination, connectivity and communication between the agents at all time steps. Furthermore, in the Monte-Carlo-based methods, an accurate belief-approximation requires an exponential growth in the number of representative samples with $n$.

### *1.2.3 Small Noise Theory*

Much of the work conducted in the small noise control of stochastic systems has been devoted to the fully-observed single-agent problem. Earlier works, such as [63], have considered asymptotic expansions of the control correction term in the presence of small perturbations. [64], considers a special case of nonlinear systems with perfect information where the process model is linear in the control variable, i.e., $\mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) = f_1(\mathbf{x}_t) + f_2(\mathbf{x}_t)\mathbf{u}_t$, and the process model is perturbed by additive noise with $\epsilon$-variance. In this work, three results are proven. The first result concerns the $O(\epsilon)$-optimality of the optimal deterministic law under convexity of $J$ in the control (i.e., $\mathbf{v}^T(\nabla_{\mathbf{uu}}J)\mathbf{v} \succeq 0 \ , \forall \mathbf{v}$), and additional smoothness and regularity conditions. The second result concerns the $O(\epsilon^2)$-optimality of the optimal deterministic law under a stronger convexity condition of $J$ in the control (i.e., $\mathbf{v}^T(\nabla_{\mathbf{uu}}J)\mathbf{v} \succeq c(\|\mathbf{u}\|)\|\mathbf{v}\|^2 \ , \forall \mathbf{v}$, and $c(\cdot) : \mathbb{R} \to \mathbb{R}$ is a monotonically non-increasing positive function), and some smoothness and regularity conditions. The third result concerns the $O(\epsilon)$-optimality of the optimal deterministic sequence under the latter condition. Our result, on the other hand, provide the $O(\epsilon)$-optimality of the proposed design approach for a broader class of processes $\mathbf{f}(\mathbf{x}_t, \mathbf{u}_t)$ with nonlinear dependence in the control variable and more general cost functions. Most importantly, they do not assume the linear dependence on the control sequence. In fact, our simulations in [65] are performed for a car-like robot with nonlinear dependence on the control variables. We also prove the first-order optimality of the globally optimal deterministic policy to be found by DP for the same cost and dynamics that we have considered. Furthermore, while the above mentioned results are for single-agent fully-observed systems, our results hold for multi-agent partially-observed systems, as well.

Later works, such as [66] for linear quadratic problems, or [67] for open-loop control, have successfully utilized the results of [64] and provided a deeper insight into the fully-observed problem. [68] has considered the small noise control of discrete-time systems and [69] has considered discrete-time Wentzell-Freidlin theory. [70] has considered the asymptotic small noise expansions of the HJB equation for certain classes of problems. [71] has also utilized the HJB equation for asymptotic small noise results. Last, [72] provides results similar in nature to the result of [64] via a different approach.

In the context of partially-observed problems, much of the literature has been devoted to the effort of separating the control policy design problem from the estimation problem [73, 74, 75], and its various generalizations or special cases to more broad classes of problems, e.g., [76, 77, 78, 76, 79, 80, 81]. Other separated stochastic control problems have also been introduced as in [82], based on defining a measure-valued process for the unnormalized conditional distribution of state given the past observation and controls. [83] has discussed similar partially-observed diffusion processes. Regarding the small noise perturbation of the partially-observed systems, [84, 85, 86, 87] have discussed the small noise filtering problem.

While Pontryagin's Maximum Principle provides the necessary conditions for open-loop control optimality in deterministic control [31], the Stochastic Maximum Principle (SMP) provides necessary condition in order to obtain the extremal controls [88, 89, 90]. It has been also proven that designs based on the SMP for some special cases are optimal, as well [91]. More importantly, the SMP proves that the optimal control in a stochastic setting is necessarily a feedback law [91] and examples have been provided for fully-observed systems [92]. The extensions of the SMP to partially-observed problems have also been provided in [93], which shows that the solution in this case is necessarily a time-varying feedback function of the observation process. Last, [94] has also discussed necessary conditions for partially-observed problems.

### *1.2.4 Point-Based POMDP Solvers*

Major point-based POMDP solvers [95] like PBVI[96], HSVI [97], Perseus [98], SARSOP [99], consider finite state, observation and action spaces (that results from discretizing the underlying spaces) and develop a decision tree that can exactly solve the POMDP problem for the initial belief state [100, 101, 102]. Recent point-based solvers such as MCVI [103, 104] can allow continuous state spaces. However, they still handle the belief space though a global discrete representation of the value function. These algorithms consequently suffer from the curse of dimensionality [5], [105]. Generally, in point-based solvers, the time complexity of the algorithms grows exponentially with the number of (sampled) states and time horizon [49, 106]. They also suffer from the curse of history [96] due to the exponential growth of decision choices because of dependency of future decisions on previous ones. Further, they guarantee optimality of their solution only for the particular initial belief state. This means that if there is a deviation from the planned trajectory during the execution (which happens with probability one), it becomes impractical to re-plan and compensate for the accumulated errors due to te computational cost. Therefore, these methods are not suitable for use in on-line planning where the planner should constantly compensate for the errors due to the stochastic nature of the system. In such applications, the environment can also change, obstacles can move, or new objects might appear. Control strategies such as Receding Horizon Control (RHC) [107, 108, 109] are better suited for such on-line applications if they have a fast re-planning algorithm.

### *1.2.5 More on LQG-Based Methods*

In this subsection, we review some of the LQG-based methods that tackle similar problems.

Feedback-based Information RoadMap (FIRM) [110], [15] is a general framework to overcome the curse of history that attempts to solve an MDP in the sampled belief space. The graph-based solution of FIRM introduced an elegant method for solving POMDPS with continuous underlying spaces. However, attention is restricted to Gaussian [111, 112] belief spaces which can be insufficient in some problems. In [113] the stochastic control problem is reduced to a path planning algorithm in the spaces of poses×covariances, and two algorithms are given to minimize the execution time and minimize final covariance. The first algorithm extends classical graph-search methods, and the second a back-projection of uncertainty constraints in the grid-based space. [114] proposes an algorithm that restricts attention to the most-likely observation and finds trajectories using non-linear optimization methods.

Basic LQG methods [7] find locally optimal feedback laws. However, in these methods, the policy is independent of process and measurement uncertainties. Iterative Linear Quadratic Gaussian (iLQG) [34] generalizes the LQG framework to incorporate the process uncertainty with full observation (or an independent estimator) of the state. Several methods incorporate partial or noisy observations where the controller needs to actively gain information about the state. Belief roadmaps

(BRM) [115] and icLQG [116] which are based on Probabilistic Roadmaps (PRM) [117, 118, 119, 120, 121] provide locally optimal solutions, by combining iterative LQG with a roadmap. LQG-MP method [59] simulates LQG on a finite set of RRT generated paths and compares its performance on those paths to find the better trajectory. However, this method does not utilize the most likely observation assumption which was used in [122, 108] to make the belief propagation deterministic. Therefore, LQG-MP does not construct a trajectory; rather, it finds the best among given trajectories. [123] builds a belief tree over paths generated by RRT [124]; however, they use RRT* [125] to find the optimal underlying trajectory and then apply a variant of LQG-MP to find a global optimal trajectory in the belief space. In [126] a chance-constrained optimal control problem is solved by assuming fixed control gains on each segment of the trajectory. [127] through interleaving the iteration of the controller and estimator to find a locally optimal solution, in a setting with control-dependent process and observation uncertainty, but no obstacles. Moreover, their controller is only optimal under the fixed estimator gain assumption. [128] uses stochastic differential dynamic programming (sDDP) to extend the LQG-MP methods to roadmaps. In [35], they extend this method by performing the value iterations using iLQG which improves their speed by one order. In fact, their approach is a belief space variant of iLQG to perform value iteration. However, the time-complexity of the latter method is still of order 6 in state dimension. Moreover, the number of cycles to be performed for near convergence is not generally known. This method also takes a feasible solution such as a RRT-generated path and computes the control law by a backward recursion of the quadratic value function. Then, this policy is used to compute a new nominal trajectory starting from the initial distribution. The procedure is performed iteratively for new trajectories until it converges to the locally optimal trajectory. This is mainly due to the line search algorithm that is used in the Newton-like optimization methods that require a feasible solution to begin with and an appropriate step size to avoid divergence.

Generally, roadmap methods return an optimal trajectory instead of a feedback law. Therefore, re-planning becomes unavoidable because of large deviations from the nominal path caused by uncertainty and noise. However, unless the planning domain and horizon are small, computationally expensive methods are impractical [129] since, in case of a large deviation a new query for a new initial belief is requested. In our research, we provide a method whose core problem is computationally light and the number of optimization decision variables is the same as the number of control inputs. Moreover, in the existence of obstacles, the problem is still computationally efficient because of the low number of decision variables and hard constraints. Therefore, our method is scalable, and, as we will discuss later, it utilizes the stochasticity of the problem in its planning.

### *1.2.6 Model Predictive Control (MPC)-Based Methods*

Other closely related methods to our method are Model Predictive Control (MPC) or RHC-based methods[130, 28]. In MPC-based methods, at each sampling step and given the initial state of the system, a finite horizon open-loop optimal control problem is solved [131, 132, 133]. The first control in the optimal control sequence resulting from the optimization is applied to the plant and the new state of the system is used as the initial state for the next period. MPCs can cope with hard constraints on controls and states [134, 135], and therefore have been widely used in deterministic constrained problems where the evolution of the state is considered noiseless and the observations are perfect. An overview of industrial applications of MPCs is provided in [136]. Their stability and optimality results have been extensively studied in [131]. Although MPC solves a standard optimal control problem, it differs from the $H_2$ or $H_\infty$ linear optimal control problems in that they usually solve for infinite horizon while in MPC the optimal control problem is solved for a finite horizon [137, 138, 139]. This is the appealing advantage of MPCs for our research in that, unlike the traditional POMDP solvers which are used in obtaining off-line a feedback policy (which determines the optimal control for all (belief) states whose computation is expensive), MPCs have a natural on-line planning method for the current state of the plant. In most practical robotic problems, due to the inherent stochasticity of the problem resulting from unmodeled or unpredicted uncertainty, uncertainty in a robot's actions and inherent noise in sensor measurements or changes in environment map (such as moving objects), off-line plans are not reliable enough after execution of a few steps of planned actions. In such problems, the planner needs to re-plan to compensate and refine its policy. From the implementation point of view, MPC's solving of an open loop policy where the initial state is the current state to be controlled, can be considered as a mathematical program [140, 141]. Whereas, in determining the feedback control law, the solution of Hamilton-Jacobi-Bellman (HJB) equation [142, 143] basically deals with a differential or difference equation which is generally more difficult [144]. However, it is required in MPCs that the finite horizon control problem is solvable in a reasonable amount of time. Moreover, as mentioned before, MPCs have been extensively used for deterministic problems.

Much of the work on stochastic MPCs has been performed on robust planning over process uncertainty. There have been two major methods that have been practiced. In the first category, it is common to ignore the uncertainty in the planning and solve for control actions for a given initial state using the nominal model whose resulting control sequence is robustly stable for small disturbances under some conditions [28, 145, 146, 147]. A second approach is to robustly program for all possible disturbances or just account for a range of uncertainties [148, 149]. A major disadvantage of this approach is that the diameter of the tube that considers the trajectories resulting from the disturbances can become so large that the problem can become infeasible [150]. Moreover, the tube generated open-loop sequence might be significantly different from the infinite horizon feedback policy which tends to keep the trajectories in a small neighborhood with small dispersion from the nominal trajectory [151, 150, 28]. However, the most important disadvantage of these methods is their conservatism due to the fact that the tube-generated trajectories are a poor prediction of closed-loop behavior [28]. In the same category, state-dependent uncertainties have been discussed in [152]. The tube-based MPCs have been introduced to partly mitigate these problems [153, 154]. This method applies a local feedback about a nominal trajectory keeping the resultant trajectories of disturbances in a small neighborhood of the reference trajectory [155]. However, in these methods, uncertainties are assumed to be bounded. A class of other methods has considered soft constraints where the constraints need not be satisfied for all possible realization of uncertainties. Much of the attention in the literature regarding this area has been limited to linear systems (both process and observation models) with additive uncertainties. In most of the methods, the resulting optimizations are non-convex and the resulting programs are computationally expensive. Monte-Carlo based methods [156, 157], and related methods such as scenario approach [158], have also been successful in providing high confidence probabilistic guarantees for convex problems, and the results have been applied on MPCs in [27]. There is an extensive overview of the feedback control methods as well as recent developments in [159].

# 2. GENERAL BACKGROUND

In this chapter, we define the general background regarding the stochastic optimal control of a single-agent system. We will only consider the problem with imperfect state information, which is more general than the problem with perfect state information. In the next chapter, we define the specific problems that are tackled in this research.

## 2.1 Single-Agent Model

*Probability space (notation):* Let $\{\Omega, \mathscr{F}, P\}$ be a probability space with the random variables on some measurable space $(\mathbb{X}, \mathscr{B})$, where $\mathbb{X}$ is generally a Euclidean space with dimension of $n_x$ or a smooth manifold in this space, and $\mathscr{B}$ is the corresponding $\sigma$-algebra of Borel sets.

*Notations:* Let $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^{n_x}$, $\mathbf{u} \in \mathbb{U} \subset \mathbb{R}^{n_u}$, and $\mathbf{z} \in \mathbb{Z} \subset \mathbb{R}^{n_z}$ denote the state, control and observation vectors, respectively, and $\mathbf{f} : \mathbb{X} \times \mathbb{U} \times \mathbb{R} \to \mathbb{X}$, and $\mathbf{h} : \mathbb{X} \times \mathbb{R} \to \mathbb{Z}$ denote the process and measurement model, respectively.

*Discrete-time system equations:* We consider the general discrete-time system equations:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\omega}_t), \tag{2.1a}$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \boldsymbol{\nu}_t), \tag{2.1b}$$

where the $n_x$- and $n_z$-dimensional random sequences $\{\boldsymbol{\omega}_t, t \geq 0\}$ and $\{\boldsymbol{\nu}_t, t \geq 0\}$ are mutually independent zero-mean i.i.d. (independent, identically distributed), and $\mathbf{x}_0 \sim p_0(\cdot)$.

*Data history:* Let us define the data history of observations and actions for $1 \leq$

$t \leq K$ as $\mathbb{D}_t := \{\mathbf{z}_{0:t}, \mathbf{u}_{0:t-1}\}$, where $\mathbf{u}_{0:t-1}$ and $\mathbf{z}_{0:t}$ denote the actions and observations from beginning to time step $t$. Note there is no observation at time 0, and $\mathbf{z}_0$ is only defined artificially to model the initial distribution. This will be useful later in the definition of the control policy.

*The conditional distribution:* The conditional distribution of $\boldsymbol{\theta}_t := \mathbf{x}_t | \mathbb{D}_t, 1 \leq t \leq K$, denoted by $p_t$, is the conditional distribution of the original system. It is a sufficient statistic for the estimation and control of the original system. The evolution of $p_t$ is based on the Bayesian update equation, which can be summarized as a function $\tau_t : \mathbb{R} \times \mathbb{I} \times \mathbb{U} \times \mathbb{Z} \rightarrow \mathbb{I}$ [2, 12, 15], where $p_{t+1} = \tau_t(p_t, \mathbf{u}_t, \mathbf{z}_{t+1})$, $p_0$ is given, and $\mathbb{I}$ denotes the space of conditional distributions. Also we define $\boldsymbol{\theta}_0 := \mathbf{x}_0$. We will denote $p_t(\mathbf{x}_t = \mathbf{x}, \mathbb{D}_t = \mathbb{D})$ by $p_t(\mathbf{x}, \mathbb{D})$ throughout the text.

Next, we revisit some of the concepts related to the conditional distribution and derive $\tau_t$.

### *2.1.1 Features of the Conditional Distribution*

*Sufficient statistic:* A statistic is a function of the observations $\mathbf{z}_{0:t}$. A statistic $g(\mathbf{z}_{0:t})$ is said to be "sufficient" for the parameter set $\mathbb{\Theta}$ if the conditional density of $\mathbf{z}_{0:t}$ given $g(\mathbf{z}_{0:t})$, does not depend on $\boldsymbol{\theta}$. That is, $p(\mathbf{z}_{0:t} | g(\mathbf{z}_{0:t}, \boldsymbol{\theta}))$ does not depend on $\boldsymbol{\theta}$. It is proved in [2] that $g(\mathbf{z}_{0:t})$ is a sufficient statistic for $\mathbb{\Theta}$ if and only if there are functions $q_1, q_2$ such that:

$$p(\mathbf{z}_{0:t} | \boldsymbol{\theta}) = q_1(g(\mathbf{z}_{0:t}), \boldsymbol{\theta}) q_2(\mathbf{z}_{0:t}), \boldsymbol{\theta} \in \mathbb{\Theta}$$

That is, if $p(\mathbf{z}_{0:t} | \boldsymbol{\theta})$ depends on $\boldsymbol{\theta}$ only through $g(\mathbf{z}_{0:t})$.

*Conditional distribution as a sufficient statistic:* In a system where the state is only partially observed, the controller needs to keep track of its knowledge about the current state of the system given the data history. The conditional distribution of the state given the data history is a sufficient statistic for the given history. That means that it contains all the necessary information for decision making at time $t$. Here, $g(\mathbf{z}_{0:t}) = p_{\mathbf{X}_t|\mathbf{Z}_{0:t};\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t};\mathbf{u}_{0:t-1};p_0)$. Therefore,

$$\begin{aligned} p_{\mathbf{Z}_{0:t}|\mathbf{U}_{0:t-1}\mathbf{X}_t}(\mathbf{z}_{0:t}|\mathbf{u}_{0:t-1},\mathbf{x}) &= \frac{p_{\mathbf{X}_t|\mathbf{Z}_{0:t},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t},\mathbf{u}_{0:t-1})p_{\mathbf{Z}_{0:t}|\mathbf{U}_{0:t-1}}(\mathbf{z}_{0:t}|\mathbf{u}_{0:t-1})}{p_{\mathbf{X}_t|\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{u}_{0:t-1})} \\ &= \frac{p_{\mathbf{X}_t|\mathbf{Z}_{0:t},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t},\mathbf{u}_{0:t-1})p_{\mathbf{Z}_{0:t}|\mathbf{U}_{0:t-1}}(\mathbf{z}_{0:t}|\mathbf{u}_{0:t-1})}{p_{\mathbf{X}_t|\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{u}_{0:t-1})} \\ &= q_1(p_{\mathbf{X}_t|\mathbf{Z}_{0:t},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t},\mathbf{u}_{0:t-1}),\mathbf{x})q_2(\mathbf{z}_{0:t}), \end{aligned}$$

where

$$q_1(p_{\mathbf{X}_t|\mathbf{Z}_{0:t},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t},\mathbf{u}_{0:t-1}),\mathbf{x}) = p_{\mathbf{X}_t|\mathbf{Z}_{0:t},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t},\mathbf{u}_{0:t-1})/p_{\mathbf{X}_t|\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{u}_{0:t-1}),$$

and $q_2(\mathbf{z}_{0:t}) = p_{\mathbf{Z}_{0:t}|\mathbf{U}_{0:t-1}}(\mathbf{z}_{0:t}|\mathbf{u}_{0:t-1})$. Therefore, the conditional distribution over the augmented state is indeed a sufficient statistic for the parameter.

*Transition function:* $T_t : \mathbb{X} \times \mathbb{U} \times \mathbb{X} \to \mathbb{R}$ is the transition function describing the probability of transitioning from state $\mathbf{x}'$ to state $\mathbf{x}$ after taking action $\mathbf{u}$ at time step $t$, where $T_t(\mathbf{x},\mathbf{u},\mathbf{x}') := p_{\mathbf{X}_{t+1}|\mathbf{U}_t,\mathbf{X}_t}(\mathbf{x}|\mathbf{u},\mathbf{x}')$. Note that this function, which is an equivalent representation of the process model $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t,\mathbf{u}_t,\boldsymbol{\omega}_t)$, describes the uncertainty in the effect of the action or process uncertainty.

*Likelihood function:* $\Omega_t : \mathbb{Z}\times\mathbb{X} \to \mathbb{R}$ is the likelihood function describing the probability of observing $\mathbf{z}$ at state $\mathbf{x}$ at time step $t$, where $\Omega_t(\mathbf{z},\mathbf{x}) := p_{\mathbf{Z}_t|\mathbf{X}_t}(\mathbf{z}|\mathbf{x})$. Similarly, this function, which equivalently describes the observation model $\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t,\boldsymbol{\nu}_t)$, is needed to describe the uncertainty in perception or measurement uncertainty.

*Bayesian update:* Since the system is only partially observable, there is a need for the estimation module to update the conditional distribution after taking an action and perceiving an observation. The well-known Bayesian update equation [12, 2, 14] gives us the general mechanism to update the conditional distribution over the state after taking an action and perceiving an observation:

$$p_{t+1}(\mathbf{x}, \mathbb{D}) = \eta \Omega_{t+1}(\mathbf{z}, \mathbf{x}) \int_{\mathbf{x}' \in \mathbb{X}} T_t(\mathbf{x}, \mathbf{u}, \mathbf{x}') p_t(\mathbf{x}', \mathbb{D}) d\mathbf{x}', \tag{2.2}$$

where $\eta$ is a normalizing constant. This equation is summarized as $p_{t+1} = \tau_t(p_t, \mathbf{u}_t, \mathbf{z}_{t+1})$.

*Information state:* $\varUpsilon_t$ is an information state for the stochastic system (3.1) if it is both a function of $\mathbb{D}_t$, and $\varUpsilon_{t+1}$ can be determined from $\varUpsilon_t$, $\mathbf{z}_{t+1}$ and $\mathbf{u}_t$ [2]. We show that the conditional distribution over the state is a information state. Moreover, it is a sufficient statistic for the stochastic control problem.

*Conditional distribution is an information state:* We now derive the Bayesian recursion formula for the conditional distribution:

$$p_{\mathbf{X}_t|\mathbf{Z}_{0:t},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t-1}) = \frac{p_{\mathbf{Z}_t|\mathbf{X}_t}(\mathbf{z}_t|\mathbf{x}) p_{\mathbf{X}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1})}{p_{\mathbf{Z}_{0:t},\mathbf{U}_{0:t-1}}(\mathbf{z}_{0:t}, \mathbf{u}_{0:t-1})}.$$

We have:

$$\begin{aligned}
&p_{\mathbf{X}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1}) \\
&= \int_{\mathbf{x}' \in \mathbb{X}} p_{\mathbf{X}_t|\mathbf{X}_{t-1},\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{x}', \mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1}) p_{\mathbf{X}_{t-1}|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1}}(\mathbf{x}'|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1}) d\mathbf{x}' \\
&= \int_{\mathbf{x}' \in \mathbb{X}} p_{\mathbf{X}_t|\mathbf{U}_{t-1},\mathbf{X}_{t-1}}(\mathbf{x}|\mathbf{u}_{t-1}, \mathbf{x}') p_{\mathbf{X}_{t-1}|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-2}}(\mathbf{x}'|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-2}) d\mathbf{x}' \\
&= \int_{\mathbf{x}' \in \mathbb{X}} T_{t-1}(\mathbf{x}, \mathbf{u}, \mathbf{x}') p_{t-1}(\mathbf{x}', \mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-2}, p_0) d\mathbf{x}' \\
&:= \Psi_t(p_{\mathbf{X}_{t-1}|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-2}}(\cdot|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-2}), \mathbf{u}_{t-1})(\mathbf{x}) \\
&= \Psi_t(p_{t-1}(\cdot, \mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-2}, p_0), \mathbf{u}_{t-1})(\mathbf{x}),
\end{aligned} \tag{2.3}$$

where $p_{t-1}(\mathbf{x}', \mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-2}, p_0) = p_{\mathbf{X}_{t-1}|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-2}}(\mathbf{x}'|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-2})$. Moreover,

$$\begin{aligned}
p_{\mathbf{X}_t|\mathbf{Z}_{0:t},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t-1}) &= \frac{p_{\mathbf{Z}_t|\mathbf{X}_t}(\mathbf{z}_t|\mathbf{x}) p_{\mathbf{X}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1})}{p_{\mathbf{Z}_{0:t},\mathbf{U}_{0:t-1}}(\mathbf{z}_{0:t}, \mathbf{u}_{0:t-1})} \\
&= \frac{p_{\mathbf{Z}_t|\mathbf{X}_t}(\mathbf{z}_t|\mathbf{x}) p_{\mathbf{X}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1})}{\int_{\mathbf{x}\in\mathbb{X}} p_{\mathbf{Z}_t|\mathbf{X}_t}(\mathbf{z}_t|\mathbf{x}) p_{\mathbf{X}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1}) d\mathbf{x}} \\
&= \frac{\Omega_t(\mathbf{z}, \mathbf{x}) p_{\mathbf{X}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1})}{\int_{\mathbf{x}\in\mathbb{X}} \Omega_t(\mathbf{z}, \mathbf{x}) p_{\mathbf{X}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1}) d\mathbf{x}} \\
&:= \Phi_t(p_{\mathbf{X}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1}}(\cdot|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1}), \mathbf{z}_t)(\mathbf{x}). \qquad (2.4)
\end{aligned}$$

Hence, we have:

$$\begin{aligned}
p_{\mathbf{X}_t|\mathbf{Z}_{0:t},\mathbf{U}_{0:t-1}}(\mathbf{x}|\mathbf{z}_{0:t}, \mathbf{u}_{0:t-1}) &= \Phi_t[\Psi_t(p_{\mathbf{X}_{t-1}|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-2}}(\cdot|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-2}), \mathbf{u}_{t-1}), \mathbf{z}_t] \\
&:= \tau_t(p_{\mathbf{X}_{t-1}|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-2}}(\cdot|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-2}), \mathbf{u}_{t-1}, \mathbf{z}_t), \qquad (2.5)
\end{aligned}$$

which is the same formula obtained in (2.2). Therefore, we can compute the conditional distribution at time $t$ through the conditional distribution at time $t-1$, using $\mathbf{z}_t$ and $\mathbf{u}_{t-1}$. This also proves that conditional distribution over state is an information state. Note that in order to solve the above recursion, we need the initial condition:

$$p_{\mathbf{X}_0|\mathbf{Z}_0,\mathbf{U}_{-1}}(\cdot|\mathbf{z}_0, \mathbf{u}_{-1}) := p_{\mathbf{X}_0}(\cdot). \qquad (2.6)$$

## 2.2 Elements of the Stochastic Control Problem

*Incremental cost function:* Assuming that the time horizon is finite, $K < \infty$, $c_t(\mathbf{x}_t, \mathbf{u}) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ denotes the one-step or immediate cost incurred by executing action $\mathbf{u}$ at state $\mathbf{x}_t$. Moreover, $c_K(\mathbf{x}_K)$ denotes the terminal cost.

*Policy function:* The feedback policy (planner or the feedback control law), is a sequence of functions $\boldsymbol{\pi} = \{\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \cdots\}$ where $\boldsymbol{\pi}_t : \mathbb{Z}^{t+1} \to \mathbb{U}$ specifies the action given the output (i.e., the observations). In a problem with perfect state measurements, the output of the system is a direct function of the state and therefore, the policy is state-dependent. Thus, $\mathbf{u}_t = \boldsymbol{\pi}_t(\mathbf{z}_{0:t})$, where $\boldsymbol{\pi} = \{\boldsymbol{\pi}_0, \cdots, \boldsymbol{\pi}_t\}$ is a policy denoted by a finite sequence (since $K < \infty$). A policy is feasible if $\mathbf{u}_t = \boldsymbol{\pi}_t(\mathbf{z}_{0:t}) \in \mathbb{U}$. We denote the space of feasible policies by $\mathbb{\Pi}$.

*Cost associated with the policy:* Let $\boldsymbol{\pi} \in \mathbb{\Pi}$, and $\{\mathbf{x}_t^{\boldsymbol{\pi}}\}$, $\{\mathbf{u}_t^{\boldsymbol{\pi}}\}$ and $\{\mathbf{z}_t^{\boldsymbol{\pi}}\}$ be the random processes associated with (and dependent on) that policy. We can define the cost function $J_{\boldsymbol{\pi}} : \mathbb{X}^{K+1} \times \mathbb{U}^K \to \mathbb{R}$ associated with $\boldsymbol{\pi}$ as:

$$J_{\boldsymbol{\pi}} := \sum_{t=0}^{K-1} c_t(\mathbf{x}_t^{\boldsymbol{\pi}}, \mathbf{u}_t^{\boldsymbol{\pi}}) + c_K(\mathbf{x}_K^{\boldsymbol{\pi}}).$$

For notational simplicity, we denote the cost associated with the policy $\boldsymbol{\pi}$ by $\sum\limits_{t=0}^{K-1} c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t) +$ $c_K^{\boldsymbol{\pi}}(\mathbf{x}_K)$. A proper choice of this cost function is an important aspect of the overall modeling of the problem.

*Cost-to-go function:* Due to the randomness of the processes $\{\mathbf{x}_t^{\boldsymbol{\pi}}\}$ and $\{\mathbf{u}_t^{\boldsymbol{\pi}}\}$, $J_{\boldsymbol{\pi}}$ is a random variable. Therefore, we define the cost-to-go as the expected cost $\mathbb{E}[J_{\boldsymbol{\pi}}]$ which is deterministic, with the expectation taken over all randomness. This expectation can be written as:

$$\begin{aligned}
\mathbb{E}[J_{\boldsymbol{\pi}}(\mathbf{x}_{0:K}, \mathbf{u}_{0:K-1})] &= \mathbb{E}[\sum_{t=0}^{K-1} c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t) + c_K^{\boldsymbol{\pi}}(\mathbf{x}_K)] \\
&= \mathbb{E}[\sum_{t=0}^{K-1} \mathbb{E}[c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t)|\mathcal{D}_t] + \mathbb{E}[c_K^{\boldsymbol{\pi}}(\mathbf{x}_K)|\mathcal{D}_K]] \\
&= \mathbb{E}[\sum_{t=0}^{K-1} \int_{\mathbb{X}} [c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t) p_t(\mathbf{x}_t|\mathbf{z}_{0:t}, \mathbf{u}_{0,t-1}) d\mathbf{x}_t] \\
&\qquad + \int_{\mathbb{X}} [c_K^{\boldsymbol{\pi}}(\mathbf{x}_K) p_K(\mathbf{x}_K|\mathbf{z}_{0:K}, \mathbf{u}_{0,K-1}) d\mathbf{x}_K]]
\end{aligned}$$

$$=: \mathbb{E}[\sum_{t=0}^{K-1} c_t^{\boldsymbol{\pi},p}(p_t, \mathbf{u}_t) + c_K^{\boldsymbol{\pi},p}(p_K)]$$

$$=: \mathbb{E}[J'_{\boldsymbol{\pi}}(p_{0:K}, \mathbf{u}_{0:K-1})]$$

where $c_t^{\boldsymbol{\pi},p}$, $c_K^{\boldsymbol{\pi},p}$, and $J'$ are defined using the above equations with respect to the conditional distribution, and the last expectation is taken over all possible conditional distributions.

*Problem ingredients:* The stochastic control problem can be represented by an $n-$tuple: $\{\mathbb{X}, \mathbb{U}, \mathbb{Z}, p_0, T_t, \Omega_t, c_t, K\}$.

**Problem 1 General stochastic control problem** *The objective in our stochastic control problem is to find an optimal policy which minimizes the cost-to-go function. Therefore, the problem can be formulated as follows:*

$$\min_{\boldsymbol{\pi}\in\Pi} \mathbb{E}[J_{\boldsymbol{\pi}}] \tag{2.7}$$

*and the optimal policy $\boldsymbol{\pi}^*$ is defined as:*

$$\boldsymbol{\pi}^* := \arg\min_{\boldsymbol{\pi}\in\Pi} \mathbb{E}[J_{\boldsymbol{\pi}}]$$

Note that since the state is not directly observed, the optimal policy is not a Markovian policy (it can become Markov if the entire trajectory of observations is defined as a variable). However, as we showed in equation (2.5), the information state does not depend on $\boldsymbol{\pi}$. Therefore, the optimal policy is only a function of the information state. We show in the next section that the optimal policy is separated. We call a policy separated if $\boldsymbol{\pi}_t$ depends on the output $\mathbf{z}_{0:t}$ only through the information state, that is, $\mathbf{u}_t = \boldsymbol{\pi}_t(p_t(\cdot|\mathbf{z}_{0:t}))$, and $\Pi_S$ denotes the space of all separated policies.

## 2.3 Theoretical Solution of the General Problem

In this section, we provide the solution of the POMDP problem. It is proven in [2, 12] that the optimal policy for problem (1) can be found using the dynamic programming equations. We provide the result without further elaboration and refer the reader to the book [2] for its proof and further details.

**Theorem 1** *Define recursively the functions $V_t(p)$, $0 \le t \le K$, $p \in \mathbb{I}$, by*

$$V_K(p) := \mathbb{E}\{c_K(\mathbf{x}_K)|p_K = p\}, \tag{2.8}$$

$$V_t(p) := \inf_{\mathbf{u}\in\mathbf{U}} \mathbb{E}\{c_t(\mathbf{x}_t, \mathbf{u}) + V_{t+1}(\tau_t(p, \mathbf{u}, \mathbf{z}_{t+1}))|p_t = p\} \tag{2.9}$$

*(i) Let $\boldsymbol{\pi} \in \mathbb{\Pi}$, then*

$$V_t(p_t(\mathbb{D}_t)) \le J_t^{\boldsymbol{\pi}} \tag{2.10}$$

*(i) Let $\boldsymbol{\pi} \in \mathbb{\Pi}_S$, such that for all $p \in \mathbb{I}$, $\boldsymbol{\pi}_t(p)$ achieves the minimum in (2.9); then $\boldsymbol{\pi}$ is optimal and $V_t(p_t(\mathbb{D}_t)) = J_t^{\boldsymbol{\pi}}$ w.p.1.*

The optimal policy is only a function of the information state and is a separated policy. This solution involves solving an optimization problem in the space of actions, for all information states. However, the information-state space over a *continuous* finite-dimensional state space is an *infinite*-dimensional *function* space, which makes finding the optimal policy of the problem (1) through the solution of Theorem 1 an intractable task. The computational complexity of such an effort is PSPACE-complete, which is higher in the hierarchy than the NP-complete problems [160]. However, this problem is significant for many applications, such as many robotics problems. This has motivated research into suboptimal or near-optimal solutions of the problem using techniques in optimization, control and algorithms theory. In the next section, we provide our proposed method for tackling this problem in order to find near-optimal solutions under a small-noise assumption, which can be found in polynomial time.

# 3. DECOUPLING PRINCIPLE: FOUR PROBLEMS, FOUR RESULTS

In this chapter, we define the four specific closely-related stochastic optimal control problems that we tackle in this research. First, we consider the single-agent and multi-agent problems with perfect state information, and then we proceed to the problems with imperfect state information. Then, we state the main results for each of these problems. In the next chapters, we lay down our theoretical approach for each of these problems and prove the decoupling principle for each one. Multiple results and related aspects of this dissertation have been presented in [161, 162, 163, 164, 62, 165, 166, 167, 168, 169, 170].

## 3.1 Single-Agent Model

*Probability space (notation):* Let $\{\Omega, \mathscr{F}, P\}$ be a probability space with the random variables on some measurable space $(\mathbb{X}, \mathscr{B})$, where $\mathbb{X}$ is generally a Euclidean space with dimension of $n_x$ or a smooth manifold in this space, and $\mathscr{B}$ is the corresponding $\sigma$-algebra of Borel sets.

*Notations:* Let $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^{n_x}$, $\mathbf{u} \in \mathbb{U} \subset \mathbb{R}^{n_u}$, and $\mathbf{z} \in \mathbb{Z} \subset \mathbb{R}^{n_z}$ denote the state, control and observation vectors, respectively, and $\mathbf{f} : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$ and $\boldsymbol{\sigma}^{\mathbf{f}} : \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ denote the drift and diffusion terms of the motion model, $\mathbf{h} : \mathbb{X} \to \mathbb{Z}$ and $\boldsymbol{\sigma}^{\mathbf{h}} : \mathbb{R} \to \mathbb{R}^{n_z \times n_z}$ denote the drift and diffusion terms of the observation model, respectively.

*Discrete-time system equations:* We consider the general discrete-time system equations with additive noise as:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \epsilon \boldsymbol{\sigma}^{\mathbf{f}}(t)\mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}}), \tag{3.1a}$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \epsilon \boldsymbol{\sigma}^{\mathbf{h}}(t)\mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{v}}), \tag{3.1b}$$

where the $n_x$- and $n_z$-dimensional Gaussian random sequences $\{\mathbf{w}_t, t \geq 0\}$ and $\{\mathbf{v}_t, t \geq 0\}$ are mutually independent zero-mean i.i.d. (independent, identically distributed), $\epsilon > 0$, and $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \epsilon^2 \boldsymbol{\Sigma}_{\mathbf{x}_0})$. Define $\mathbf{a} : \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$, $\mathbf{a} := \boldsymbol{\sigma}(\boldsymbol{\sigma})^T = (a_{j,k})_{0 \leq j,k \leq n_x}$, and let $\mathbf{f} = (f_j)_{0 \leq j \leq n_x}$. We assume the drift and diffusion coefficients, $f_j, a_{j,k}$, are twice continuously differentiable, bounded and uniformly Lipschitz continuous functions, and that the diffusion matrix is uniformly positive-definite (hence, non-degenerate). We also assume similar smoothness conditions for $\mathbf{h}$ and $\boldsymbol{\sigma}^{\mathbf{h}}$ as $\mathbf{f}$ and $\boldsymbol{\sigma}^{\mathbf{f}}$, respectively. Note that, at times for simplicity, we will denote the process and observation models by $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t)$ and $\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{v}_t)$, but we only mean it as a short form for the above equations, particularly the dependence on the noise, unless otherwise stated.

*Data history:* Let us define the data history of observations and actions for $1 \leq t \leq K$ as $\mathbb{D}_t := \{\mathbf{z}_{1:t}, \mathbf{u}_{0:t-1}\}$, where $\mathbf{u}_{0:t-1}$ and $\mathbf{z}_{0:t}$ denote the actions and observations from beginning to time step $t$.

*The conditional distribution:* The conditional distribution of $\boldsymbol{\theta}_t := \mathbf{x}_t | \mathbb{D}_t, 1 \leq t \leq K$, denoted by $p_t$, is the conditional distribution of the original system. It is a sufficient statistic for the estimation and control of the original system. The evolution of $p_t$ is based on the Bayesian update equation, summarized as a function $\tau_t : \mathbb{R} \times \mathbb{I} \times \mathbb{U} \times \mathbb{Z} \to \mathbb{I}$ [2, 12, 15], where $p_{t+1} = \tau_t(p_t, \mathbf{u}_t, \mathbf{z}_{t+1})$, $p_0$ is given, and $\mathbb{I}$ denotes the space of conditional distributions. For our system, $p_0 = \mathcal{N}(\bar{\mathbf{x}}_0, \epsilon^2 \boldsymbol{\Sigma}_{\mathbf{x}_0})$. Also we define $\boldsymbol{\theta}_0 := \mathbf{x}_0$. We will denote $p_t(\mathbf{x}_t = \mathbf{x}, \mathbb{D}_t = \mathbb{D})$ by $p_t(\mathbf{x}, \mathbb{D})$ throughout the text.

## 3.2 Multi-Agent Model

*Agent index set:* We assume there are $m$ agents with the index set of $i \in \mathcal{I} := \{1, \cdots, m\}$.

*Notations:* For agent $i$, let $\mathbf{x}^i \in \mathbb{X}^i \subset \mathbb{R}^{n_x^i}$, $\mathbf{u}^i \in \mathbb{U}^i \subset \mathbb{R}^{n_u^i}$, and $\mathbf{z}^i \in \mathbb{Z}^i \subset \mathbb{R}^{n_z^i}$

denote its state, control and observation vectors, respectively, and $\mathbf{f}^i : \mathbb{X}^i \times \mathbb{U}^i \to \mathbb{X}^i$ and $\boldsymbol{\sigma}^{\mathbf{f}_i} : \mathbb{R} \to \mathbb{R}^{n_x^i \times n_x^i}$ denote the drift and diffusion terms of the motion model, $\mathbf{h}^i : \mathbb{X}^i \to \mathbb{Z}^i$ and $\boldsymbol{\sigma}^{\mathbf{h}_i} : \mathbb{R} \to \mathbb{R}^{n_z^i \times n_z^i}$ denote the drift and diffusion terms of the observation model, respectively. We assume independent process and observation dynamics for different agents.

*Discrete-time system equations:* We consider the general discrete-time system equations with additive noise as:

$$\mathbf{x}_{t+1}^i = \mathbf{f}^i(\mathbf{x}_t^i, \mathbf{u}_t^i) + \epsilon\boldsymbol{\sigma}^{\mathbf{f}_i}(t)\mathbf{w}_t^i, \quad \mathbf{w}_t^i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}^i}), \tag{3.2a}$$

$$\mathbf{z}_t^i = \mathbf{h}^i(\mathbf{x}_t^i) + \epsilon\boldsymbol{\sigma}^{\mathbf{h}_i}(t)\mathbf{v}_t^i, \qquad \mathbf{v}_t^i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{v}^i}), \tag{3.2b}$$

where the $n_x^i$- and $n_z^i$-dimensional random sequences $\{\mathbf{w}_t^i, t \geq 0\}$ and $\{\mathbf{v}_t^i, t \geq 0\}$ are mutually independent zero-mean i.i.d. (independent, identically distributed), $\epsilon > 0$, and $\mathbf{x}_0^i \sim \mathcal{N}(\bar{\mathbf{x}}_0^i, \epsilon^2\boldsymbol{\Sigma}_{\mathbf{x}_0^i})$. Define $\mathbf{a}^i : \mathbb{R} \to \mathbb{R}^{n_x^i \times n_x^i}$, $\mathbf{a}^i := \boldsymbol{\sigma}^i(\boldsymbol{\sigma}^i)^T = (a_{j,k})_{0 \leq j,k \leq n_x^i}$, and let $\mathbf{f}^i = (f_j^i)_{0 \leq j \leq n_x^i}$. We assume the drift and diffusion coefficients, $f_j^i, a_{j,k}$, are twice continuously differentiable, bounded and uniformly Lipschitz continuous functions. and that the diffusion matrix is uniformly positive-definite (hence, non-degenerate). We also assume similar smoothness conditions for $\mathbf{h}^i$ and $\boldsymbol{\sigma}^{\mathbf{h}_i}$ as $\mathbf{f}^i$ and $\boldsymbol{\sigma}^{\mathbf{f}_i}$, respectively.

*Data history:* Let us define the data history of observations and actions of agent $i$ for $1 \leq t \leq K$ as $\mathbb{D}_t^i := \{\mathbf{z}_{1:t}^i, \mathbf{u}_{0:t-1}^i\}$.

*The conditional distribution:* The conditional distribution of $\boldsymbol{\theta}_t^i := \mathbf{x}_t^i | \mathbb{D}_t^i, 1 \leq t \leq K$, denoted by $p_t^i$, is the conditional distribution of the system. It is proven to be a sufficient statistic for the estimation and control of the systems. The evolution of $p_t^i$ is based on the Bayesian update equation summarized as $p_{t+1}^i = \tau_t(p_t^i, \mathbf{u}_t^i, \mathbf{z}_{t+1}^i)$ and $p_0^i$ is given. For our system, $p_0 = \mathcal{N}(\bar{\mathbf{x}}_0^i, \epsilon^2\boldsymbol{\Sigma}_{\mathbf{x}_0^i})$. Also we define $\boldsymbol{\theta}_0^i := \mathbf{x}_0^i$.

*Joint agent spaces:* Let us define the Cartesian products of the individual agent spaces as the joint agent spaces denoted by $\mathbb{X}^{\mathcal{I}}, \mathbb{U}^{\mathcal{I}}, \mathbb{Z}^{\mathcal{I}}$, and $\mathbb{I}^{\mathcal{I}}$. Similarly, denote by superscript $\mathcal{I}$ the appropriate collection of joint agent variables, e.g., $\mathbf{u}_t^{\mathcal{I}} = [(\mathbf{u}_t^1)^T, \cdots, (\mathbf{u}_t^m)^T]^T$. Similarly, for the states $\mathbf{x}_t^{\mathcal{I}}$, observations $\mathbf{z}_t^{\mathcal{I}}$, etc. The dynamics of this concatenated set of all agent states can be described by an appropriate block matrix concatenation of the joint dynamics Jacobians and just a simple set collection of feedback policies that are defined precisely later.

Now we proceed to the specific problem definitions.

## 3.3 Problem Definitions

We consider all the problems in discrete-time. Later, we dedicate one section for each of these problems.

**Problem 2 Single-Agent Stochastic Optimal Control with Perfect State Information** *Given an initial state* $\mathbf{x}_0$*, solve to determine an optimal or near-optimal policy for*

$$\begin{aligned} \min_{\boldsymbol{\pi}} \ & \mathbb{E}[\sum_{t=0}^{K-1} c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t) + c_K^{\boldsymbol{\pi}}(\mathbf{x}_K)] \\ & s.t.\ \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \epsilon \boldsymbol{\sigma}^{\mathbf{f}}(t)\mathbf{w}_t, \end{aligned} \tag{3.3}$$

*where the optimization is over continuously differentiable Markov, i.e., time-varying state-feedback policies, with*

- $J_{\boldsymbol{\pi}} : \mathbb{\Pi} \to \mathbb{R}$ *is the cost function, and* $J_{\boldsymbol{\pi}} := \sum_{t=0}^{K-1} c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t) + c_K^{\boldsymbol{\pi}}(\mathbf{x}_K)$*;*
- $\boldsymbol{\pi} \in \mathbb{\Pi}$ *defines the policy, where* $\boldsymbol{\pi} := \{\boldsymbol{\pi}_0, \cdots, \boldsymbol{\pi}_t\}$*;*
- $\boldsymbol{\pi}_t : \mathbb{X} \to \mathbb{U}$ *specifies the optimal action:* $\mathbf{u}_t = \boldsymbol{\pi}_t(\mathbf{x}_t)$*;*
- $K > 0$ *is the planning horizon;*
- $c_t^{\boldsymbol{\pi}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ *is the incremental cost; and,*

- $c^{\boldsymbol{\pi}}_K : \mathbb{X} \rightarrow \mathbb{R}$ *is the terminal cost.*

**Problem 3 Centralized Multi-Agent Stochastic Optimal Control with Perfect State Information** *Given an initial joint state $\mathbf{x}^{\mathcal{I}}_0$, solve to determine an optimal or near-optimal policy for*

$$\min_{\boldsymbol{\pi}^{\mathcal{I}}} \ \mathbb{E}[\sum_{t=0}^{K_{\mathcal{I}}-1} c^{\boldsymbol{\pi}^{\mathcal{I}}}_t(\mathbf{x}^{\mathcal{I}}_t, \mathbf{u}^{\mathcal{I}}_t) + c^{\boldsymbol{\pi}^{\mathcal{I}}}_{K_{\mathcal{I}}}(\mathbf{x}^{\mathcal{I}}_{K_{\mathcal{I}}})]$$

$$s.t. \ \mathbf{x}^i_{t+1} = \mathbf{f}^i(\mathbf{x}^i_t, \mathbf{u}^i_t) + \epsilon\boldsymbol{\sigma}^{\mathbf{f}_i}(t)\mathbf{w}^i_t, \ \forall i \in \mathcal{I}, \tag{3.4}$$

*where the optimization is over continuously differentiable Markov, i.e., time-varying state-feedback policies, with*

- $J_{\boldsymbol{\pi}^{\mathcal{I}}} : \mathbb{\Pi}^{\mathcal{I}} \rightarrow \mathbb{R}$ *is the cost function, and* $J_{\boldsymbol{\pi}^{\mathcal{I}}} := \sum_{t=0}^{K_{\mathcal{I}}-1} c^{\boldsymbol{\pi}^{\mathcal{I}}}_t(\mathbf{x}^{\mathcal{I}}_t, \mathbf{u}^{\mathcal{I}}_t) + c^{\boldsymbol{\pi}^{\mathcal{I}}}_{K_{\mathcal{I}}}(\mathbf{x}^{\mathcal{I}}_{K_{\mathcal{I}}})$;
- $\boldsymbol{\pi}^{\mathcal{I}} \in \mathbb{\Pi}^{\mathcal{I}}$ *defines the policy, where* $\boldsymbol{\pi}^{\mathcal{I}} := \{\boldsymbol{\pi}^1, \cdots, \boldsymbol{\pi}^m\}$, *and* $\boldsymbol{\pi}^i := \{\boldsymbol{\pi}^i_0, \cdots, \boldsymbol{\pi}^i_t\}$;
- $\boldsymbol{\pi}^i_t : \mathbb{X}^{\mathcal{I}} \rightarrow \mathbb{U}^i$ *specifies the optimal action for agent i:* $\mathbf{u}^i_t = \boldsymbol{\pi}^i_t(\mathbf{x}^{\mathcal{I}}_t)$;
- $K_{\mathcal{I}} := \max_{i \in \mathcal{I}} K_i$, *and* $K_i > 0$ *is agent i's planning horizon;*
- $c^{\boldsymbol{\pi}^{\mathcal{I}}}_t : \mathbb{X}^{\mathcal{I}} \times \mathbb{U}^{\mathcal{I}} \rightarrow \mathbb{R}$ *is the incremental cost; and,*
- $c^{\boldsymbol{\pi}^{\mathcal{I}}}_{K_{\mathcal{I}}} : \mathbb{X}^{\mathcal{I}} \rightarrow \mathbb{R}$ *is the terminal cost.*

**Problem 4 Single-Agent Stochastic Optimal Control with Imperfect State Information** *Given an initial distribution $p_0$, solve for an optimal or near-optimal policy:*

$$\min_{\boldsymbol{\pi}} \mathbb{E}[\sum_{t=0}^{K-1} c^{\boldsymbol{\pi}}_t(\mathbf{x}_t, \mathbf{u}_t) + c^{\boldsymbol{\pi}}_K(\mathbf{x}_K)]$$

$$s.t. \ \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \epsilon\boldsymbol{\sigma}^{\mathbf{f}}(t)\mathbf{w}_t, \tag{3.5a}$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \epsilon\boldsymbol{\sigma}^{\mathbf{h}}(t)\mathbf{v}_t, \tag{3.5b}$$

*where the optimization is over continuously differentiable time-varying observation-trajectory-feedback policies, and:*

- $J_{\boldsymbol{\pi}} : \Pi \to \mathbb{R}$ *is the cost function, and* $J_{\boldsymbol{\pi}} := \sum_{t=0}^{K-1} c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t) + c_K^{\boldsymbol{\pi}}(\mathbf{x}_K)$*;*
- $\boldsymbol{\pi} \in \Pi$ *defines the policy, where* $\boldsymbol{\pi} := \{\boldsymbol{\pi}_0, \cdots, \boldsymbol{\pi}_t\}$*;*
- $\boldsymbol{\pi}_t : \mathbb{Z}^t \to \mathbb{U}$ *specifies the optimal action:* $\mathbf{u}_t = \boldsymbol{\pi}_t(\mathbf{z}_{1:t})$*;*
- $K > 0$ *is the planning horizon;*
- $c_t^{\boldsymbol{\pi}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ *is the incremental cost; and,*
- $c_K^{\boldsymbol{\pi}} : \mathbb{X} \to \mathbb{R}$ *is the terminal cost.*

**Problem 5** ***Centralized Multi-Agent Stochastic Optimal Control with Imperfect State Information*** *Given an initial joint distribution* $p_0^{\mathcal{I}}$*, solve for an optimal or near-optimal policy:*

$$\min_{\boldsymbol{\pi}^{\mathcal{I}}} \mathbb{E}[\sum_{t=0}^{K_{\mathcal{I}}-1} c_t^{\boldsymbol{\pi}^{\mathcal{I}}}(\mathbf{x}_t^{\mathcal{I}}, \mathbf{u}_t^{\mathcal{I}}) + c_{K_{\mathcal{I}}}^{\boldsymbol{\pi}^{\mathcal{I}}}(\mathbf{x}_{K_{\mathcal{I}}}^{\mathcal{I}})]$$

$$s.t.\ \mathbf{x}_{t+1}^i = \mathbf{f}^i(\mathbf{x}_t^i, \mathbf{u}_t^i) + \epsilon \boldsymbol{\sigma}^{\mathbf{f}_i}(t)\mathbf{w}_t^i,\ \forall i \in \mathcal{I}, \tag{3.6a}$$

$$\mathbf{z}_t^i = \mathbf{h}^i(\mathbf{x}_t^i) + \epsilon \boldsymbol{\sigma}^{\mathbf{h}_i}(t)\mathbf{v}_t^i,\ \forall i \in \mathcal{I}, \tag{3.6b}$$

*where the optimization is over continuously differentiable time-varying joint observation-trajectory-feedback policies, and:*

- $J_{\boldsymbol{\pi}^{\mathcal{I}}} : \Pi^{\mathcal{I}} \to \mathbb{R}$ *is the cost function, and* $J_{\boldsymbol{\pi}^{\mathcal{I}}} := \sum_{t=0}^{K_{\mathcal{I}}-1} c_t^{\boldsymbol{\pi}^{\mathcal{I}}}(\mathbf{x}_t^{\mathcal{I}}, \mathbf{u}_t^{\mathcal{I}}) + c_{K_{\mathcal{I}}}^{\boldsymbol{\pi}^{\mathcal{I}}}(\mathbf{x}_{K_{\mathcal{I}}}^{\mathcal{I}})$*;*
- $\boldsymbol{\pi}^{\mathcal{I}} \in \Pi^{\mathcal{I}}$ *defines the policy, where* $\boldsymbol{\pi}^{\mathcal{I}} := \{\boldsymbol{\pi}^1, \cdots, \boldsymbol{\pi}^m\}$*, and* $\boldsymbol{\pi}^i := \{\boldsymbol{\pi}_0^i, \cdots, \boldsymbol{\pi}_t^i\}$*;*
- $\boldsymbol{\pi}_t^i : (\mathbb{Z}^{\mathcal{I}})^t \to \mathbb{U}^i$ *specifies the optimal action for agent i:* $\mathbf{u}_t^i = \boldsymbol{\pi}_t^i(\mathbf{z}_{1:t}^{\mathcal{I}})$*;*
- $K_{\mathcal{I}} := \max_{i \in \mathcal{I}} K_i$*, and* $K_i > 0$ *is agent i's planning horizon;*
- $c_t^{\boldsymbol{\pi}^{\mathcal{I}}} : \mathbb{X}^{\mathcal{I}} \times \mathbb{U}^{\mathcal{I}} \to \mathbb{R}$ *is the incremental cost; and,*
- $c_{K_{\mathcal{I}}}^{\boldsymbol{\pi}^{\mathcal{I}}} : \mathbb{X}^{\mathcal{I}} \to \mathbb{R}$ *is the terminal cost.*

## 3.4 Main Results of Part I

The main results which lead up to a tractable, near-optimal decoupled solution of Problem (5) are the following. These results are each devoted a chapter in this part of the thesis.

**Result 1** *Consider a system with just a single agent which observes its state perfectly. Then a two step design approach, where first, the nominal trajectory of the system is designed and optimized taking into account the nonlinearities of the system but without any noise, and, second, the system equations are linearized around the nominal trajectory and a linear feedback policy is designed to track that nominal trajectory, is* $O(\epsilon^{2-\gamma})$*-optimal for* $0 < \gamma \ll 1$.

**Result 2** *Consider a system of* $m$ *agents, where each observes its state perfectly. Then a two-step design approach, where, first, the nominal trajectories of all the agents in the system are designed and optimized taking into account the nonlinearities of the system but without any noises, and, second, the system equation of each agent is linearized around its nominal trajectory and each agent applies an LQG-optimal feedback policy to track its own nominal trajectory, is* $O(\epsilon^{2-\gamma})$*-optimal for* $0 < \gamma \ll 1$. *The import of this result is that in the first step, the optimal nominal trajectory of the entire system is designed jointly incorporating the joint costs of the system, such as collision avoidance, etc.. Subsequently, the feedback policy of each agent is designed separately to track its own nominal trajectory using LQG optimal control. Thus, the centralized multi-agent problem can be tractably reduced near-optimally to a decentralized factored MDP, and then solved in a decoupled manner.*

**Result 3** *Consider a system with just one agent which imperfectly observes its own state. Then a two-step design approach, where, first, the nominal trajectory of the* *agent is designed and optimized taking into account the nonlinearity of its system but without any noise, and, second, its system equation is linearized around this nominal trajectory and the agent applies an LQG-optimal policy to track the nominal trajectory, is $O(\epsilon^{2-\gamma})$-optimal for $0 < \gamma \ll 1$. The second step is particularly simple since it is a simple LQG design.*

**Result 4** *Consider a system of $m$ agents, where each observes its state imperfectly in the presence of noise in the observations. Then a two-step design approach, where, first, the nominal trajectories of all the agents in the system are designed and optimized taking into account the nonlinearities of the system but without any noises, and, second, the system equation of each agent is linearized around its nominal trajectory and each agent applies an LQG-optimal policy to track its own nominal trajectory, is $O(\epsilon^{2-\gamma})$-optimal for $0 < \gamma \ll 1$. Thus, each agent can optimally implement a decentralized estimator without utilizing the belief-state information of the other agents. The resulting algorithm's computation is of a polynomial order in the state-dimension, number of agents and time-horizon. Thereby we have obtained a solution which is tractable, where linear feedbacks of the agents do not require knowledge of other agents' states, and which is nearly-optimal. The centralized multi-agent system with imperfect observations can so be reduced near-optimally to a decentralized LQG, and thereby solved near-optimally.*

# 4. FULLY-OBSERVED SINGLE-AGENT SYSTEM

In this chapter, we consider the single-agent fully-observed stochastic control problem. After a brief introduction, we first attempt to prove the near-first-order optimality of the considered policies. Then, in the last two sections we prove the near-second-order optimality of the proposed policies.

## 4.1 Introduction

Many robotic systems, in particular, mobile aerial and ground robots, are equipped with noisy actuators that require feedback compensation or planning ahead in a policy that accounts for the random perturbations. Simply ignoring the noise and planning for the unperturbed equivalent of the stochastic system can result in crucial errors leading to failure in reaching the end-goal, or result in the system falling into unsafe states. Moreover, the solution should not require a fully centralized control since that would require pervasive constant communication among all robots.

In a stochastic setting, the general problem of sequential decision-making can be formulated as a Markov Decision Problem (MDP) [2, 7]. The optimal solution of the stochastic control problem can be obtained iteratively by value or policy iteration methods to solve the Hamilton-Jacobi-Bellman equation [7]. Except in special cases, such as in a linear Gaussian environment, this involves discretization of the underlying spaces [8]; an approach whose scalability faces the curse of dimensionality [9]. As a result, the solutions require a computation time that is provably exponential in the state dimension, in a real number based model of complexity, without any assumption that $P \neq NP$ [10].

Many approaches have been proposed based on their tractability. Model Predictive Control (MPC)-based methods [27, 28], robust formulations [29, 30], and other designs that relate to the Pontryagin's Maximum Principle [31] are some of the methods that have been successfully used as surrogate design approaches. Another popular approach utilizes Differential Dynamic Programing (DDP) [32] and DDP-based variations such as the Stochastic DDP [33], iLQR and iLQG [34]–Stochastic DDP relies on second order approximation of the dynamics and cost, whereas iLQR and iLQG use second order approximation of the cost but first order linearization of the dynamics. These methods propose iterative methods that attempt to find "locally-optimal" solutions in a tube around a nominal trajectory [34] by coupling the design of feedback policy and the nominal trajectory of the system.

In this chapter, we address the nonlinear stochastic control problem and propose an architecture under which the decoupled design of an optimal open-loop control sequence and a decentralized feedback policy is both tractable and near-optimal. In particular, we show that under a small noise assumption, a decoupling into globally-optimal trajectory design and a decentralized feedback control law holds for fully-observed nonlinear stochastic systems of the type of interest in mobile robotic systems.

The design can be broken into two parts: *i)* an open-loop optimal control problem that designs the nominal trajectory of the LQR controller, which respects the nonlinearities as well as state and control constraints; *ii)* the design of a decentralized LQR policy around the optimized nominal trajectory. The quality of the design is rigorously provided by the main results of the chapter. We quantify the first and second order stochastic error for small-noise levels based on large deviations theory. We thereby arrive at what we call a Trajectory-optimized decoupled Linear Quadratic Regulator (T-LQR) design for fully-observed nonlinear stochastic systems under Gaussian small-noise perturbations.

The organization of the chapter is as follows. Section 4.2 states a simple large deviations result for linear Gaussian systems. Section 4.3 defines a general stochastic control problem for a fully-observed system. Section 6.1 analyzes the near-first-order optimality of the deterministic policy applied to the stochastic system under the assumption that the function are in $\mathbb{C}^1$. Section 4.5 proves the near-first-order optimality of the T-LQR policy under the assumption that the function are in $\mathbb{C}^1$. Section 4.6 analyzes a design based on T-LQR for a non-holonomic car-like robot and provides numerical results illustrating the proposed approach to design. Section 6.3 analyzes the near-second-order optimality of the deterministic policy applied to the stochastic system under the assumption that the function are in $\mathbb{C}^2$. Section 4.8 analyzes the near-first-order optimality of the T-LQR policy under the assumption that the function are in $\mathbb{C}^2$.

## 4.2 Small Random Perturbations of a Linear System

In this section, we consider the small noise perturbations of a linear Gaussian system. We state a simple Large Deviations probability for a linear Gaussian system. A general discussion regarding large deviations of the trajectories of a perturbed system from that of its unperturbed counterparts and related theories can be found in [6, 171, 172, 64, 173, 174, 175, 176, 177].

**Lemma 1 Large Deviations for Linear Gaussian System:** *Let*

$$\mathbf{x}_{t+1} = \mathbf{A}_t \mathbf{x}_t + \epsilon \boldsymbol{\sigma}_t \mathbf{w}_t,\ \ \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}}), \tag{4.1}$$

*where $\mathbf{x}_t \in \mathbb{X} \subset \mathbb{R}^{n_x}$, $\mathbf{x}_t = \mathbf{0}$, $\epsilon > 0$, and $\boldsymbol{\Sigma}_{\mathbf{w}}, \boldsymbol{\sigma}_t \succ 0$. Then, for each $\delta > 0$, and for some $\bar{\beta} > 0$ and $\bar{\gamma} > 0$,*

$$P(\max_{1 \le t \le K} \|\mathbf{x}_t\| > \delta) \le K n_x \bar{\beta} \frac{\epsilon}{\delta} \exp(-\bar{\gamma} \frac{\delta^2}{\epsilon^2}). \tag{4.2}$$

***Proof 1*** *First note that,*

$$\mathbf{x}_t = \epsilon \sum_{s=0}^{t-1} (\Pi_{r=s+1}^{t-1} \mathbf{A}_r) \boldsymbol{\sigma}_s \mathbf{w}_s =: \epsilon \sum_{s=0}^{t-1} \boldsymbol{\Phi}_{s,t} \mathbf{w}_s,$$

*where* $\boldsymbol{\Phi}_{s,t} := (\Pi_{r=s+1}^{t-1} \mathbf{A}_r) \boldsymbol{\sigma}_s, 0 \leq s \leq t-1, 2 \leq t \leq K,$ *and* $\boldsymbol{\Phi}_{0,1} = \boldsymbol{\sigma}_0$*. Now, if* $\mathbf{x}_t = (x_t^i), \mathbf{w}_t = (w_t^i), 1 \leq i \leq n_x$ *and* $\boldsymbol{\Phi}_{s,t} = (\Phi_{s,t}^{ij}), 1 \leq i, j \leq n_x$*, then*

$$x_t^i = \epsilon \sum_{s=0}^{t-1} \sum_{j=1}^{n_x} \Phi_{s,t}^{ij} w_s^i \sim \mathcal{N}(0, \epsilon^2 \alpha_{i,t}),$$

*where* $\alpha_{i,t} := \sum_{s=0}^{t-1} \sum_{j=1}^{n_x} (\Phi_{s,t}^{ij})^2, 1 \leq i \leq n_x, 1 \leq t \leq K$*, whence* $\alpha_{i,t} > 0$*. Now, let* $z \sim \mathcal{N}(0,1)$ *be a standard normal random variable. Then, for* $0 < \delta \leq u$*, we have* $1 \leq u/\delta$*, and the tail probability of z is [178]:*

$$\begin{aligned} P(z > \delta) &= \frac{1}{\sqrt{2\pi}} \int_\delta^\infty \exp(-\frac{u^2}{2}) du \\ &\leq \frac{1}{\sqrt{2\pi}} \int_\delta^\infty \frac{u}{\delta} \exp(-\frac{u^2}{2}) du \leq \frac{1}{\delta\sqrt{2\pi}} \exp(-\frac{\delta^2}{2}). \end{aligned}$$

*Hence, we have*

$$P(z^2 > \delta^2) = P(z > \delta) + P(z < -\delta) \leq \frac{2}{\delta\sqrt{2\pi}} \exp(-\frac{\delta^2}{2}).$$

*So,*

$$\begin{aligned} P((x_t^i)^2 > \delta^2) &= P((\frac{x_t^i}{\epsilon\alpha_{i,t}})^2 > \frac{\delta^2}{\epsilon^2 \alpha_{i,t}^2}) \\ &\leq \frac{\epsilon\alpha_{i,t}}{\delta} \sqrt{\frac{2}{\pi}} \exp(-\frac{\delta^2}{2\epsilon^2 \alpha_{i,t}^2}). \end{aligned}$$

*Now, let* $\bar{\beta} := \sqrt{\frac{2}{\pi}}(\max_{1 \leq i \leq n_x, 1 \leq t \leq K} \alpha_{i,t})$ *and* $\bar{\gamma} := 1/(\bar{\beta}^2 \pi)$*, whence* $\bar{\beta}, \bar{\gamma} > 0$*. Then,*

$$P((x_t^i)^2 > \delta^2) \leq \bar{\beta}\frac{\epsilon}{\delta}\exp(-\bar{\gamma}\frac{\delta^2}{\epsilon^2}),$$

*Hence,*

$$\begin{aligned} P(\max_{1 \leq t \leq K} \|\mathbf{x}_t\| > \delta) &\leq \sum_{t=1}^{K} P(\|\mathbf{x}_t\| > \delta) \\ &= \sum_{t=1}^{K} P(\|\mathbf{x}_t\|^2 > \delta^2) = \sum_{t=1}^{K} P(\sum_{i=1}^{n_x}(x_t^i)^2 > \delta^2) \\ &\leq \sum_{t=1}^{K}\sum_{i=1}^{n_x} P((x_t^i)^2 > \delta^2) \leq \sum_{t=1}^{K}\sum_{i=1}^{n_x} \bar{\beta}\frac{\epsilon}{\delta}\exp(-\bar{\gamma}\frac{\delta^2}{\epsilon^2}) \\ &= K n_x \bar{\beta}\frac{\epsilon}{\delta}\exp(-\bar{\gamma}\frac{\delta^2}{\epsilon^2}). \end{aligned}$$

*Remark:* Note that using the above lemma, for a fixed $\delta > 0$ we have:

$$P(\max_{1 \leq t \leq K} \|\mathbf{x}_t\| > \delta) = o(\exp(-\frac{1}{\epsilon^2})), \tag{4.3}$$

which tends to zero much faster than $o(\epsilon)$, as $\epsilon \downarrow 0$. Thus, for a fixed $\delta$, the probability that the trajectory of $\mathbf{x}$ ever exits the tube of radius $\delta$ around the nominal zero trajectory in the time interval $[0, t]$ goes to zero exponentially.

*Remark:* Note that in the above lemma, $\bar{\beta} = 1/\sqrt{\bar{\gamma}\pi}$, and the lemma can be rewritten with only one constant, where

$$\bar{\beta} = \sqrt{\frac{2}{\pi}}(\max_{1 \leq i \leq n_x, 1 \leq t \leq K} \sum_{s=0}^{t-1}\sum_{j=1}^{n_x}(\Phi_{s,t}^{ij})^2)$$

where $\mathbf{\Phi}_{s,t} = (\Pi_{r=s+1}^{t-1}\mathbf{A}_r)\boldsymbol{\sigma}_s, 0 \leq s \leq t-1, 2 \leq t \leq K$, and $\mathbf{\Phi}_{0,1} = \boldsymbol{\sigma}_0$. This means that $\bar{\beta}$ is proportional to the aggregated effect of the noise (or the variance of the trajectory's perturbation) in a direction which it is highest. In fact, using the large deviations theory one can find the first exit probability, as well as the most probable exit path.

*Remark:* This probability also linearly increases with the time horizon, $K$, and the dimension of the state $n_x$.

*Remark:* Let us provide a simple example and compute the above probability. Let $\mathbf{x} \in \mathbb{R}^{n_x}$, $\mathbf{x}_0 = \mathbf{0}$, and

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \epsilon \mathbf{w}_t,\ \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Then, $\mathbf{\Phi}_{s,t} = \mathbf{I}, 0 \leq s \leq t-1, 1 \leq t \leq K$ and

$$\bar{\beta} = \sqrt{\frac{2}{\pi}}(\max_{1 \leq i \leq n_x, 1 \leq t \leq K} \sum_{s=0}^{t-1} 1) = \sqrt{\frac{2}{\pi}}(\max_{1 \leq t \leq K} t) = \sqrt{\frac{2}{\pi}} K.$$

Therefore,

$$P(\max_{1 \leq t \leq K} \|\mathbf{x}_t\| > \delta) \leq \sqrt{\frac{2}{\pi}} K^2 n_x \frac{\epsilon}{\delta} \exp(-\frac{1}{2K^2}\frac{\delta^2}{\epsilon^2}).$$

Now, let us fix $\delta = 1, n_x = 2$, and $K = 10$. Then, the right hand side probability becomes $\sqrt{\frac{2}{\pi}} 200\epsilon \exp(-\frac{1}{200\epsilon^2})$, which equals 0.28 for $\epsilon = 0.04$. Therefore, the probability of staying in the 1-meter tube around zero after 10 steps is at least 0.72. For any higher $\epsilon$, this probability will be out of a reasonable tolerance range. In the next sections, for a car-like robot model we numerically show that this probability improves significantly using feedback, and show that higher levels of noise also can be tolerated with high probability.

*Remark:* Note that although we provided an example for a fixed $\delta$, in fact for our proofs we will use an $\epsilon$-dependent definition of $\delta$ such that as $\epsilon \downarrow 0$, $\delta \downarrow 0$, as well.

This is mainly because, we will prove that the errors of our proposed policies are dependent on $\delta$. Hence, a fixed $\delta$ (independent from $\epsilon$) does not provide our desired characteristics. We also will show that for such a choice of $\delta$, the above probability is not anymore exponential in $\epsilon$, rather it is polynomial.

We will use the analysis of this section to analyze the optimality of our design in the next section.

## 4.3 The Fully-Observed System

The general stochastic control problem of interest for a fully-observed system can be formulated as an optimization problem in the space of feedback policies. Without loss of generality, we consider discrete-time systems.

*Process model:* We denote the state and control by $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^{n_x}$ and $\mathbf{u} \in \mathbb{U} \subset \mathbb{R}^{n_u}$, respectively. Given $\mathbf{x}_0 \in \mathbb{X}$, the process model with $\mathbf{f} : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$ is defined as:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \epsilon \boldsymbol{\sigma}_t \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}_t}) \tag{4.4}$$

where $\{\mathbf{w}_t\}$ is independent, identically distributed (i.i.d.).

Now, we pose the general stochastic control problem [2, 12]. We restrict attention to continuously differentiable policies throughout this section. We will also need to assume that there exists an optimal policy in this class.

**Problem 6 Stochastic Control Problem for Fully-Observed System***: Given an initial state* $\mathbf{x}_0$*, we wish to determine an optimal or near-optimal policy for*

$$\begin{aligned} \min_{\boldsymbol{\pi}} \ \mathbb{E}[\sum_{t=0}^{K-1} c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t) + c_K^{\boldsymbol{\pi}}(\mathbf{x}_K)] \\ s.t. \ \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \epsilon \boldsymbol{\sigma}_t \mathbf{w}_t, \end{aligned} \tag{4.5}$$

*where the optimization is over continuously differentiable Markov, i.e., time-varying state-feedback policies,* $\boldsymbol{\pi} \in \Pi$, *and*

- $\boldsymbol{\pi} := \{\boldsymbol{\pi}_0, \cdots, \boldsymbol{\pi}_t\}$, $\boldsymbol{\pi}_t : \mathbb{X} \to \mathbb{U}$, *and* $\mathbf{u}_t = \boldsymbol{\pi}_t(\mathbf{x}_t)$ *specifies the action taken given the state;*
- $c_t^{\pi}(\cdot, \cdot) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ *is the one-step cost function;*
- $c_K^{\pi}(\cdot) : \mathbb{X} \to \mathbb{R}$ *denotes the terminal cost;*
- $K > 0$ *is the time horizon; and*
- *We also assume that the cost function is continuously differentiable and bounded. That is* $|c_t| \leq M$ *and* $|c_K| \leq M$ *for some* $M > 0$.

*Assumption:* For the analysis of Sections 4.7 and 4.8, we will add the assumption that all the functions are in $\mathbb{C}^2$, where $\mathbb{C}^r, r \geq 1$ denotes the space of continuous functions that are differentiable to the $r$-th order and their derivatives are also continuous up to the $r$-th order. However, for the analysis of Sections 4.4 and 4.5 we only assume that the functions are in $\mathbb{C}^1$.

## 4.4 Case I: The Deterministic Optimal Policy

In this section, we analyze the performance of the deterministic optimal control policy used in the stochastic problem.

**Problem 7 Deterministic Closed-Loop Problem**: *Given an initial state* $\mathbf{x}_0$, *we begin by determining a continuously differentiable optimal feedback policy for*

$$\min_{\boldsymbol{\pi}} \sum_{t=0}^{K-1} c_t(\mathbf{x}_t, \mathbf{u}_t) + c_K(\mathbf{x}_K)$$

$$s.t.\ \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t). \tag{4.6}$$

*Nominal trajectories:* For $0 \leq t \leq K-1$, let $\boldsymbol{\pi}^d$ be the optimal feedback law of the deterministic problem above, and let $\mathbf{x}_t^p$ be the corresponding state, where

$$\mathbf{u}_t^p := \boldsymbol{\pi}_t^d(\mathbf{x}_t^p), \ \mathbf{x}_{t+1}^p := \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p), \tag{4.7}$$

where $\mathbf{x}_0^p := \mathbf{x}_0$. We refer to this as the nominal trajectories.

*Linearization of the system equations:* We consider the application of a control $\mathbf{u}_t = \boldsymbol{\pi}_t^d(\mathbf{x}_t)$ to the stochastic system. Then the resulting trajectory is:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \boldsymbol{\pi}_t^d(\mathbf{x}_t)) + \epsilon\boldsymbol{\sigma}_t\mathbf{w}_t. \tag{4.8}$$

Let $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}_t^p$ denote the state error. Then we linearize the drift of the process model around the nominal trajectory. Hence, for $0 \leq t \leq K-1$:

$$\begin{aligned}
\tilde{\mathbf{x}}_{t+1} &= \mathbf{f}(\mathbf{x}_t, \boldsymbol{\pi}_t^d(\mathbf{x}_t)) - \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p) + \epsilon\boldsymbol{\sigma}_t\mathbf{w}_t && \text{(4.9a)}\\
&= \mathbf{A}_t\tilde{\mathbf{x}}_t - \mathbf{B}_t\mathbf{L}_t\tilde{\mathbf{x}}_t + \epsilon\boldsymbol{\sigma}_t\mathbf{w}_t + o(\|\tilde{\mathbf{x}}_t\|) && \text{(4.9b)}\\
&=: \mathbf{D}_t\tilde{\mathbf{x}}_t + \mathbf{G}_t\mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty), && \text{(4.9c)}
\end{aligned}$$

as $(\|\tilde{\mathbf{x}}\|_\infty) \downarrow 0$, where we have:

- $\mathbf{A}_t := \nabla_\mathbf{x}\mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{B}_t := \nabla_\mathbf{u}\mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{L}_t := -\nabla_\mathbf{x}\boldsymbol{\pi}_t^d(\mathbf{x})|_{\mathbf{x}_t^p}$, $\mathbf{G}_t := \epsilon\boldsymbol{\sigma}_t$;
- $\mathbf{D}_t := \mathbf{A}_t - \mathbf{B}_t\mathbf{L}_t, 1 \leq t \leq K-1$, $\mathbf{D}_0 = \mathbf{G}_0$; and
- $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}_0^p = \mathbf{0}$.

*The exactly linear l-system:* From the above system of (4.25), we remove the $o(\cdot)$ terms, and define an exactly linear system:

$$\tilde{\mathbf{x}}_{t+1}^l := \mathbf{D}_t\tilde{\mathbf{x}}_t^l + \mathbf{G}_t\mathbf{w}_t, \tag{4.10}$$

where $\tilde{\mathbf{x}}_0^l := \tilde{\mathbf{x}}_0 = \mathbf{0}$.

*The difference d-system:* We denote the difference between the two systems of (4.9c) and (4.10) by a superscript $d$, and define for $0 \leq t \leq K-1$, $\tilde{\mathbf{x}}_{t+1}^d := \tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_{t+1}^l$, where $\tilde{\mathbf{x}}_0^d = \tilde{\mathbf{x}}_0 - \tilde{\mathbf{x}}_0^l = \mathbf{0}$. Therefore,

$$\tilde{\mathbf{x}}_{t+1}^d = \mathbf{D}_t \tilde{\mathbf{x}}_t^d + o(\|\tilde{\mathbf{x}}\|_\infty) = \tilde{\mathbf{D}}_{0:t} \tilde{\mathbf{x}}_0^d + o(\|\tilde{\mathbf{x}}\|_\infty) = o(\|\tilde{\mathbf{x}}\|_\infty) \tag{4.11}$$

where $\tilde{\mathbf{D}}_{t_1:t_2} = \Pi_{t=t_1}^{t_2} \mathbf{D}_t, t_2 \geq t_1 \geq 0$, otherwise, it is the identity matrix. This leads to $o(\|\tilde{\mathbf{x}}^d\|_\infty) = o(\|\tilde{\mathbf{x}}\|_\infty)$. Hence,

$$O(\|\tilde{\mathbf{x}}^l\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty) + o(\|\tilde{\mathbf{x}}\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty). \tag{4.12}$$

This means that all the errors in the original system, the $l$-system, and the $d$-system are of the order of $O(\|\tilde{\mathbf{x}}\|_\infty)$. Moreover, $O(\|\tilde{\mathbf{x}}\|_\infty)$ is itself $O(\|\tilde{\mathbf{x}}^l\|_\infty)$, which we calculate next.

*Large deviations:* The $l$-system is a linear Gaussian system with additive noise, for which we use the large deviations result of Lemma 1 modifying the definition of $\mathbf{\Phi}_{s,t}$ for $0 \leq s \leq t-1, 2 \leq t \leq K$ as $\mathbf{\Phi}_{s,t} := (\Pi_{r=s+1}^{t-1} \mathbf{D}_r) \boldsymbol{\sigma}_s, 0 \leq s \leq t-1, 2 \leq t \leq K$. Thus, for each finite $\delta \geq 0$, we have $P\{\max_{0 \leq t \leq K} \|\tilde{\mathbf{x}}_t^l\| \geq \delta\} = o(\epsilon)$.

Let $\Omega(\delta)$ be the set where $\max_{0 \leq t \leq K} \|\tilde{\mathbf{x}}_t^l\| \leq \delta$. Then, $P(\Omega(\delta)) \geq 1 - o(\epsilon)$ and for $\omega \in \Omega(\delta)$, $\|\tilde{\mathbf{x}}^l\|_\infty = O(\delta)$. Therefore, from the calculations above, we have that $O(\|\tilde{\mathbf{x}}\|_\infty) = O(\delta)$, and hence all the other errors are also $O(\delta)$ for $\omega \in \Omega(\delta)$.

Then for $\omega \in \Omega(\delta)$ and for all $0 \leq t \leq K-1$,

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^p + \tilde{\mathbf{x}}_{t+1}^l + O(\delta), \tag{4.13}$$

which means that the linear Gaussian stochastic $\tilde{(\cdot)}^l$-system along with the deterministic $p$-system can be used to control the original system given the $O(\delta)$ approximations hold (with probability of at least $1 - o(\epsilon)$). In another interpretation, the original system can be approximated for all $0 \leq t \leq K - 1$ as:

$$\mathbf{x}_{t+1} = \mathbf{x}^l_{t+1} + O(\delta). \tag{4.14}$$

*Remark:* Note that choosing different open-loop policies other than the optimal, results in a different feedback gain $\mathbf{L}_t$, and therefore, a different transfer function for the system. Particularly, the $\mathbf{\Phi}$ function defined in Lemma 1 changes and the pre-constant and the exponent's constant in the large deviations probability changes, as well.

*Linear iterative equation:* Before proceeding to the next lemma, let us first solve a general linear iterative formula. Let

$$\mathbf{x}_{t+1} = \mathbf{D}_t \mathbf{x}_t + \mathbf{f}_t, \tag{4.15}$$

for some given $\mathbf{f}_s, 0 \leq s \leq t$ and $\mathbf{x}_0$. Then,

$$\begin{aligned}
\mathbf{x}_{t+1} =& \mathbf{D}_t \mathbf{x}_t + \mathbf{f}_t = \mathbf{D}_t(\mathbf{D}_{t-1}\mathbf{x}_{t-1} + \mathbf{f}_{t-1}) + \mathbf{f}_t = \mathbf{D}_t\mathbf{D}_{t-1}\mathbf{x}_{t-1} + \mathbf{D}_t\mathbf{f}_{t-1} + \mathbf{f}_t \\
=& \mathbf{D}_t\mathbf{D}_{t-1} \times \cdots \times \mathbf{D}_{t-t}\mathbf{x}_{t-t} + \sum_{s=0}^{t} (\mathbf{D}_t\mathbf{D}_{t-1} \times \cdots \times \mathbf{D}_{t-s+1})\mathbf{f}_{t-s} \\
=& \tilde{\mathbf{D}}_{0:t}\mathbf{x}_0 + \sum_{s=0}^{t} \tilde{\mathbf{D}}_{t-s+1:t}\mathbf{f}_{t-s} \\
=& \tilde{\mathbf{D}}_{0:t}\mathbf{x}_0 + \sum_{r=0}^{t} \tilde{\mathbf{D}}_{r+1:t}\mathbf{f}_r,
\end{aligned}$$

where we used $r = t - s$ in the last equation. Note that this formula can be easily verified using mathematical induction.

**Lemma 2 State Error Propagation:** *For the l-system of* (4.10)*, the state error* $\tilde{\mathbf{x}}^l_{t+1}$ *can be written as:*

$$\tilde{\mathbf{x}}^l_{t+1} = \sum_{s=0}^{t} \tilde{\mathbf{D}}^{\mathbf{w}}_{s,t}\mathbf{w}_s, \ 0 \leq t \leq K-1, \tag{4.16}$$

*where we have:*

- $\tilde{\mathbf{D}}^{\mathbf{w}}_{s,t} := \tilde{\mathbf{D}}_{s+1:t}\mathbf{G}_s, 0 \leq s \leq t-1, t \geq 1$*; and*
- $\tilde{\mathbf{D}}^{\mathbf{w}}_{t,t} := \tilde{\mathbf{D}}_{t+1:t}\mathbf{G}_t = \mathbf{G}_t, t \geq 0.$

***Proof 2*** *Given* $\tilde{\mathbf{x}}^l_0 = \mathbf{0}$*, we have:*

$$\tilde{\mathbf{x}}^l_{t+1} = \mathbf{D}_t\tilde{\mathbf{x}}^l_t + \mathbf{G}_t\mathbf{w}_t = \tilde{\mathbf{D}}_{0:t}\tilde{\mathbf{x}}^l_0 + \sum_{r=0}^{t} \tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r =: \sum_{s=0}^{t} \tilde{\mathbf{D}}^{\mathbf{w}}_{s,t}\mathbf{w}_s.$$

Next, we linearize the cost function and provide the near-first-order optimality of this design.

*Linearization of the cost function:* We similarly linearize the cost function around the nominal trajectories of state and control actions:

$$J = J^p + \tilde{J}_1 + o(\sum_{t=1}^{K} \|\tilde{\mathbf{x}}_t\|) \tag{4.17a}$$

$$= J^p + \tilde{J}_1 + o(\|\tilde{\mathbf{x}}\|_\infty), \tag{4.17b}$$

where we have:

- $J^p := \sum_{t=0}^{K-1} c_t(\mathbf{x}^p_t, \mathbf{u}^p_t) + c_K(\mathbf{x}^p_K)$ denotes the nominal cost;
- $\tilde{J}_1 := \sum_{t=0}^{K-1}(\mathbf{C}^{\mathbf{x}}_t\tilde{\mathbf{x}}_t - \mathbf{C}^{\mathbf{u}}_t\mathbf{L}_t\tilde{\mathbf{x}}_t) + \mathbf{C}^{\mathbf{x}}_K\tilde{\mathbf{x}}_K$ is the first order cost error;

- $J_1 := J^p + \tilde{J}_1$ is the first order approximation of the cost;
- and $\mathbf{C}_t^{\mathbf{x}} = \nabla_{\mathbf{x}} c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{u}} = \nabla_{\mathbf{u}} c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_K^{\mathbf{x}} = \nabla_{\mathbf{x}} c_K(\mathbf{x})|_{\mathbf{x}_K^p}$.

Therefore, for $\omega \in \Omega(\delta)$, and

$$
\begin{aligned}
J &= J^p + \sum_{t=0}^{K-1} (\mathbf{C}_t^{\mathbf{x}} - \mathbf{C}_t^{\mathbf{u}} \mathbf{L}_t) \tilde{\mathbf{x}}_t + \mathbf{C}_K^{\mathbf{x}} \tilde{\mathbf{x}}_K + O(\delta) && \text{(4.18a)} \\
&= J^p + \sum_{t=0}^{K-1} (\mathbf{C}_t^{\mathbf{x}} - \mathbf{C}_t^{\mathbf{u}} \mathbf{L}_t) \tilde{\mathbf{x}}_t^l + \mathbf{C}_K^{\mathbf{x}} \tilde{\mathbf{x}}_K^l + O(\delta). && \text{(4.18b)}
\end{aligned}
$$

Hence, $J - J_1 = O(\delta)$ for $\omega \in \Omega(\delta)$.

Next, we provide the main result regarding the expected first order error of the cost function.

**Theorem 2 First-Order Cost Function Error for a Fully-Observed System Using a Deterministic Policy:** *Given that process noises are zero mean i.i.d. Gaussian, under a first-order approximation for the small noise paradigm, the stochastic cost function is dominated by the nominal part of the cost function, and the expected first-order error is $O(\delta)$. That is,*

$$\mathbb{E}[\tilde{J}_1] = O(\delta), \ \text{and } \mathbb{E}[J] = J^p + O(\delta).$$

*Moreover, by choosing $\delta = \sqrt{\log(\frac{1}{\epsilon})}\epsilon$, we have*

$$\mathbb{E}[\tilde{J}_1] = O(\epsilon^{1-\gamma}), \ \text{and } \mathbb{E}[J] = J^p + O(\epsilon^{1-\gamma}),$$

*for some $0 < \gamma \ll 1$, which shows that this error tends to zero with a near-first-order rate as $\epsilon \downarrow 0$.*

***Proof 3*** *Let $\tilde{J}_1^l := \sum_{t=0}^{K-1} (\mathbf{C}_t^{\mathbf{x}} - \mathbf{C}_t^{\mathbf{u}} \mathbf{L}_t) \tilde{\mathbf{x}}_t^l + \mathbf{C}_K^{\mathbf{x}} \tilde{\mathbf{x}}_K^l$. Also note $\tilde{\mathbf{x}}_0 = \mathbf{0}$, and $\mathbb{E}[\mathbf{w}_t] = 0$*

*for all $t$. Then, we use Lemmas 3 and 4:*

$$\begin{aligned}\mathbb{E}[\tilde{J}_1^l] &= \sum_{t=0}^{K-1}((\mathbf{C}_t^{\mathbf{x}} - \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_t)\mathbb{E}[\tilde{\mathbf{x}}_t^l]) + \mathbf{C}_K^{\mathbf{x}}\mathbb{E}[\tilde{\mathbf{x}}_K^l] \\ &= \sum_{t=0}^{K-1}((\mathbf{C}_t^{\mathbf{x}} - \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_t)\mathbb{E}[\sum_{s=0}^{t-1}\tilde{\mathbf{D}}_{s,t-1}^{\mathbf{w}}\mathbf{w}_s]) + \mathbf{C}_K^{\mathbf{x}}\mathbb{E}[\sum_{s=0}^{K-1}\tilde{\mathbf{D}}_{s,K-1}^{\mathbf{w}}\mathbf{w}_s] \\ &= \sum_{t=0}^{K-1}((\mathbf{C}_t^{\mathbf{x}} - \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_t)\sum_{s=0}^{t-1}\tilde{\mathbf{D}}_{s,t-1}^{\mathbf{w}}\mathbb{E}[\mathbf{w}_s]) + \mathbf{C}_K^{\mathbf{x}}\sum_{s=0}^{K-1}\tilde{\mathbf{D}}_{s,K-1}^{\mathbf{w}}\mathbb{E}[\mathbf{w}_s] = 0.\end{aligned}$$

***The probabilistic argument and choosing the proper $\delta$:*** *Now, we take expectation from both sides of (4.38b). Since, for $\omega \notin \Omega(\delta)$, $J \leq M$, then*

$$\begin{aligned}\mathbb{E}[J - J^p] &= P(\Omega(\delta))(\mathbb{E}[\tilde{J}_1^l] + O(\delta)) + M(1 - P(\Omega(\delta))) \\ &= P(\Omega(\delta))O(\delta) + M(1 - P(\Omega(\delta))) \qquad (4.19)\end{aligned}$$

*As mentioned before, $P(\Omega(\delta)) \geq 1 - Kn_x\bar{\beta}\frac{\epsilon}{\delta}\exp(-\bar{\gamma}\frac{\delta^2}{\epsilon^2})$. Since, we are only interested in the order of the above expectation, then, we will calculate the $O(P(\Omega(\delta))O(\delta) + M(1 - P(\Omega(\delta))))$. Therefore, for the purpose of calculations, we ignore the inequality and also the $O(\cdot)$ notation. As it is noticed, the above expectation depends on both delta and epsilon. Therefore, a proper choice of $\delta$ is required in order to make the expression in terms of one of the parameters (particularly, $\epsilon$). Without loss of generality let $\delta := k(\epsilon)\epsilon$, where $k : \mathbb{R}^+ \to [1, \infty)$ is a function of $\epsilon$. Therefore,*

$$\begin{aligned}P(\Omega(\delta))\delta + M(1 - P(\Omega(\delta))) &= (1 - Kn_x\bar{\beta}\frac{\epsilon}{\delta}\exp(-\bar{\gamma}\frac{\delta^2}{\epsilon^2}))\delta + MKn_x\bar{\beta}\frac{\epsilon}{\delta}\exp(-\bar{\gamma}\frac{\delta^2}{\epsilon^2}) \\ &= k(\epsilon)\epsilon - Kn_x\bar{\beta}\epsilon\exp(-\bar{\gamma}k^2(\epsilon)) + MKn_x\bar{\beta}\frac{\exp(-\bar{\gamma}k^2(\epsilon))}{k(\epsilon)}. \qquad (4.20)\end{aligned}$$

*Now, since in this section we are only interested in proving the near-first-order op-*

*timality of the provided policy, let us choose the value of* $\frac{\exp(-\bar{\gamma}k^2(\epsilon))}{k(\epsilon)} = O(\epsilon)$*. As a result, the second term in* (4.19) *becomes* $O(\epsilon)$*, and after determining the function* $k(\cdot)$*, we test if the order of the first term is also* $O(\epsilon)$*. Now, since in* $\frac{1}{k(\epsilon)\exp(\bar{\gamma}k^2(\epsilon))}$*, the exponential term finally dominates, we choose* $\exp(-\bar{\gamma}k^2(\epsilon)) = \epsilon$ *or by ignoring the constant term,* $k(\epsilon) = \sqrt{-\log(\epsilon)}$*, where the* log *denotes the natural logarithm. Therefore, we choose* $\delta := \sqrt{-\log(\epsilon)}\epsilon$*. Now, let us verify that all the three terms in* (4.20) *are* $O(\epsilon^{1-\gamma})$*. The calculations for the first term are:*

$$\lim_{\epsilon\downarrow 0}\frac{\delta}{\epsilon^{1-\gamma}} = \lim_{\epsilon\downarrow 0}\frac{k(\epsilon)\epsilon}{\epsilon^{1-\gamma}} = \lim_{\epsilon\downarrow 0}\frac{\sqrt{-\log(\epsilon)}\epsilon}{\epsilon^{1-\gamma}} = \lim_{\epsilon\downarrow 0}\frac{\sqrt{-\log(\epsilon)}}{\epsilon^{-\gamma}} = \lim_{\epsilon\downarrow 0}\frac{(-\log(\epsilon))^{-0.5}(-0.5)\epsilon^{-1}}{-\gamma\epsilon^{-\gamma-1}}$$
$$= \lim_{\epsilon\downarrow 0}\frac{0.5(-\log(\epsilon))^{-0.5}}{\gamma\epsilon^{-\gamma}} = \lim_{\epsilon\downarrow 0}\frac{0.5\epsilon^{\gamma}}{\gamma(-\log(\epsilon))^{0.5}} = 0,$$

*where we used the L'Hospital's rule. Hence,* $\delta = o(\epsilon^{1-\gamma})$*. However, for the sake of this proof, since we want* $O(\delta)$*, we will use* $O(\delta) = O(\epsilon^{1-\gamma})$*. The calculations for the third term are as follows (we ignore the constants in front of the fraction and exponent):*

$$\lim_{\epsilon\downarrow 0}\frac{\exp(-k^2(\epsilon))}{k(\epsilon)\epsilon^{1-\gamma}} = \lim_{\epsilon\downarrow 0}\frac{\exp(\log(\epsilon))}{\sqrt{-\log(\epsilon)}\epsilon^{1-\gamma}} = \lim_{\epsilon\downarrow 0}\frac{\epsilon}{\sqrt{-\log(\epsilon)}\epsilon^{1-\gamma}} = \lim_{\epsilon\downarrow 0}\frac{\epsilon^{\gamma}}{\sqrt{-\log(\epsilon)}} = 0.$$

*Therefore, the third term is also at least* $O(\epsilon^{1-\gamma})$*. In fact, this term is* $o(\epsilon)$ *(verified by setting* $\gamma$ *to zero); however, since, the bottle neck is the first term, we can just replace it with* $O(\epsilon^{1-\gamma})$*. The second term consists of the third term times the first term (ignoring the constants). Therefore, this term also is at least* $O(\epsilon^{1-\gamma})$*. As a result, we have* $\mathbb{E}[J] = J^p + O(\epsilon^{1-\gamma})$ *and the other statements hold, as well.*

*Remark:* Note that using this choice of $\delta$, the probability $1 - P(\Omega(\delta))$ is

$$1 - P(\Omega(\delta)) \leq K n_x \bar{\beta} \frac{\exp(-\bar{\gamma} k^2(\epsilon))}{k(\epsilon)} = K n_x \bar{\beta} \exp(-\bar{\gamma}) \epsilon (-\log(\epsilon))^{-\frac{1}{2}}, \tag{4.21}$$

which we proved that it decreases to zero with at least $o(\epsilon)$ rate as $\epsilon \downarrow 0$. However, since we have taken the expectation in (4.19), this probability does not have an independent meaning. That is, although the linearizations are valid only with probability $P(\Omega(\delta))$, the expectation in (4.19) incorporates that and uses the fact that the cost is bounded to calculate the overall cost performance of the design. In the next Corollary, we address the relations between this design and an optimal policy.

*Remark:* The chosen value for $\delta$ guaranteers that $\delta > \epsilon$. Since $\delta = o(\epsilon^{1-\gamma})$ as $\epsilon \downarrow 0$. In fact, $\delta/\epsilon = k(\epsilon) = \sqrt{-\log(\epsilon)} > 1$ for $0 < \epsilon < e^{-1} \simeq 0.368$ where $e \simeq 2.71828$ is Euler's number (aka Napier's constant). In a word, the tube size is bigger than the value of $\epsilon$ for $\epsilon$ less than 36 percent which is a very large noise. As an example, for $\epsilon = 0.1$, $\delta = 0.1517$, whereas for $\epsilon = 0.01$, $\delta = 0.0215$.

*Remark:* Note that using (4.20) with any fixed choice of $\delta > 0$ and letting $\epsilon$ be small enough, the error in the cost becomes $O(\delta)$, and even if we decrease $\epsilon$ further, the error will not decrease much from $O(\delta)$. However, using the proper choice of $\delta$ such as $\delta = \sqrt{-\log(\epsilon)}\epsilon$ means that the error will always decrease (to zero) as $\epsilon \downarrow 0$.

*Remark:* Note that designing an optimized feedback changes the constants $\bar{\beta}$ and $\bar{\gamma}$, hence optimizes the probability (4.21)'s pre-constants (rather than order) as well as the cost error's pre-constants in (4.19). This in fact is valuable and helps the proposed algorithms to tolerate moderate levels of noise as well with a proper optimized feedback. The T-LQR framework that is proposed in the next section provides an example such an optimized policy, for which we will show that the order of errors are the same as the design of this section.

*Remark:* Using this result, we can prove that under small noise for a fully-observed system, the deterministic policy is near-first-order optimal when the functions are in $\mathbb{C}^1$, which is summarized next. However, later in Section 4.7 we add the assumption that the functions are in $\mathbb{C}^2$ and expand the equations to the second-order. As a result we improve this result and prove that for the same design approach the cost function error is in fact near-second-order in $\epsilon$, i.e., we will show that $\mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma})$. Therefore, we will prove that the policy is also near-second-order optimal. Nevertheless, the calculations of this section provides a valuable insight on the probabilistic arguments, which we use in that section, as well.

**Corollary 1 Near-First-Order Optimality of the Deterministic Optimal Policy for a Stochastic Fully-Observed System Under Small Noise.** *Based on Theorem 2, for a fully-observed system where the function are in $\mathbb{C}^1$ under the small noise paradigm, as $\epsilon \downarrow 0$, the deterministic optimal control law becomes $O(\epsilon^{1-\gamma})$-optimal with $0 < \gamma \ll 1$ for the stochastic problem.*

***Proof 4*** *Using Theorem 2, for $\omega \in \Omega(\delta)$ we have $\mathbb{E}[J] = J^p + O(\epsilon^{1-\gamma})$, which is the cost of applying policy $\boldsymbol{\pi}^d$ to the stochastic system. Now, suppose $\boldsymbol{\pi}^*$ is the optimal stochastic policy. By assumption $\boldsymbol{\pi}^*$ is continuously differentiable. Therefore, by modifying the definition of $\mathbf{L}_t$ as $\mathbf{L}_t = -\nabla_{\mathbf{x}} \boldsymbol{\pi}^*_t(\mathbf{x})|_{\mathbf{x}^{*p}_t}$, defining $\mathbf{u}^{*p}_t = \boldsymbol{\pi}^*_t(\mathbf{x}^{*p}_t)$ and replacing $p$ with $*p$ in (4.7), we have $\boldsymbol{\pi}^*_t(\mathbf{x}_t) = \mathbf{u}^{*p}_t - \mathbf{L}_t(\mathbf{x}_t - \mathbf{x}^{*p}_t) + o(\|\tilde{\mathbf{x}}_t\|)$. Similarly, by using appropriate modifications, the entire calculations of this section hold for this policy, as well. Hence, using Theorem 2 for this system, the cost function of policy $\boldsymbol{\pi}^*$ can be written as $\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{1-\gamma})$, where $J^{*p}$ is defined similarly as $J^p$ as well. Now, by construction $J^p \leq J^{*p}$, and*

$$\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{1-\gamma}) \geq J^p + O(\epsilon^{1-\gamma}) = \mathbb{E}[J_{\boldsymbol{\pi}^d}] + O(\epsilon^{1-\gamma})$$

*As a result, policy $\boldsymbol{\pi}^d$ is within $O(\epsilon^{1-\gamma})$ of the optimal stochastic policy.*

## 4.5 Case II: Trajectory-optimized LQR (T-LQR)

In this section, we provide the theoretical basis for our proposed T-LQR design approach. The analysis employs a Taylor series expansion of the process model and large deviations theory. We also prove its near-first-order optimality in this section.

### *4.5.1 Preliminaries*

**Problem 8 Deterministic Open-Loop Problem***: Given an initial state* $\mathbf{x}_0$*, we begin by determining an optimal open-loop sequence for*

$$\min_{\mathbf{u}_{0:K-1}} \sum_{t=0}^{K-1} c_t(\mathbf{x}_t, \mathbf{u}_t) + c_K(\mathbf{x}_K)$$

$$s.t.\ \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t). \tag{4.22}$$

*Nominal trajectories:* For $0 \leq t \leq K-1$, let $\mathbf{u}_t^p$ be the optimal open-loop solution of the deterministic problem above, and let $\mathbf{x}_t^p$ be the corresponding state, where

$$\mathbf{x}_{t+1}^p := \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p), \tag{4.23}$$

where $\mathbf{x}_0^p := \mathbf{x}_0$. We refer to this as the nominal trajectories.

*Linearization of the system equations:* We consider the application of a control $\mathbf{u}_t = \mathbf{u}_t^p + \tilde{\mathbf{u}}_t$ to the stochastic system. Denote the resulting trajectory by $\mathbf{x}_t = \mathbf{x}_t^p + \tilde{\mathbf{x}}_t$, where $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}_t^p$ denotes the state error. Then,

$$\mathbf{x}_{t+1}^p + \tilde{\mathbf{x}}_{t+1} = \mathbf{f}(\mathbf{x}_t^p + \tilde{\mathbf{x}}_t, \mathbf{u}_t^p + \tilde{\mathbf{u}}_t) + \epsilon \boldsymbol{\sigma}_t \mathbf{w}_t. \tag{4.24}$$

Next, we linearize the drift of the process model around its nominal counterparts. Then for $0 \leq t \leq K-1$:

$$\begin{aligned} \tilde{\mathbf{x}}_{t+1} &= \mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{G}_t\mathbf{w}_t + o(\|\tilde{\mathbf{x}}_t\| + \|\tilde{\mathbf{u}}_t\|) && \text{(4.25a)} \\ &= \mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{G}_t\mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{u}}\|_\infty), && \text{(4.25b)} \end{aligned}$$

as $(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{u}}\|_\infty) \downarrow 0$, where we have:

- $\mathbf{A}_t := \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^p,\mathbf{u}_t^p}$, $\mathbf{B}_t := \nabla_{\mathbf{u}}\mathbf{f}(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^p,\mathbf{u}_t^p}$, $\mathbf{G}_t := \epsilon\boldsymbol{\sigma}_t$;
- $\tilde{\mathbf{u}}_0 = \mathbf{u}_0 - \mathbf{u}_0^p = \mathbf{0}$, and $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}_0^p = \mathbf{0}$.

*The exactly linear $l$-system:* From the above system of (4.25), we remove the $o(\cdot)$ terms, and define an exactly linear system:

$$\tilde{\mathbf{x}}_{t+1}^l := \mathbf{A}_t\tilde{\mathbf{x}}_t^l + \mathbf{B}_t\tilde{\mathbf{u}}_t^l + \mathbf{G}_t\mathbf{w}_t, \tag{4.26a}$$

where $\tilde{\mathbf{x}}_0^l := \tilde{\mathbf{x}}_0 = \mathbf{0}$.

*LQR policy:* Now we consider the design of an LQR policy for the $l$-system with the cost:

$$\min_{\boldsymbol{\pi}} \ \mathbb{E}[\sum_{t=0}^{K-1} (\tilde{\mathbf{x}}_t^l)^T\mathbf{W}_t^x\tilde{\mathbf{x}}_t^l + (\tilde{\mathbf{u}}_t^l)^T\mathbf{W}_t^u\tilde{\mathbf{u}}_t^l], \tag{4.27}$$

where $\mathbf{W}_t^u, \mathbf{W}_t^x \succeq 0$ are positive-definite matrices. This problem results in a policy $\tilde{\mathbf{u}}_t^l = -\mathbf{L}_t\tilde{\mathbf{x}}_t^l$, where the linear feedback gain $\mathbf{L}_t$ for $K-1 \geq t \geq 0$ can be obtained by:

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

and the matrix $\mathbf{P}_t^f$ is the result of backward iteration of the dynamic Riccati equation

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

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

Now, since $\tilde{\mathbf{x}}_t^l$ is fictitious, we use $\tilde{\mathbf{u}}_t = -\mathbf{L}_t \tilde{\mathbf{x}}_t$ in the original system. Then (4.25) can be rewritten as:

$$\begin{aligned} \tilde{\mathbf{x}}_{t+1} &= \mathbf{A}_t \tilde{\mathbf{x}}_t - \mathbf{B}_t \mathbf{L}_t \tilde{\mathbf{x}}_t + \mathbf{G}_t \mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{u}}\|_\infty), \qquad (4.28a) \\ &= \mathbf{A}_t \tilde{\mathbf{x}}_t - \mathbf{B}_t \mathbf{L}_t \tilde{\mathbf{x}}_t + \mathbf{G}_t \mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty), \qquad (4.28b) \end{aligned}$$

and the $l$-system becomes:

$$\tilde{\mathbf{x}}_{t+1}^l = \mathbf{A}_t \tilde{\mathbf{x}}_t^l - \mathbf{B}_t \mathbf{L}_t \tilde{\mathbf{x}}_t^l + \mathbf{G}_t \mathbf{w}_t. \qquad (4.29)$$

*The difference d-system:* We denote the difference between the two systems of (4.28) and (4.29) by a superscript $d$, and define for $0 \leq t \leq K-1$:

$$\begin{aligned} \tilde{\mathbf{u}}_t^d &:= \tilde{\mathbf{u}}_t - \tilde{\mathbf{u}}_t^l, & \tilde{\mathbf{u}}_t^d &= -\mathbf{L}_t(\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_t^l), \qquad (4.30a) \\ \tilde{\mathbf{x}}_{t+1}^d &:= \tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}_{t+1}^l, & \tilde{\mathbf{x}}_{t+1}^d &= \mathbf{A}_t \tilde{\mathbf{x}}_t^d + \mathbf{B}_t \tilde{\mathbf{u}}_t^d + o(\|\tilde{\mathbf{x}}\|_\infty), \qquad (4.30b) \end{aligned}$$

where $\tilde{\mathbf{u}}_0^d = \tilde{\mathbf{u}}_0 - \tilde{\mathbf{u}}_0^l = \mathbf{0}$, and $\tilde{\mathbf{x}}_0^d = \tilde{\mathbf{x}}_0 - \tilde{\mathbf{x}}_0^l = \mathbf{0}$. Thus,

$$\begin{aligned} \tilde{\mathbf{u}}_t^d &= -\mathbf{L}_t \tilde{\mathbf{x}}_t^d, \\ \tilde{\mathbf{x}}_{t+1}^d &= (\mathbf{A}_t - \mathbf{B}_t \mathbf{L}_t) \tilde{\mathbf{x}}_t^d + o(\|\tilde{\mathbf{x}}\|_\infty) = \mathbf{D}_t \tilde{\mathbf{x}}_t^d + o(\|\tilde{\mathbf{x}}\|_\infty) \\ &= \tilde{\mathbf{D}}_{0:t} \tilde{\mathbf{x}}_0^d + o(\|\tilde{\mathbf{x}}\|_\infty) = o(\|\tilde{\mathbf{x}}\|_\infty), \end{aligned}$$

where $\mathbf{D}_t := \mathbf{A}_t - \mathbf{B}_t\mathbf{L}_t, 1 \leq t \leq K-1$, $\mathbf{D}_0 := \mathbf{A}_0$, 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 the identity matrix. This leads to $\tilde{\mathbf{u}}_t^d = o(\|\tilde{\mathbf{x}}\|_\infty)$. Hence,

$$O(\|\tilde{\mathbf{x}}^l\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty) + o(\|\tilde{\mathbf{x}}\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty), \tag{4.31}$$

$$O(\|\tilde{\mathbf{u}}^l\|_\infty) = O(\|\tilde{\mathbf{x}}^l\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty), \tag{4.32}$$

$$O(\|\tilde{\mathbf{u}}\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty). \tag{4.33}$$

This means that all the errors in the original system, the $l$-system, and the $d$-system are of the order of $O(\|\tilde{\mathbf{x}}\|_\infty)$. Moreover, $O(\|\tilde{\mathbf{x}}\|_\infty)$ is itself $O(\|\tilde{\mathbf{x}}^l\|_\infty)$, which we calculate next.

*Large deviations:* The $l$-system is a linear Gaussian system with additive noise, for which we use the large deviations result of Lemma 1 modifying the definition of $\mathbf{\Phi}_{s,t}$ for $0 \leq s \leq t-1, 2 \leq t \leq K$ as $\mathbf{\Phi}_{s,t} := (\Pi_{r=s+1}^{t-1}\mathbf{D}_r)\boldsymbol{\sigma}_s$. Thus, for each finite $\delta \geq 0$, we have $P\{\max_{0\leq t\leq K}\|\tilde{\mathbf{x}}_t^l\| \geq \delta\} = o(\epsilon)$.

Let $\Omega(\delta)$ be the set where $\max_{0\leq t\leq K}\|\tilde{\mathbf{x}}_t^l\| \leq \delta$. Then, $P(\Omega(\delta)) \geq 1 - o(\epsilon)$ and for $\omega \in \Omega(\delta)$, $\|\tilde{\mathbf{x}}^l\|_\infty = O(\delta)$. Therefore, from the calculations above, we have that $O(\|\tilde{\mathbf{x}}\|_\infty) = O(\delta)$, and hence all the other errors are also $O(\delta)$ for $\omega \in \Omega(\delta)$.

Then for $\omega \in \Omega(\delta)$ and for all $0 \leq t \leq K-1$,

$$\mathbf{u}_t = \mathbf{u}_t^p + \tilde{\mathbf{u}}_t^l + O(\delta), \tag{4.34a}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^p + \tilde{\mathbf{x}}_{t+1}^l + O(\delta), \tag{4.34b}$$

which means that the linear Gaussian stochastic $\tilde{(\cdot)}^l$-system with the T-LQR control law along with the deterministic $p$-system can be used to control the original system given the $O(\delta)$ approximations hold (with probability of at least $1-o(\epsilon)$). In another interpretation, the original system can be approximated for all $0 \leq t \leq K-1$ as:

$$\mathbf{u}_t = \mathbf{u}_t^l + O(\delta), \tag{4.35a}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^l + O(\delta). \tag{4.35b}$$

### *4.5.2 First-Order Analysis*

In this section, we quantify the performance obtained from the above design.

**Lemma 3 State Error Propagation:** *For the l-system of* (4.29)*, the state error* $\tilde{\mathbf{x}}_{t+1}^l$ *can be written as:*

$$\tilde{\mathbf{x}}_{t+1}^l = \sum_{s=0}^{t} \tilde{\mathbf{D}}_{s,t}^{\mathbf{w}} \mathbf{w}_s, \ \ 0 \leq t \leq K-1, \tag{4.36}$$

*where we have:*

- $\tilde{\mathbf{D}}_{s,t}^{\mathbf{w}} := \tilde{\mathbf{D}}_{s+1:t}\mathbf{G}_s, 0 \leq s \leq t-1, t \geq 1$*; and*
- $\tilde{\mathbf{D}}_{t,t}^{\mathbf{w}} := \tilde{\mathbf{D}}_{t+1:t}\mathbf{G}_t = \mathbf{G}_t, t \geq 0.$

***Proof 5*** *Given* $\tilde{\mathbf{x}}_0^l = \mathbf{0}$*, we have:*

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

The following lemma follows directly by using the feedback law and the result of Lemma 3.

**Lemma 4 Control Error Propagation:** *For the l-system of* (4.29)*, the control* *error $\tilde{\mathbf{u}}_t^l$ can be written as*

$$\tilde{\mathbf{u}}_t^l = -\sum_{s=0}^{t-1} \mathbf{L}_{s,t}^{\mathbf{w}} \mathbf{w}_s,\ 1 \leq t \leq K-1,$$

*where $\mathbf{L}_{s,t}^{\mathbf{w}} := \mathbf{L}_t \tilde{\mathbf{D}}_{s,t-1}^{\mathbf{w}}, t \geq 1, t-1 \geq s \geq 0$.*

***Proof 6*** *Note that $\tilde{\mathbf{u}}_0^l = \mathbf{0}$. Now, using Lemma 3, we have:*

$$\tilde{\mathbf{u}}_t^l = -\mathbf{L}_t \tilde{\mathbf{x}}_t^l = -\mathbf{L}_t \sum_{s=0}^{t-1} \tilde{\mathbf{D}}_{s,t-1}^{\mathbf{w}} \mathbf{w}_s =: -\sum_{s=0}^{t-1} \mathbf{L}_{s,t}^{\mathbf{w}} \mathbf{w}_s.$$

Next, we linearize of the cost function and provide the decoupling principle for a fully-observed system.

*Linearization of the cost function:* We similarly linearize the cost function around the nominal trajectories of state and control actions:

$$J = J^p + \tilde{J}_1 + o(\sum_{t=1}^{K-1} (\|\tilde{\mathbf{x}}_t\| + \|\tilde{\mathbf{u}}_t\|) + \|\tilde{\mathbf{x}}_K\|) \tag{4.37a}$$

$$= J^p + \tilde{J}_1 + o(\|\tilde{\mathbf{x}}\|_\infty), \tag{4.37b}$$

where we have:

- $J^p := \sum_{t=0}^{K-1} c_t(\mathbf{x}_t^p, \mathbf{u}_t^p) + c_K(\mathbf{x}_K^p)$ denotes the nominal cost;
- $\tilde{J}_1 := \sum_{t=0}^{K-1} (\mathbf{C}_t^{\mathbf{x}} \tilde{\mathbf{x}}_t + \mathbf{C}_t^{\mathbf{u}} \tilde{\mathbf{u}}_t) + \mathbf{C}_K^{\mathbf{x}} \tilde{\mathbf{x}}_K$ is the first order cost error;
- $J_1 := J^p + \tilde{J}_1$ is the first order approximation of the cost;
- and $\mathbf{C}_t^{\mathbf{x}} = \nabla_{\mathbf{x}} c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{u}} = \nabla_{\mathbf{u}} c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_K^{\mathbf{x}} = \nabla_{\mathbf{x}} c_K(\mathbf{x})|_{\mathbf{x}_K^p}$.

Therefore, for $\omega \in \Omega(\delta)$, and

$$J = J^p + \sum_{t=0}^{K-1} (\mathbf{C}_t^{\mathbf{x}} \tilde{\mathbf{x}}_t + \mathbf{C}_t^{\mathbf{u}} \tilde{\mathbf{u}}_t) + \mathbf{C}_K^{\mathbf{x}} \tilde{\mathbf{x}}_K + O(\delta) \tag{4.38a}$$

$$= J^p + \sum_{t=0}^{K-1} (\mathbf{C}_t^{\mathbf{x}} \tilde{\mathbf{x}}_t^l + \mathbf{C}_t^{\mathbf{u}} \tilde{\mathbf{u}}_t^l) + \mathbf{C}_K^{\mathbf{x}} \tilde{\mathbf{x}}_K^l + O(\delta). \tag{4.38b}$$

Hence, $J - J_1 = O(\delta)$ for $\omega \in \Omega(\delta)$.

Next, we provide the main result regarding the expected first order error of the cost function.

**Theorem 3 First-Order Cost Function Error for a Fully-Observed System with T-LQR Policy:** *Given that process noises are zero mean i.i.d. Gaussian and all the functions are in $\mathbb{C}^1$, under a first-order approximation for the small noise paradigm, the stochastic cost function is dominated by the nominal part of the cost function, and the expected first-order error is $O(\delta)$. That is,*

$$\mathbb{E}[\tilde{J}_1] = O(\delta), \ \textit{and} \ \mathbb{E}[J] = J^p + O(\delta).$$

*Moreover, by choosing $\delta = \sqrt{\log(\frac{1}{\epsilon})\epsilon}$, we have*

$$\mathbb{E}[\tilde{J}_1] = O(\epsilon^{1-\gamma}), \ \textit{and} \ \mathbb{E}[J] = J^p + O(\epsilon^{1-\gamma}),$$

*for some $0 < \gamma \ll 1$, which shows that this error tends to zero with a near-first-order rate as $\epsilon \downarrow 0$.*

***Proof 7*** *Let $\tilde{J}_1^l{:=}\sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t^l{+}\mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t^l){+}\mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K^l$. Also note $\tilde{\mathbf{x}}_0 = \mathbf{0}$, and $\mathbb{E}[\mathbf{w}_t] = 0$ for all $t$. Then, we use Lemmas 3 and 4:*

$$\begin{aligned}
\mathbb{E}[\tilde{J}_1^l] &= \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\mathbb{E}[\tilde{\mathbf{x}}_t^l] + \mathbf{C}_t^{\mathbf{u}}\mathbb{E}[\tilde{\mathbf{u}}_t^l]) + \mathbf{C}_K^{\mathbf{x}}\mathbb{E}[\tilde{\mathbf{x}}_K^l] \\
&= \sum_{t=0}^{K} \mathbf{C}_t^{\mathbf{x}}\mathbb{E}[\sum_{s=0}^{t-1} \tilde{\mathbf{D}}_{s,t-1}^{\mathbf{w}}\mathbf{w}_s] + \sum_{t=0}^{K-1} \mathbf{C}_t^{\mathbf{u}}\mathbb{E}[-\sum_{s=0}^{t-1} \mathbf{L}_{s,t}^{\mathbf{w}}\mathbf{w}_s] \\
&= \sum_{t=0}^{K}\sum_{s=0}^{t-1} \mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{D}}_{s,t-1}^{\mathbf{w}}\mathbb{E}[\mathbf{w}_s] - \sum_{t=0}^{K-1}\sum_{s=0}^{t-1} \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}^{\mathbf{w}}\mathbb{E}[\mathbf{w}_s] = 0.
\end{aligned}$$

*Now, we take expectation from both sides of* (4.38b)*. Since, for $\omega \notin \Omega(\delta)$, $J \leq M$, then*

$$\begin{aligned}\mathbb{E}[J - J^p] &= P(\Omega(\delta))(\mathbb{E}[\tilde{J}_1^l] + O(\delta)) + M(1 - P(\Omega(\delta))) \\ &= P(\Omega(\delta))O(\delta) + M(1 - P(\Omega(\delta))) \qquad (4.39)\end{aligned}$$

*Now, the last expression is the same as* (4.19)*. Although $\Omega(\delta)$ is not the same as in Theorem 2, $P(\Omega(\delta))$ is still the same. In the proof of Theorem 2 while we discussed on the probabilistic argument and choosing the proper $\delta$, we showed that by choosing $\delta := \sqrt{-\log(\epsilon)}\epsilon$, the $\mathbb{E}[J - J^p] = O(\epsilon^{1-\gamma})$. The same argument follows through and this theorem is proved.*

*Remark:* This result shows that the T-LQR algorithm provides the same order of error as the deterministic policy. However, the T-LQR feedback gain is obtained once utilizing the dynamic Riccati equation, whereas to obtain the optimal deterministic policy the dynamic programming equation has to be solved which imposes a much higher computational burden. This result leads to the decoupled design of the feedback law from the open-loop control sequence, summarized next as the Decoupling Principle. Therefore, when the functions are in $\mathbb{C}^1$, the expected stochastic cost is equal to the nominal cost with a higher probability as $\epsilon \downarrow 0$ and the design approach is near-first-order optimal. However, in Section 4.8, we add the $\mathbb{C}^2$ assumption for all functions and prove that the Decoupling Principle (and the T-LQR design approach) is near-second-order optimal.

**Corollary 2 Decoupling Principle: Decoupling of the Open-Loop and Closed-Loop Designs Under Small Noise.** *Based on Theorem 3, for a fully-observed system under the small noise paradigm, as $\epsilon \downarrow 0$, the design of the feedback law can* *be decoupled from the design of the open-loop optimized trajectory. If the functions are in* $\mathbb{C}^1$*, this result is* $O(\epsilon^{1-\gamma})$*-optimal for* $0 < \gamma \ll 1$ *as* $\epsilon \downarrow 0$*.*

***Proof 8*** *Using Theorem 3, for* $\omega \in \Omega(\delta)$ *we have* $\mathbb{E}[J] = J^p + O(\epsilon^{1-\gamma})$*, which is the cost of applying policy* $\boldsymbol{\pi}_t(\mathbf{x}_t) = \mathbf{u}_t^p - \mathbf{L}_t(\mathbf{x}_t - \mathbf{x}_t^p)$ *to the stochastic system. Now, suppose* $\boldsymbol{\pi}^*$ *is the optimal stochastic policy. By assumption* $\boldsymbol{\pi}^*$ *is continuously differentiable. Therefore, by modifying the definition of* $\mathbf{L}_t$ *as* $\mathbf{L}_t = -\nabla_{\mathbf{x}}\boldsymbol{\pi}_t^*(\mathbf{x})|_{\mathbf{x}_t^{*p}}$*, defining* $\mathbf{u}_t^{*p} = \boldsymbol{\pi}_t^*(\mathbf{x}_t^{*p})$ *and replacing* $p$ *with* $*p$ *in (4.23), we have* $\boldsymbol{\pi}_t^*(\mathbf{x}_t) = \mathbf{u}_t^{*p} - \mathbf{L}_t(\mathbf{x}_t - \mathbf{x}_t^{*p}) + o(\|\tilde{\mathbf{x}}_t\|)$*. Similarly, we modify* $\tilde{\mathbf{u}}_t^d = -\mathbf{L}_t\tilde{\mathbf{x}}_t^d + o(\|\tilde{\mathbf{x}}_t\|)$ *and use appropriate modifications, whence the entire calculations of the previous sections hold for this policy. Hence, using Theorem 3 for this system, the cost function of policy* $\boldsymbol{\pi}^*$ *can be written as* $\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{1-\gamma})$*. Now, by construction* $J^p \leq J^{*p}$*, and*

$$\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{1-\gamma}) \geq J^p + O(\epsilon^{1-\gamma}) = \mathbb{E}[J_{\boldsymbol{\pi}}] + O(\epsilon^{1-\gamma})$$

*As a result, policy* $\boldsymbol{\pi}$ *is within* $O(\epsilon^{1-\gamma})$ *of the optimal stochastic policy.*

### *4.5.3 Discussion*

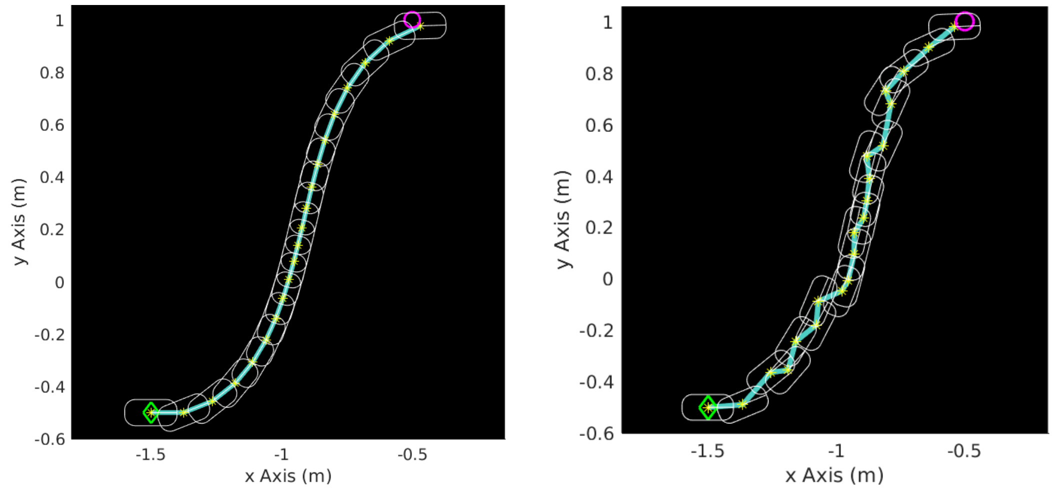
*Remark:* This result means that under a small noise assumption, an open-loop nominal trajectory of the system can be designed by replacing the stochastic equations with their nominal counterparts. Then, a decentralized feedback control law can be designed using the LQG theory. This design is near-optimal as the intensity of noise tends to zero. We show in the example below that this design procedure can be used even for moderate levels of noise.

*Remark:* In Ref. [64], for a special case of nonlinear systems where the process model is linear in the control variable, i.e., $\mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) = f_1(\mathbf{x}_t) + f_2(\mathbf{x}_t)\mathbf{u}_t$, and the process model is perturbed by additive noise with $\epsilon$-variance, three results are proven.

The first result, concerns the $\epsilon$-optimality of the optimal deterministic law under convexity of $J$ in the control (i.e., $\mathbf{v}^T(\nabla_{\mathbf{u},\mathbf{u}}J)\mathbf{v} \succeq 0 \ , \forall \mathbf{v}$), and additional smoothness and regularity conditions. The second result concerns the $O(\epsilon^2)$-optimality of the optimal deterministic law under a stronger convexity condition of $J$ in the control (i.e., $\mathbf{v}^T(\nabla_{\mathbf{u},\mathbf{u}}J)\mathbf{v} \succeq c(\|\mathbf{u}\|)\|\mathbf{v}\|^2 \ , \forall \mathbf{v}$, $c(\cdot) : \mathbb{R} \to \mathbb{R}$ is a monotonically non-increasing positive function), and some smoothness and regularity conditions. The third result concerns the $\epsilon$-optimality of the optimal deterministic sequence under the latter condition. Our result, on the other hand, provides $O(\epsilon)$-optimality of the proposed design approach for a broader class of processes $\mathbf{f}(\mathbf{x}_t, \mathbf{u}_t)$ with nonlinear dependence in the control variable and more general cost functions. Most importantly, it does not assume the linear dependence on the control sequence. In fact, our simulations are performed for a car-like robot with nonlinear dependence on the control variables. One can find more details on these and similar results in the literature review chapter of the Dissertation.

*Feedback control:* The proposed approach aims at designing an LQR controller with an optimal nominal underlying trajectory based on the decoupling principle of Corollary 2 and Theorem 3. As a result, we term this method as the Trajectory-optimized LQR (T-LQR). Although we utilize an LQR controller, it is important to note that the decoupling result only assumes a linear form of feedback and other types of designs [159] can be used as well.

*Remark:* The computation involved in problem (8) is of the order of $O(Kn_x^2)$ for typically smooth dynamics for one iteration. Let us assume $O(\ell)$ is the order of the number of iterations in the optimizer until convergence. The LQR policy calculation is of order of $O(Kn_x^3)$. Therefore, T-LQR's computations are $O(\ell K n_x^2 + K n_x^3)$ for a typical process model (such as our example in the next section). The low computational complexity of this approach results in fast replanning in case of deviations



(a) Optimized trajectory of problem (8).

(b) A typical ground truth trajectory with noise standard deviation equal to 10% of the maximum control signal.

Figure 4.1: Optimized vs. a typical execution trajectory for a car-like robot.

during execution. This renders the T-LQR scheme eminently implementable for use in on-line applications.

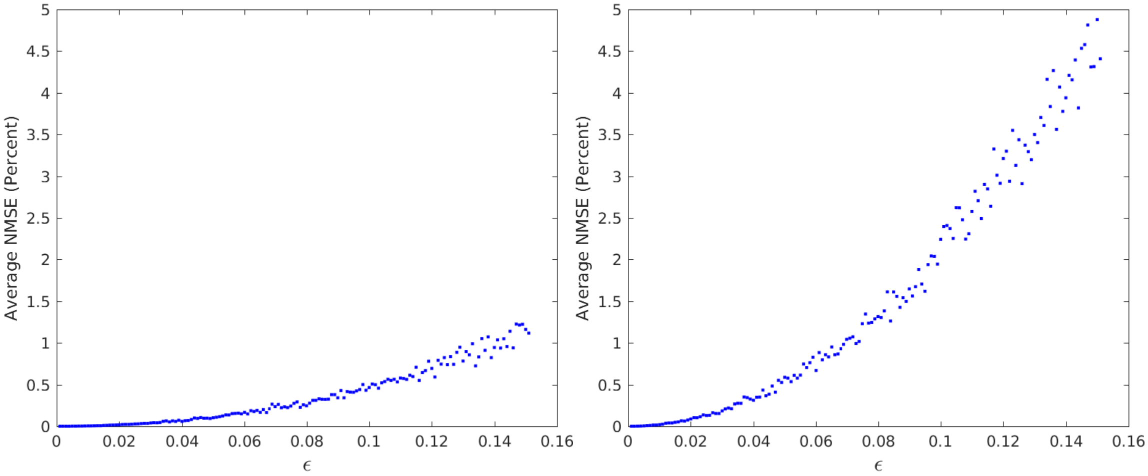
*Remark:* For the specific class of problems considered in [64] the design approach of [64] requires calculation of the optimal control law through intractable dynamic programming. In contrast, the proposed design approach utilizes the more tractable solution via Maximum Principle, followed by an LQR design. Even implementing the result of [64] through a model predictive approach would require more computations of at least an order of the planning horizon (from $O(K)$ to $O(K^2)$). In such an implementation, the online computation of the approach of [64] is $O(\ell K n_x^2)$ compared to only $O(n_x^2)$ in our algorithm.

## 4.6 Example

Let us consider a car-like four-wheel robot with process model [179]:

$$\dot{x} = v\cos(\theta),\ \dot{y} = v\sin(\theta),\ \dot{\theta} = \frac{v}{L}\tan(\phi), \tag{4.40}$$

where $(x, y, \theta)$ is the state, and $(v, \phi)$ is the control input. We suppose that, $|\phi| < \phi_{\max} = \pi/2$, $|v| \leq v_{\max} = 0.6$, $\mathbf{x}_0 = (-1.5, 0.5, 0)$, $K = 20$, and the time discretization period is 0.7. We incorporate the control constraints and the terminal goal, $\mathbf{x}_g = (-0.5, 1, 0)$, in the cost function. Last, the initial control sequence used for the optimization is just a sequence of zero inputs. The process noise is additive mean zero Gaussian noise with a standard deviation equal to $\epsilon \max_t\{\|\mathbf{u}_t\|_2\}$. Figure 4.1a shows the result of the optimization problem (8) whereas Fig. 4.1b shows a typical ground truth trajectory with $\epsilon = 0.1$. We have used MATLAB 2016b and its `fmincon` solver for simulations.



(a) Feedback-compensated system.

(b) Open-loop system.

Figure 4.2: Evolution of average NMSE as $\epsilon \downarrow 0$ for feedback compensated and open-loop systems with the same nominal trajectories.

In the next experiment, we increase $\epsilon$ from 0.001 to 0.1501, in step sizes of 0.001. For each value of $\epsilon$, we execute the resulting policy 100 times and compute the average Normalized Mean Squared Error (NMSE) as:

$$\text{Average NMSE (\%)} = \frac{1}{100}\sum_{j=1}^{100}\frac{\|\mathbf{x}^p - \mathbf{x}^j\|_2^2}{\|\mathbf{x}^p\|_2^2} \times 100, \tag{4.41}$$

where $\mathbf{x}^p$ indicates the planned trajectory and $\mathbf{x}^j$ indicates the ground truth trajectory at $j$th experiment. The results of this experiment are shown in Fig. 4.2a, where the evolution of the average NMSE is depicted for various values of noise level $\epsilon$. As indicated in this figure, as $\epsilon \downarrow 0$, the average NMSE tends to zero at a near-exponential rate, which is consistent with the theory developed in Section 4.2. Moreover, this figure indicates that through the feedback compensation, moderate noise levels can be tolerated, rather than just small levels.

Last, Fig. 4.2b depicts the evolution of the average NMSE for an experiment with the same setting as in Fig. 4.2a, except that only the open-loop planned control sequence is applied during execution. As predicted by the theory, the error still decreases exponentially as the noise level decreases. However, the rate of convergence is about one-fifth of the previous rate. The results of Fig. 4.2 show that our design can be used for relatively moderate levels of noise, using the power of feedback.

*Remark:* In practice, if at any point in the execution the calculated error exceeds a threshold, very rapid replanning can be triggered very fast due to the low computational burden of the optimization problem.

## 4.7 Second-Order Optimality of The Deterministic Law

In this section, we provide a second-order analysis of the deterministic feedback law and show that applying the optimal feedback law of the deterministic problem to the stochastic problem results in a second-order optimality as well. Therefore, we improve the results of Section 4.4.

*Assumptions:* Other than the assumptions of Section 4.4, we assume for the analysis of this section that all the functions (including the dynamics, feedback law, and the cost functions) are in $\mathbb{C}^2$, i.e., they are continuously differentiable to the second order.

*Second-order expansion of the control law:* Here, we will use the same policy $\mathbf{u}_t = \boldsymbol{\pi}_t^d(\mathbf{x}_t)$ defined in Section 4.4. However, as opposed to that section, for the analysis of this section we expand this law to the second-order. Let us define $\mathbf{u}_t^p := \boldsymbol{\pi}_t^d(\mathbf{x}_t^p)$, $\tilde{\mathbf{u}}_t := \mathbf{u}_t - \mathbf{u}_t^p$ and $\tilde{\mathbf{x}}_t$ as before. Then,

$$\tilde{\mathbf{u}}_t = \boldsymbol{\pi}_t^d(\mathbf{x}_t) - \boldsymbol{\pi}_t^d(\mathbf{x}_t^p) = -\mathbf{L}_t\tilde{\mathbf{x}}_t + \begin{pmatrix} \tilde{\mathbf{x}}_t^T \mathbf{H}_t^{\pi^1} \tilde{\mathbf{x}}_t \\ \vdots \\ \tilde{\mathbf{x}}_t^T \mathbf{H}_t^{\pi^{nx}} \tilde{\mathbf{x}}_t \end{pmatrix} + o(\|\tilde{\mathbf{x}}_t\|^2) \tag{4.42a}$$

$$= -\mathbf{L}_t\tilde{\mathbf{x}}_t + \sum_{k=1}^{n_u} (\tilde{\mathbf{x}}_t^T \mathbf{H}_t^{\pi^k} \tilde{\mathbf{x}}_t)\mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}\|_\infty^2), \tag{4.42b}$$

- $\mathbf{L}_t := -\nabla_{\mathbf{x}} \boldsymbol{\pi}_t^d(\mathbf{x})|_{\mathbf{x}_t^p}$;
- $\boldsymbol{\pi}_t^d(\mathbf{x}) = (\pi^{d_k}(\mathbf{x})), 1 \leq k \leq n_u$;
- $\mathbf{H}_t^{\pi^k} := \frac{1}{2}\nabla^2_{\mathbf{x}\mathbf{x}} \boldsymbol{\pi}_t^{d_k}(\mathbf{x})|_{\mathbf{x}_t^p}$, where $\nabla^2_{\cdot,\cdot}$ denotes the second order derivative (Hessian) operator with respect to two variables in the written order;
- $\mathbf{e}_k^{n_u}$ is the $n_u$-dimensional unit vector with all the elements being zero except for the $k$-th element which equals one; and
- $\tilde{\mathbf{u}}_0 = \mathbf{u}_0 - \mathbf{u}_0^p = \mathbf{0}$, and $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}_0^p = \mathbf{0}$.

*Second-order expansion of the dynamics:* Let us first obtain the second-order expansion of the process model around the nominal trajectory. Then for $0 \leq t \leq$

$K-1$:

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) - \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p) + \epsilon \boldsymbol{\sigma}_t \mathbf{w}_t \tag{4.43a}$$

$$= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \mathbf{G}_t \mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,1} & \mathbf{F}_{\mathbf{xu}}^{t,1} \\ \mathbf{F}_{\mathbf{ux}}^{t,1} & \mathbf{F}_{\mathbf{uu}}^{t,1} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,n_x} & \mathbf{F}_{\mathbf{xu}}^{t,n_x} \\ \mathbf{F}_{\mathbf{ux}}^{t,n_x} & \mathbf{F}_{\mathbf{uu}}^{t,n_x} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}_t\|^2 + \|\tilde{\mathbf{u}}_t\|^2) \tag{4.43b}$$

$$= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \mathbf{G}_t \mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,1} & \mathbf{F}_{\mathbf{xu}}^{t,1} \\ \mathbf{F}_{\mathbf{ux}}^{t,1} & \mathbf{F}_{\mathbf{uu}}^{t,1} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,n_x} & \mathbf{F}_{\mathbf{xu}}^{t,n_x} \\ \mathbf{F}_{\mathbf{ux}}^{t,n_x} & \mathbf{F}_{\mathbf{uu}}^{t,n_x} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2), \tag{4.43c}$$

as $(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2) \downarrow 0$, where we have:

- $\mathbf{A}_t := \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{B}_t := \nabla_{\mathbf{u}} \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{G}_t := \epsilon \boldsymbol{\sigma}_t$;
- $\mathbf{f}(\mathbf{x}, \mathbf{u}) = (f^j(\mathbf{x}, \mathbf{u})), 1 \leq j \leq n_x$;
- $\mathbf{F}_{\mathbf{xx}}^{t,j} := \frac{1}{2} \nabla_{\mathbf{xx}}^2 f^j(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}, \mathbf{F}_{\mathbf{xu}}^{t,j} := \frac{1}{2} \nabla_{\mathbf{xu}}^2 f^j(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}, \mathbf{F}_{\mathbf{ux}}^{t,j} := \frac{1}{2} \nabla_{\mathbf{ux}}^2 f^j(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, and $\mathbf{F}_{\mathbf{uu}}^{t,j} := \frac{1}{2} \nabla_{\mathbf{uu}}^2 f^j(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$;
- $\tilde{\mathbf{u}}_0 = \mathbf{u}_0 - \mathbf{u}_0^p = \mathbf{0}$, and $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}_0^p = \mathbf{0}$.

*Feedback compensation:* Next, we replace the feedback law of (4.42b) into the last equation. Note that after the feedback compensation, the first-order terms of (4.43c) which are linear in $\tilde{\mathbf{u}}_t$, result in both first-order and second-order expressions in $\tilde{\mathbf{x}}_t$.

On the other hand, replacing the second order terms of the feedback law into the second order terms of the dynamics in (4.43c) results in second-, third- and fourth-order expressions in $\tilde{\mathbf{x}}_t$. However, since the error term in (4.43c) includes $o(\|\tilde{\mathbf{x}}\|_\infty^2)$, the third- and fourth-order terms can be ignored. As a result, we replace those terms with $o(\|\tilde{\mathbf{x}}\|_\infty^2)$, and write the following:

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{G}_t\mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,1} & \mathbf{F}_{\mathbf{xu}}^{t,1} \\ \mathbf{F}_{\mathbf{ux}}^{t,1} & \mathbf{F}_{\mathbf{uu}}^{t,1} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,n_x} & \mathbf{F}_{\mathbf{xu}}^{t,n_x} \\ \mathbf{F}_{\mathbf{ux}}^{t,n_x} & \mathbf{F}_{\mathbf{uu}}^{t,n_x} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2) \tag{4.44a}$$

$$\begin{aligned} =&\mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t(-\mathbf{L}_t\tilde{\mathbf{x}}_t + \sum_{k=1}^{n_u}(\tilde{\mathbf{x}}_t^T\mathbf{H}_t^{\pi^k}\tilde{\mathbf{x}}_t)\mathbf{e}_k^{n_u}) + \mathbf{G}_t\mathbf{w}_t \\ &+ \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\mathbf{L}_t\tilde{\mathbf{x}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,1} & \mathbf{F}_{\mathbf{xu}}^{t,1} \\ \mathbf{F}_{\mathbf{ux}}^{t,1} & \mathbf{F}_{\mathbf{uu}}^{t,1} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\mathbf{L}_t\tilde{\mathbf{x}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\mathbf{L}_t\tilde{\mathbf{x}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,n_x} & \mathbf{F}_{\mathbf{xu}}^{t,n_x} \\ \mathbf{F}_{\mathbf{ux}}^{t,n_x} & \mathbf{F}_{\mathbf{uu}}^{t,n_x} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\mathbf{L}_t\tilde{\mathbf{x}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \end{aligned} \tag{4.44b}$$

$$=\mathbf{A}_t\tilde{\mathbf{x}}_t - \mathbf{B}_t\mathbf{L}_t\tilde{\mathbf{x}}_t + \sum_{k=1}^{n_u}(\tilde{\mathbf{x}}_t^T\mathbf{H}_t^{\pi^k}\tilde{\mathbf{x}}_t)\mathbf{B}_t\mathbf{e}_k^{n_u} + \mathbf{G}_t\mathbf{w}_t + \begin{pmatrix} \tilde{\mathbf{x}}_t^T\mathbf{H}_t^{f^1}\tilde{\mathbf{x}}_t \\ \vdots \\ \tilde{\mathbf{x}}_t^T\mathbf{H}_t^{f^{nx}}\tilde{\mathbf{x}}_t \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \tag{4.44c}$$

$$=\mathbf{D}_t\tilde{\mathbf{x}}_t + \sum_{k=1}^{n_u}(\tilde{\mathbf{x}}_t^T\mathbf{H}_t^{\pi^k}\tilde{\mathbf{x}}_t)\mathbf{B}_t\mathbf{e}_k^{n_u} + \sum_{j=1}^{n_x}(\tilde{\mathbf{x}}_t^T\mathbf{H}_t^{f^j}\tilde{\mathbf{x}}_t)\mathbf{e}_j^{n_x} + \mathbf{G}_t\mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty^2) \tag{4.44d}$$

$$=\tilde{\mathbf{D}}_{0:t}\tilde{\mathbf{x}}_0 + \sum_{r=0}^{t}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + \sum_{r=0}^{t}\tilde{\mathbf{D}}_{r+1:t}\sum_{k=1}^{n_u}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{\pi^k}\tilde{\mathbf{x}}_r)\mathbf{B}_t\mathbf{e}_k^{n_u}$$

$$
\begin{aligned}
&+\sum_{r=0}^{t} \tilde{\mathbf{D}}_{r+1:t} \sum_{j=1}^{n_x} (\tilde{\mathbf{x}}_r^T \mathbf{H}_r^{f^j} \tilde{\mathbf{x}}_r) \mathbf{e}_j^{n_x} + o(\|\tilde{\mathbf{x}}\|_\infty^2) &\text{(4.44e)}\\
=&\sum_{r=0}^{t} \tilde{\mathbf{D}}_{r+1:t} \mathbf{G}_r \mathbf{w}_r + \sum_{r=0}^{t} \sum_{k=1}^{n_u} (\tilde{\mathbf{x}}_r^T \mathbf{H}_r^{\pi^k} \tilde{\mathbf{x}}_r) \tilde{\mathbf{D}}_{r+1:t} \mathbf{B}_t \mathbf{e}_k^{n_u} \\
&+\sum_{r=0}^{t} \sum_{j=1}^{n_x} (\tilde{\mathbf{x}}_r^T \mathbf{H}_r^{f^j} \tilde{\mathbf{x}}_r) \tilde{\mathbf{D}}_{r+1:t} \mathbf{e}_j^{n_x} + o(\|\tilde{\mathbf{x}}\|_\infty^2), &\text{(4.44f)}
\end{aligned}
$$

where we have used the fact that for $1 \leq j \leq n_x$, we can evaluate the following scalar value, and define $\mathbf{H}_t^{f^j} := (\mathbf{F}_{\mathbf{xx}}^{t,j} - \mathbf{F}_{\mathbf{xu}}^{t,j} \mathbf{L}_t - \mathbf{L}_t^T \mathbf{F}_{\mathbf{ux}}^{t,j} + \mathbf{L}_t^T \mathbf{F}_{\mathbf{uu}}^{t,j} \mathbf{L}_t)$, such that

$$
\begin{aligned}
&\begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\mathbf{L}_t \tilde{\mathbf{x}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,j} & \mathbf{F}_{\mathbf{xu}}^{t,j} \\ \mathbf{F}_{\mathbf{ux}}^{t,j} & \mathbf{F}_{\mathbf{uu}}^{t,j} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\mathbf{L}_t \tilde{\mathbf{x}}_t \end{pmatrix} \\
&= \tilde{\mathbf{x}}_t^T \mathbf{F}_{\mathbf{xx}}^{t,j} \tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_t^T \mathbf{F}_{\mathbf{xu}}^{t,j} \mathbf{L}_t \tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_t^T \mathbf{L}_t^T \mathbf{F}_{\mathbf{ux}}^{t,j} \tilde{\mathbf{x}}_t + \tilde{\mathbf{x}}_t^T \mathbf{L}_t^T \mathbf{F}_{\mathbf{uu}}^{t,j} \mathbf{L}_t \tilde{\mathbf{x}}_t \\
&= \tilde{\mathbf{x}}_t^T (\mathbf{F}_{\mathbf{xx}}^{t,j} - \mathbf{F}_{\mathbf{xu}}^{t,j} \mathbf{L}_t - \mathbf{L}_t^T \mathbf{F}_{\mathbf{ux}}^{t,j} + \mathbf{L}_t^T \mathbf{F}_{\mathbf{uu}}^{t,j} \mathbf{L}_t) \tilde{\mathbf{x}}_t \\
&= \tilde{\mathbf{x}}_t^T \mathbf{H}_t^{f^j} \tilde{\mathbf{x}}_t.
\end{aligned}
$$

Note, $\tilde{\mathbf{D}}_{r+1:t} \mathbf{e}_j^{n_x}$ evaluates to the $j$-th column of the $\tilde{\mathbf{D}}_{r+1:t}$ matrix (which is multiplied by the scalar value of $(\tilde{\mathbf{x}}_r^T \mathbf{H}_r^{f^j} \tilde{\mathbf{x}}_r)$ in (4.44f)). Similarly, $\tilde{\mathbf{D}}_{r+1:t} \mathbf{B}_t \mathbf{e}_k^{n_u}$ evaluates to the $k$-th column of $\tilde{\mathbf{D}}_{r+1:t} \mathbf{B}_t$. Note since we assume the continuity and second-order differentiability of the functions and continuity of the second-order partial derivatives, the Hessian (by Schwarz's integrability condition [180]) are also symmetric. Hence, $\mathbf{F}_{\mathbf{xu}}^{t,j} = (\mathbf{F}_{\mathbf{ux}}^{t,j})^T$.

*Validity region:* Note that the definition of $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}_t^p$. Therefore, the properties of $O(\|\tilde{\mathbf{x}}_t\|_\infty)$ that we have proven in Section 4.4 for a deterministic feedback design still hold for the above linearization, as well. Particularly, we proved that for the $\boldsymbol{\pi}^d$ design, $O(\|\tilde{\mathbf{x}}_t\|_\infty) = O(\delta)$ in a set $\Omega(\delta)$ properly defined as before with probability $1 - o(\epsilon)$. Hence, for $\omega \in \Omega(\delta)$, $O(\|\tilde{\mathbf{x}}_t\|_\infty^2) = O(\delta^2)$. Thus, for $\omega \in \Omega(\delta)$ (the same set and with the same probability), we have:

$$\begin{aligned}\tilde{\mathbf{x}}_{t+1} =& \sum_{r=0}^{t}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + \sum_{r=0}^{t}\sum_{j=1}^{n_x}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{f^j}\tilde{\mathbf{x}}_r)\tilde{\mathbf{D}}_{r+1:t}\mathbf{e}_j \\ &+\sum_{r=0}^{t}\sum_{k=1}^{n_u}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{\pi^k}\tilde{\mathbf{x}}_r)\tilde{\mathbf{D}}_{r+1:t}\mathbf{B}_t\mathbf{e}_k^{n_u} + O(\delta^2). \qquad (4.45)\end{aligned}$$

*Second-order expansion of the cost function:* Similarly, we obtain the second-order Taylor series expansion of the cost function around the nominal trajectory:

$$\begin{aligned}J =& J^p + \tilde{J}_1 + \tilde{J}_2 + o(\sum_{t=1}^{K-1}(\|\tilde{\mathbf{x}}_t\|^2 + \|\tilde{\mathbf{u}}_t\|^2) + \|\tilde{\mathbf{x}}_K\|^2) \qquad (4.46a)\\ =& J^p + \tilde{J}_1 + \tilde{J}_2 + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2), \qquad (4.46b)\end{aligned}$$

as $(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2) \downarrow 0$. Moreover, we have:

- $J^p{:=}\sum_{t=0}^{K-1}c_t(\mathbf{x}_t^p, \mathbf{u}_t^p) + c_K(\mathbf{x}_K^p)$ denotes the nominal cost;
- $\tilde{J}_1{:=}\sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t + \mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t) + \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K$ is the first order cost error;
- $\tilde{J}_2 := \sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xx}}\tilde{\mathbf{x}}_t + \frac{1}{2}\tilde{\mathbf{u}}_t^T\mathbf{C}_t^{\mathbf{uu}}\tilde{\mathbf{u}}_t + \tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xu}}\tilde{\mathbf{u}}_t) + \frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}_K^{\mathbf{xx}}\tilde{\mathbf{x}}_K$ is the second order cost error.
- $J_2 := J^p + \tilde{J}_1 + \tilde{J}_2$ is the second order approximation of the cost function;
- $\mathbf{C}_t^{\mathbf{xx}} = \nabla^2_{\mathbf{xx}}c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{uu}} = \nabla^2_{\mathbf{uu}}c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{xu}} = \nabla^2_{\mathbf{xu}}c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, and $\mathbf{C}_K^{\mathbf{xx}} = \nabla^2_{\mathbf{xx}}c_K(\mathbf{x})|_{\mathbf{x}_K^p}$, where we have used the fact that $c_t \in \mathbb{C}^2$.

Next, we provide the main result regarding the expected second order error of the cost function.

**Theorem 4 Second-Order Cost Function Error for a Fully-Observed System Using a Deterministic Policy:** *Given that process noises are zero mean i.i.d. Gaussian and all the functions are in $\mathbb{C}^2$, under a first-order approximation for the* *small noise paradigm, the stochastic cost function is dominated by the nominal part of the cost function, and the expected first-order error is $O(\delta^2)$. That is,*

$$\mathbb{E}[\tilde{J}_1] = O(\delta^2), \ \textit{and}\ \mathbb{E}[J] = J^p + O(\delta^2).$$

*Moreover, by choosing $\delta = \sqrt{2\log(\frac{1}{\epsilon})}\epsilon$, we have*

$$\mathbb{E}[\tilde{J}_1] = O(\epsilon^{2-\gamma}), \ \textit{and}\ \mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma}),$$

*for some $0 < \gamma \ll 1$, which shows that this error tends to zero with a near-first-order rate as $\epsilon \downarrow 0$.*

***Proof 9*** *First, let us simply the first order cost error:*

$$\begin{aligned}
\tilde{J}_1 &= \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t+\mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t)+\mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K \\
&= \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t - \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_t\tilde{\mathbf{x}}_t + \mathbf{C}_t^{\mathbf{u}}\sum_{k=1}^{n_u}(\tilde{\mathbf{x}}_t^T\mathbf{H}_t^{\pi^k}\tilde{\mathbf{x}}_t)\mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}\|_\infty^2))+\mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K \\
&=\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{x}}_t + \sum_{t=0}^{K-1}\sum_{k=1}^{n_u}(\tilde{\mathbf{x}}_t^T\mathbf{H}_t^{\pi^k}\tilde{\mathbf{x}}_t)\mathbf{C}_t^{\mathbf{u}}\mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
&=\sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + \sum_{t=0}^{K}\sum_{r=0}^{t-1}\sum_{j=1}^{n_x}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{f^j}\tilde{\mathbf{x}}_r)\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t-1}\mathbf{e}_j^{n_x} \\
&\quad + \sum_{t=0}^{K-1}\sum_{k=1}^{n_u}(\tilde{\mathbf{x}}_t^T\mathbf{H}_t^{\pi^k}\tilde{\mathbf{x}}_t)\mathbf{C}_t^{\mathbf{u}}\mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}\|_\infty^2),
\end{aligned}$$

*where $\mathbf{C}_t^{\mathbf{L}} := \mathbf{C}_t^{\mathbf{x}} - \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_t, 0 \le t \le K-1$, and $\mathbf{C}_K^{\mathbf{L}} = \mathbf{C}_K^{\mathbf{x}}$. Note that all the terms in the above equation evaluate to scalar values. For instance, $\mathbf{C}_t^{\mathbf{u}}\mathbf{e}_k^{n_u}$ is a scalar value.*

*Next, we simplify the second order cost error. Once again, we ignore the second* *order feedback terms and replace them with* $o(\|\tilde{\mathbf{x}}\|_\infty^2)$

$$\begin{aligned}
\tilde{J}_2 &= \sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xx}}\tilde{\mathbf{x}}_t + \frac{1}{2}\tilde{\mathbf{u}}_t^T\mathbf{C}_t^{\mathbf{uu}}\tilde{\mathbf{u}}_t + \tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xu}}\tilde{\mathbf{u}}_t) + \frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}_K^{\mathbf{xx}}\tilde{\mathbf{x}}_K \\
&= \sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xx}}\tilde{\mathbf{x}}_t + \frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{L}_t^T\mathbf{C}_t^{\mathbf{uu}}\mathbf{L}_t\tilde{\mathbf{x}}_t + \tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xu}}\mathbf{L}_t\tilde{\mathbf{x}}_t) + \frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}_K^{\mathbf{xx}}\tilde{\mathbf{x}}_K + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
&= \sum_{t=0}^{K}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{LL}}\tilde{\mathbf{x}}_t + o(\|\tilde{\mathbf{x}}\|_\infty^2),
\end{aligned}$$

*where* $\mathbf{C}_t^{\mathbf{LL}} := \frac{1}{2}\mathbf{C}_t^{\mathbf{xx}} + \frac{1}{2}\mathbf{L}_t^T\mathbf{C}_t^{\mathbf{uu}}\mathbf{L}_t + \mathbf{C}_t^{\mathbf{xu}}\mathbf{L}_t, 0 \le t \le K-1$*, and* $\mathbf{C}_K^{\mathbf{LL}} := \frac{1}{2}\mathbf{C}_K^{\mathbf{xx}}$*. Hence, we have:*

$$\begin{aligned}
\tilde{J}_1 + \tilde{J}_2 =& \sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + \sum_{t=0}^{K}\sum_{r=0}^{t-1}\sum_{j=1}^{n_x}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{fj}\tilde{\mathbf{x}}_r)\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t-1}\mathbf{e}_j^{n_x} \\
&+ \sum_{t=0}^{K-1}\sum_{k=1}^{n_u}(\tilde{\mathbf{x}}_t^T\mathbf{H}_t^{\pi^k}\tilde{\mathbf{x}}_t)\mathbf{C}_t^{\mathbf{u}}\mathbf{e}_k^{n_u} + \sum_{t=0}^{K}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{LL}}\tilde{\mathbf{x}}_t + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
=& \sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + O(\|\tilde{\mathbf{x}}\|_\infty^2) + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
=& \sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + O(\|\tilde{\mathbf{x}}\|_\infty^2).
\end{aligned}$$

*Hence, for* $\omega \in \Omega(\delta)$*,*

$$\tilde{J}_1 + \tilde{J}_2 = \sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + O(\delta^2).$$

*As a result, using* (4.52b)*, for* $\omega \in \Omega(\delta)$*, we have:*

$$J = J^p + \sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + O(\delta^2).$$

*Next, note that* $\mathbb{E}[\mathbf{w}_t] = 0$ *for all* $t$*, and*

$$\mathbb{E}[\sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r] = \sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbb{E}[\mathbf{w}_r] = 0.$$

***The probabilistic argument and choosing the proper*** $\delta$***:*** *Now, we take expectation from both sides of* (4.38b)*. In order to choose* $\delta$ *we will follow a similar line of argument as that used in proving Theorem 2. First, note that since for* $\omega \notin \Omega(\delta)$*,* $J \leq M$*, then*

$$\begin{aligned}\mathbb{E}[J - J^p] &= P(\Omega(\delta))(\mathbb{E}[\sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r] + O(\delta^2)) + M(1 - P(\Omega(\delta)))\\ &= P(\Omega(\delta))O(\delta^2) + M(1 - P(\Omega(\delta))) \qquad (4.47)\end{aligned}$$

*where* $\Omega(\delta)$ *is also the same as in Theorem 2. As mentioned before,* $P(\Omega(\delta)) \geq 1 - Kn_x\bar{\beta}\frac{\epsilon}{\delta}\exp(-\bar{\gamma}\frac{\delta^2}{\epsilon^2})$*. Once again, since we are only interested in the order of the above expectation, we will calculate the* $O(P(\Omega(\delta))O(\delta)+M(1-P(\Omega(\delta))))$*. Therefore, for the purpose of calculations, we ignore the inequalities and also the* $O(\cdot)$ *notation. Without loss of generality let* $\delta := k(\epsilon)\epsilon$*, where* $k : \mathbb{R}^+ \to [1, \infty)$ *is a function of* $\epsilon$*. Therefore,*

$$\begin{aligned}P(\Omega(\delta))\delta+M(1-P(\Omega(\delta))) &= (1 - Kn_x\bar{\beta}\frac{\epsilon}{\delta}\exp(-\bar{\gamma}\frac{\delta^2}{\epsilon^2}))\delta^2 + MKn_x\bar{\beta}\frac{\epsilon}{\delta}\exp(-\bar{\gamma}\frac{\delta^2}{\epsilon^2})\\ &= k^2(\epsilon)\epsilon^2 - Kn_x\bar{\beta}\epsilon^2 k(\epsilon)\exp(-\bar{\gamma}k^2(\epsilon))\\ &\quad + MKn_x\bar{\beta}\frac{\exp(-\bar{\gamma}k^2(\epsilon))}{k(\epsilon)}. \qquad (4.48)\end{aligned}$$

*Now, since in this section we are interested in proving the near-second-order optimality of the provided policy, let us choose the value of* $\frac{\exp(-\bar{\gamma}k^2(\epsilon))}{k(\epsilon)} = O(\epsilon^2)$*. As a result, the second term in* (4.19) *becomes* $O(\epsilon^2)$*, and after determining the function*

*$k(\cdot)$, we test if the order the first term is also $O(\epsilon)$. Now, since in $\frac{1}{k(\epsilon)\exp(\bar{\gamma}k^2(\epsilon))}$, the exponential term finally dominates, we choose* $\exp(-\bar{\gamma}k^2(\epsilon)) = \epsilon^2$ *or by ignoring the $\bar{\gamma}$ constant, $k(\epsilon) = \sqrt{-2\log(\epsilon)}$. Therefore, we choose $\delta := \sqrt{-2\log(\epsilon)}\epsilon$. Now, let us verify that all the three terms in (4.20) are $O(\epsilon^{2-\gamma})$. The calculations for the first term are:*

$$\lim_{\epsilon\downarrow 0}\frac{\delta^2}{\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{k(\epsilon)^2\epsilon^2}{\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{-2\log(\epsilon)\epsilon^2}{\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{-2\log(\epsilon)}{\epsilon^{-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{-2\epsilon^{-1}}{-\gamma\epsilon^{-\gamma-1}}=\frac{2}{\gamma}\lim_{\epsilon\downarrow 0}\epsilon^{\gamma}=0,$$

*where we used the L'Hospital's rule. Hence, $\delta^2 = o(\epsilon^{2-\gamma})$. However, for the sake of this proof, since we want $O(\delta^2)$, we will use the $O(\cdot)$ and $O(\delta^2) = O(\epsilon^{2-\gamma})$. The calculations for the third term are as follows (we ignore the constants in front of the fraction and exponent):*

$$\lim_{\epsilon\downarrow 0}\frac{\exp(-k^2(\epsilon))}{k(\epsilon)\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{\exp(2\log(\epsilon))}{-\log(\epsilon)\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{\epsilon^2}{-\log(\epsilon)\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{\epsilon^{\gamma}}{-\log(\epsilon)}=0.$$

*Therefore, the third term is also at least $O(\epsilon^{2-\gamma})$. In fact, this term is $o(\epsilon^2)$ (verified by setting $\gamma$ to zero); however, since, the bottle neck is the first term, we can just replace it with $O(\epsilon^{2-\gamma})$. The second term consists of the third term times the first term (ignoring the constants). Therefore, this term also is at least $O(\epsilon^{2-\gamma})$. As a result, we have $\mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma})$ and the other statements hold, as well.*

*Remark:* Note that choosing $k(\epsilon) = \sqrt{-r\log(\epsilon)}$ for $r \geq 2$ still leads to the $O(\epsilon^{2-\gamma})$-optimality. This is due to the fact that the calculations for the first term become

$$\lim_{\epsilon\downarrow 0}\frac{k(\epsilon)^2\epsilon^2}{\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{-r\log(\epsilon)\epsilon^2}{\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{-r\log(\epsilon)}{\epsilon^{-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{-r\epsilon^{-1}}{-\gamma\epsilon^{-\gamma-1}}=\frac{r}{\gamma}\lim_{\epsilon\downarrow 0}\epsilon^{\gamma}=0,$$

and the calculations of the third term become:

$$\lim_{\epsilon\downarrow 0}\frac{\exp(-k^2(\epsilon))}{k(\epsilon)\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{\exp(r\log(\epsilon))}{-\log(\epsilon)\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{\epsilon^r}{-\log(\epsilon)\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{\epsilon^{r-2-\gamma}}{-\log(\epsilon)}=0,$$

where $r-2-\gamma>0$. However, it means that the tube gets larger which is less desirable due to the fact that smaller tube translates to more accuracy. On the other hand, choosing $1<r<2$ results in $O(\epsilon^{1-\gamma})$-optimality due to the fact that the third term's calculations become:

$$\begin{aligned}\lim_{\epsilon\downarrow 0}\frac{\exp(r\log(\epsilon))}{-\log(\epsilon)\epsilon^{2-\gamma}}&=\lim_{\epsilon\downarrow 0}\frac{\epsilon^r}{-\log(\epsilon)\epsilon^{2-\gamma}}=\lim_{\epsilon\downarrow 0}\frac{\epsilon^{\gamma+r-2}}{-\log(\epsilon)}=\lim_{\epsilon\downarrow 0}\frac{(\gamma+r-2)\epsilon^{\gamma+r-3}}{-(\epsilon)^{-1}}\\&=\lim_{\epsilon\downarrow 0}\frac{(\gamma+r-2)}{-\epsilon^{3-\gamma-r}}=\infty.\end{aligned}$$

where $\gamma+r-2<0$ since $0<\gamma\ll 1$.

*Remark:* Note that the third term of (4.48) is in fact the probability $1-P(\Omega(\delta))$. As we mentioned, choosing $k(\epsilon)=\sqrt{-r\log(\epsilon)}$ with $r>2$ still works for the purpose of our proof, and in fact by doing so, $1-P(\Omega(\delta))$ becomes even larger, which is also intuitive due to the fact that a larger tube yields a lower probability of exiting that tube, as well. Similar to above, we can how that choosing $r>2$ yields the second term become $o(\epsilon^r)$ as $\epsilon\downarrow 0$. However, as mentioned in the previous remark, the smallest value of $r$ that works for second-order optimality (for this form of $\delta$) is $r=2$, and in fact this smallest value is more desirable. Last, note that similar arguments can be made for the near-first-order optimality case of Section 4.4 (i.e., in that situation $r>1$ would work but result in a larger tube).

Therefore, when the functions are in $\mathbb{C}^2$, the expected stochastic cost is equal to the nominal cost with a higher probability as $\epsilon\downarrow 0$. Therefore, it follows that the decoupling principle holds with a higher probability, summarized below:

**Corollary 3 Near-Second-Order Optimality of the Deterministic Optimal Policy for the Stochastic Fully-Observed System Under Small Noise.** *Based on Theorem 4, for a fully-observed system under the small noise paradigm, as $\epsilon \downarrow 0$, the deterministic optimal control law becomes $O(\epsilon^{2-\gamma})$-optimal with some $0 < \gamma \ll 1$ for the stochastic problem.*

***Proof 10*** *Using Theorem 4, for $\omega \in \Omega(\delta)$ we have $\mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma})$, which is the cost of applying policy $\boldsymbol{\pi}^d$ to the stochastic system. Now, suppose $\boldsymbol{\pi}^*$ is the optimal stochastic policy. By assumption $\boldsymbol{\pi}^*$ is in $\mathbb{C}^2$. Therefore, by modifying the definition of $\mathbf{L}_t$ as $\mathbf{L}_t = -\nabla_{\mathbf{x}}\boldsymbol{\pi}_t^*(\mathbf{x})|_{\mathbf{x}_t^{*p}}$ and modifying $\mathbf{H}_t^{\pi^k}$ as $\mathbf{H}_t^{\pi^k} := \frac{1}{2}\nabla^2_{\mathbf{xx}}\boldsymbol{\pi}_t^{*k}(\mathbf{x})|_{\mathbf{x}_t^p}$, defining $\mathbf{u}_t^{*p} = \boldsymbol{\pi}_t^*(\mathbf{x}_t^{*p})$ and replacing $p$ with $*p$ in (4.23), we have $\boldsymbol{\pi}_t^*(\mathbf{x}_t) = \mathbf{u}_t^{*p} - \mathbf{L}_t\tilde{\mathbf{x}}_t + \sum_{k=1}^{n_u}(\tilde{\mathbf{x}}_t^T\mathbf{H}_t^{\pi^k}\tilde{\mathbf{x}}_t)\mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}\|_\infty^2)$ (where $\tilde{\mathbf{x}}$ is also modified to denote $(\mathbf{x}_t - \mathbf{x}_t^{*p})$). Similarly, by using appropriate modifications the entire calculations of this section hold for this policy. Hence, using Theorem 2 for this system, the cost function of policy $\boldsymbol{\pi}^*$ can be written as $\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{2-\gamma})$. Now, by construction $J^p \leq J^{*p}$, and*

$$\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{2-\gamma}) \geq J^p + O(\epsilon^{2-\gamma}) = \mathbb{E}[J_{\boldsymbol{\pi}^d}] + O(\epsilon^{2-\gamma})$$

*As a result, policy $\boldsymbol{\pi}^d$ is within $O(\epsilon^{2-\gamma})$ of the optimal stochastic policy.*

## 4.8 Near-Second-Order Optimality of T-LQR

In this section, we provide a near-second-order analysis and show that the results of the previous sections are also second-order optimal.

*Assumptions:* Other than the assumptions of previous sections, we assume for the analysis of this section that all the functions are in $\mathbb{C}^2$, i.e., they are continuously differentiable to the second order.

*Second-order expansion of the dynamics:* Let us first obtain the second-order expansion of the process model around the nominal trajectory. Then for $0 \leq t \leq K-1$:

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{G}_t\mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,1}_{\mathbf{xx}} & \mathbf{F}^{t,1}_{\mathbf{xu}} \\ \mathbf{F}^{t,1}_{\mathbf{ux}} & \mathbf{F}^{t,1}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,n_x}_{\mathbf{xx}} & \mathbf{F}^{t,n_x}_{\mathbf{xu}} \\ \mathbf{F}^{t,n_x}_{\mathbf{ux}} & \mathbf{F}^{t,n_x}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}_t\|^2 + \|\tilde{\mathbf{u}}_t\|^2) \tag{4.49a}$$

$$= \mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{G}_t\mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,1}_{\mathbf{xx}} & \mathbf{F}^{t,1}_{\mathbf{xu}} \\ \mathbf{F}^{t,1}_{\mathbf{ux}} & \mathbf{F}^{t,1}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,n_x}_{\mathbf{xx}} & \mathbf{F}^{t,n_x}_{\mathbf{xu}} \\ \mathbf{F}^{t,n_x}_{\mathbf{ux}} & \mathbf{F}^{t,n_x}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|^2_\infty + \|\tilde{\mathbf{u}}\|^2_\infty), \tag{4.49b}$$

as $(\|\tilde{\mathbf{x}}\|^2_\infty + \|\tilde{\mathbf{u}}\|^2_\infty) \downarrow 0$, where we have:

- $\mathbf{A}_t := \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x},\mathbf{u})|_{\mathbf{x}^p_t,\mathbf{u}^p_t}$, $\mathbf{B}_t := \nabla_{\mathbf{u}}\mathbf{f}(\mathbf{x},\mathbf{u})|_{\mathbf{x}^p_t,\mathbf{u}^p_t}$, $\mathbf{G}_t := \epsilon\boldsymbol{\sigma}_t$;
- $\mathbf{f}(\mathbf{x},\mathbf{u}) = (f^j(\mathbf{x},\mathbf{u})), 1 \leq j \leq n_x$;
- $\mathbf{F}^{t,j}_{\mathbf{xx}} := \frac{1}{2}\nabla^2_{\mathbf{xx}}f^j(\mathbf{x},\mathbf{u})|_{\mathbf{x}^p_t,\mathbf{u}^p_t}, \mathbf{F}^{t,j}_{\mathbf{xu}} := \frac{1}{2}\nabla^2_{\mathbf{xu}}f^j(\mathbf{x},\mathbf{u})|_{\mathbf{x}^p_t,\mathbf{u}^p_t}, \mathbf{F}^{t,j}_{\mathbf{ux}} := \frac{1}{2}\nabla^2_{\mathbf{ux}}f^j(\mathbf{x},\mathbf{u})|_{\mathbf{x}^p_t,\mathbf{u}^p_t}$, and $\mathbf{F}^{t,j}_{\mathbf{uu}} := \frac{1}{2}\nabla^2_{\mathbf{uu}}f^j(\mathbf{x},\mathbf{u})|_{\mathbf{x}^p_t,\mathbf{u}^p_t}$;
- $\tilde{\mathbf{u}}_0 = \mathbf{u}_0 - \mathbf{u}^p_0 = \mathbf{0}$, and $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}^p_0 = \mathbf{0}$.

Now, we apply the T-LQR feedback law $\tilde{\mathbf{u}}_t = -\mathbf{L}_t\tilde{\mathbf{x}}_t$ to the above equations:

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{A}_t\tilde{\mathbf{x}}_t - \mathbf{B}_t\mathbf{L}_t\tilde{\mathbf{x}}_t + \mathbf{G}_t\mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\mathbf{L}_t\tilde{\mathbf{x}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,1}_{\mathbf{xx}} & \mathbf{F}^{t,1}_{\mathbf{xu}} \\ \mathbf{F}^{t,1}_{\mathbf{ux}} & \mathbf{F}^{t,1}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\mathbf{L}_t\tilde{\mathbf{x}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\mathbf{L}_t\tilde{\mathbf{x}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,n_x}_{\mathbf{xx}} & \mathbf{F}^{t,n_x}_{\mathbf{xu}} \\ \mathbf{F}^{t,n_x}_{\mathbf{ux}} & \mathbf{F}^{t,n_x}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\mathbf{L}_t\tilde{\mathbf{x}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \tag{4.50a}$$

$$= \mathbf{D}_t\tilde{\mathbf{x}}_t + \mathbf{G}_t\mathbf{w}_t + \begin{pmatrix} \tilde{\mathbf{x}}_t^T\mathbf{H}_t^{f^1}\tilde{\mathbf{x}}_t \\ \vdots \\ \tilde{\mathbf{x}}_t^T\mathbf{H}_t^{f^{n_x}}\tilde{\mathbf{x}}_t \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \tag{4.50b}$$

$$= \mathbf{D}_t\tilde{\mathbf{x}}_t + \mathbf{G}_t\mathbf{w}_t + \sum_{j=1}^{n_x}(\tilde{\mathbf{x}}_t^T\mathbf{H}_t^{f^j}\tilde{\mathbf{x}}_t)\mathbf{e}_j^{n_x} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \tag{4.50c}$$

$$= \tilde{\mathbf{D}}_{0:t}\tilde{\mathbf{x}}_0 + \sum_{r=0}^{t}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + \sum_{r=0}^{t}\tilde{\mathbf{D}}_{r+1:t}\sum_{j=1}^{n_x}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{f^j}\tilde{\mathbf{x}}_r)\mathbf{e}_j^{n_x} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \tag{4.50d}$$

$$= \sum_{r=0}^{t}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + \sum_{r=0}^{t}\sum_{j=1}^{n_x}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{f^j}\tilde{\mathbf{x}}_r)\tilde{\mathbf{D}}_{r+1:t}\mathbf{e}_j^{n_x} + o(\|\tilde{\mathbf{x}}\|_\infty^2), \tag{4.50e}$$

where $\mathbf{H}_t^{f^j} := (\mathbf{F}^{t,j}_{\mathbf{xx}} - \mathbf{F}^{t,j}_{\mathbf{xu}}\mathbf{L}_t - \mathbf{L}_t^T\mathbf{F}^{t,j}_{\mathbf{ux}} + \mathbf{L}_t^T\mathbf{F}^{t,j}_{\mathbf{uu}}\mathbf{L}_t)$, for $1 \leq j \leq n_x$. Note, based on the $\mathbb{C}^2$ assumption for the dynamics, $\mathbf{F}^{t,j}_{\mathbf{xu}} = \mathbf{F}^{t,j}_{\mathbf{ux}}$.

*Validity region:* Note that the definition of $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}_t^p$. Therefore, the properties of $O(\|\tilde{\mathbf{x}}_t\|_\infty)$ that we have proven in Section 4.5 for the T-LQR feedback design still hold for the above linearization, as well. Particularly, we proved that for a T-LQR design, $O(\|\tilde{\mathbf{x}}_t\|_\infty) = O(\delta)$ in a set $\Omega(\delta)$ properly defined as before with probability $1 - o(\epsilon)$. Hence, for $\omega \in \Omega(\delta)$, $O(\|\tilde{\mathbf{x}}_t\|_\infty^2) = O(\delta^2)$. Thus, for $\omega \in \Omega(\delta)$, we have:

$$\tilde{\mathbf{x}}_{t+1} = \sum_{r=0}^{t}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + \sum_{r=0}^{t}\sum_{j=1}^{n_x}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{f^j}\tilde{\mathbf{x}}_r)\tilde{\mathbf{D}}_{r+1:t}\mathbf{e}_j + O(\delta^2). \tag{4.51}$$

*Second-order expansion of the cost function:* Similarly, we obtain the second-order Taylor series expansion of the cost function around the nominal trajectory:

$$J = J^p + \tilde{J}_1 + \tilde{J}_2 + o(\sum_{t=1}^{K-1}(\|\tilde{\mathbf{x}}_t\|^2 + \|\tilde{\mathbf{u}}_t\|^2) + \|\tilde{\mathbf{x}}_K\|^2) \tag{4.52a}$$

$$= J^p + \tilde{J}_1 + \tilde{J}_2 + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2), \tag{4.52b}$$

as $(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2) \downarrow 0$. Moreover, we have:

- $J^p := \sum_{t=0}^{K-1} c_t(\mathbf{x}_t^p, \mathbf{u}_t^p) + c_K(\mathbf{x}_K^p)$ denotes the nominal cost;
- $\tilde{J}_1 := \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t + \mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t) + \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K$ is the first order cost error;
- $\tilde{J}_2 := \sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xx}}\tilde{\mathbf{x}}_t + \frac{1}{2}\tilde{\mathbf{u}}_t^T\mathbf{C}_t^{\mathbf{uu}}\tilde{\mathbf{u}}_t + \tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xu}}\tilde{\mathbf{u}}_t) + \frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}_K^{\mathbf{xx}}\tilde{\mathbf{x}}_K$ is the second order cost error.
- $J_2 := J^p + \tilde{J}_1 + \tilde{J}_2$ is the second order approximation of the cost function;
- $\mathbf{C}_t^{\mathbf{xx}} = \nabla^2_{\mathbf{xx}} c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{uu}} = \nabla^2_{\mathbf{uu}} c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{xu}} = \nabla^2_{\mathbf{xu}} c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, and $\mathbf{C}_K^{\mathbf{xx}} = \nabla^2_{\mathbf{xx}} c_K(\mathbf{x})|_{\mathbf{x}_K^p}$, where we have used the fact that $c_t \in \mathbb{C}^2$.

Next, we provide the main result regarding the expected second order error of the cost function.

**Theorem 5 Second-Order Cost Function Error for a Fully-Observed System with T-LQR Policy:** *Given that process noises are zero mean i.i.d. Gaussian and all the functions are in $\mathbb{C}^2$, under a first-order approximation for the small noise paradigm, the stochastic cost function is dominated by the nominal part of the cost function, and the expected first-order error is $O(\delta^2)$. That is,*

$$\mathbb{E}[\tilde{J}_1] = O(\delta^2), \text{ and } \mathbb{E}[J] = J^p + O(\delta^2).$$

*Moreover, by choosing $\delta = \sqrt{2\log(\frac{1}{\epsilon})}\epsilon$, we have*

$$\mathbb{E}[\tilde{J}_1] = O(\epsilon^{2-\gamma}), \text{ and } \mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma}),$$

*for some $0 < \gamma \ll 1$, which shows that this error tends to zero with a near-first-order rate as $\epsilon \downarrow 0$.*

***Proof 11*** *First, let us simply the first order cost error:*

$$\begin{aligned}
\tilde{J}_1 &= \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t + \mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t) + \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K = \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t - \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_t\tilde{\mathbf{x}}_t) + \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K \\
&= \sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{x}}_t \\
&= \sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + \sum_{t=0}^{K}\sum_{r=0}^{t-1}\sum_{j=1}^{n_x}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{fj}\tilde{\mathbf{x}}_r)\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t-1}\mathbf{e}_j^{n_x} + o(\|\tilde{\mathbf{x}}\|_\infty^2),
\end{aligned}$$

*where $\mathbf{C}_t^{\mathbf{L}} := \mathbf{C}_t^{\mathbf{x}} - \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_t, 0 \leq t \leq K-1$, and $\mathbf{C}_K^{\mathbf{L}} = \mathbf{C}_K^{\mathbf{x}}$.*

*Next, we simplify the second order cost error:*

$$\begin{aligned}
\tilde{J}_2 &= \sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xx}}\tilde{\mathbf{x}}_t + \frac{1}{2}\tilde{\mathbf{u}}_t^T\mathbf{C}_t^{\mathbf{uu}}\tilde{\mathbf{u}}_t + \tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xu}}\tilde{\mathbf{u}}_t) + \frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}_K^{\mathbf{xx}}\tilde{\mathbf{x}}_K \\
&= \sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xx}}\tilde{\mathbf{x}}_t + \frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{L}_t^T\mathbf{C}_t^{\mathbf{uu}}\mathbf{L}_t\tilde{\mathbf{x}}_t + \tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xu}}\mathbf{L}_t\tilde{\mathbf{x}}_t) + \frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}_K^{\mathbf{xx}}\tilde{\mathbf{x}}_K \\
&= \sum_{t=0}^{K}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{LL}}\tilde{\mathbf{x}}_t,
\end{aligned}$$

*where $\mathbf{C}_t^{\mathbf{LL}} := \frac{1}{2}\mathbf{C}_t^{\mathbf{xx}} + \frac{1}{2}\mathbf{L}_t^T\mathbf{C}_t^{\mathbf{uu}}\mathbf{L}_t + \mathbf{C}_t^{\mathbf{xu}}\mathbf{L}_t, 0 \leq t \leq K-1$, and $\mathbf{C}_K^{\mathbf{LL}} := \frac{1}{2}\mathbf{C}_K^{\mathbf{xx}}$. Hence, we have:*

$$\begin{aligned}
\tilde{J}_1 + \tilde{J}_2 =& \sum_{t=0}^{K}\sum_{r=0}^{t-1}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + \sum_{t=0}^{K}\sum_{r=0}^{t-1}\sum_{j=1}^{n_x}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{fj}\tilde{\mathbf{x}}_r)\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{D}}_{r+1:t-1}\mathbf{e}_j^{n_x} \\
&+ \sum_{t=0}^{K}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{LL}}\tilde{\mathbf{x}}_t + o(\|\tilde{\mathbf{x}}\|_\infty^2)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{t=0}^{K}\sum_{r=0}^{t-1} \mathbf{C}_t^{\mathbf{L}} \tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + O(\|\tilde{\mathbf{x}}\|_\infty^2) + o(\|\tilde{\mathbf{x}}\|_\infty^2)\\
&= \sum_{t=0}^{K}\sum_{r=0}^{t-1} \mathbf{C}_t^{\mathbf{L}} \tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + O(\|\tilde{\mathbf{x}}\|_\infty^2).
\end{aligned}
$$

*Hence, for* $\omega \in \Omega(\delta)$*,*

$$
\tilde{J}_1 + \tilde{J}_2 = \sum_{t=0}^{K}\sum_{r=0}^{t-1} \mathbf{C}_t^{\mathbf{L}} \tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + O(\delta^2).
$$

*As a result, using* (4.52b)*, for* $\omega \in \Omega(\delta)$*, we have:*

$$
J = J^p + \sum_{t=0}^{K}\sum_{r=0}^{t-1} \mathbf{C}_t^{\mathbf{L}} \tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r + O(\delta^2).
$$

*Next, note that* $\mathbb{E}[\mathbf{w}_t] = 0$ *for all* $t$*, and*

$$
\mathbb{E}[\sum_{t=0}^{K}\sum_{r=0}^{t-1} \mathbf{C}_t^{\mathbf{L}} \tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r] = \sum_{t=0}^{K}\sum_{r=0}^{t-1} \mathbf{C}_t^{\mathbf{L}} \tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbb{E}[\mathbf{w}_r] = 0.
$$

*Now, since for* $\omega \notin \Omega(\delta)$*,* $J \leq M$*, then*

$$
\begin{aligned}
\mathbb{E}[J - J^p] &= P(\Omega(\delta))(\mathbb{E}[\sum_{t=0}^{K}\sum_{r=0}^{t-1} \mathbf{C}_t^{\mathbf{L}} \tilde{\mathbf{D}}_{r+1:t}\mathbf{G}_r\mathbf{w}_r] + O(\delta^2)) + M(1 - P(\Omega(\delta)))\\
&= P(\Omega(\delta))O(\delta) + M(1 - P(\Omega(\delta))) \qquad (4.53)
\end{aligned}
$$

*Note the last expression is the same as* (4.47)*. Although* $\Omega(\delta)$ *is not the same as in Theorem 4,* $P(\Omega(\delta))$ *is still the same. In the proof of Theorem 4 while we discussed on the probabilistic argument and choosing the proper* $\delta$*, we showed that by choosing* $\delta := \sqrt{-2\log(\epsilon)}\epsilon$*, the* $\mathbb{E}[J - J^p] = O(\epsilon^{2-\gamma})$*. The same argument follows through and this theorem is proved.*

Hence, when the functions are in $\mathbb{C}^2$, the expected stochastic cost is equal to the nominal cost with a higher probability as $\epsilon \downarrow 0$. Therefore, it follows that the decoupling principle holds with a higher probability, summarized below:

**Corollary 4 Decoupling Principle: Near-Second-Order Optimality for a Fully-Observed System.** *Based on Theorem 5, for a fully-observed system where the function are in $\mathbb{C}^2$ under the small noise paradigm, as $\epsilon \downarrow 0$, the decoupling principle holds with $O(\epsilon^{2-\gamma})$-optimality for $0 < \gamma \ll 1$. Moreover, the T-LQR approach is $O(\epsilon^{2-\gamma})$-optimal.*

***Proof 12*** *In the proof of Corollary 3 we showed that $\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{2-\gamma})$. Now, by construction $J^p \leq J^{*p}$, where $J^p$ is the nominal cost for the T-LQR policy (or in fact any other smooth linear feedback law). Hence, using Theorem 5 for this system,*

$$\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{2-\gamma}) \geq J^p + O(\epsilon^{2-\gamma}) = \mathbb{E}[J_{\boldsymbol{\pi}}] + O(\epsilon^{2-\gamma})$$

*As a result, policy $\boldsymbol{\pi}$ is within $O(\epsilon^{2-\gamma})$ of the optimal stochastic policy.*

# 5. FULLY-OBSERVED MULTI-AGENT SYSTEM

In this chapter, we extend the results of Chapter 4 to a multi-agent scenario. For multi-agent robotic systems the solution should not require a fully centralized control since that would require pervasive constant communication among all robots. We establish the decoupling of feedback for different agents, which leads us to a decentralized solution with no communication requirements during the execution, for small noise levels.

## 5.1 Multi-Agent Decoupling of Open-Loop and Closed-Loop Designs

In this section, we generalize the single-agent results of Result 2 for a multi-agent fully-observed system. The generalization is straightforward by noting the fact that a centralized multi-agent can be considered as one big single-agent system by defining appropriate concatenations of the variables.

*One joint system:* First, we concatenate the equations of control and state evolutions for all agents and consider the entire multi-agent system as one system. The vectors with index $\mathcal{I}$ are formed by just concatenating them in one column, whereas all the matrices are formed by concatenating them in a block matrix, unless otherwise stated. For instance, $\mathbf{w}_t^{\mathcal{I}}$, $\mathbf{f}^{\mathcal{I}}$, and $\mathbf{B}_t^{\mathcal{I}}$ are as follows:

$$\mathbf{w}_t^{\mathcal{I}} := \begin{pmatrix} \mathbf{w}_t^1 \\ \vdots \\ \mathbf{w}_t^m \end{pmatrix}, \mathbf{f}^{\mathcal{I}}(\mathbf{x}_t^{\mathcal{I}}, \mathbf{u}_t^{\mathcal{I}}) := \begin{pmatrix} \mathbf{f}^1(\mathbf{x}_t^1, \mathbf{u}_t^1) \\ \vdots \\ \mathbf{f}^m(\mathbf{x}_t^m, \mathbf{u}_t^m) \end{pmatrix}, \mathbf{B}_t^{\mathcal{I}} := \begin{pmatrix} \mathbf{B}_t^1 & & \\ & \ddots & \\ & & \mathbf{B}_t^m \end{pmatrix}. \qquad (5.1)$$

Note that some of these matrices are diagonal, e.g., $\mathbf{A}_t^{\mathcal{I}}$, and the others may or may not be, depending on the state, control and observation, or other dimensions. Now, if functions are in $\mathbb{C}^1$, we can simply utilize the single-agent form of Corollary 2 for the joint single agent system with index $\mathcal{I}$, which generalizes the result for the multi-agent system. Moreover, if functions are in $\mathbb{C}^2$, we utilize Corollary 4 to obtain the near-second-order optimality of the design scheme. Therefore,

$$\mathbf{x}_{t+1}^{\mathcal{I}} = \mathbf{f}^{\mathcal{I}}(\mathbf{x}_t^{\mathcal{I}}, \mathbf{u}_t^{\mathcal{I}}) + \epsilon \boldsymbol{\sigma}^{\mathbf{f}_{\mathcal{I}}}(t)\mathbf{w}_t^{\mathcal{I}}. \tag{5.2}$$

**Remark 1** *Corollary 2 states that for the multi-agent system of* (5.2) *with index $\mathcal{I}$, if functions are in $\mathbb{C}^1$ the first order approximation of the cost function does not depend on the linear feedback gain; rather, it is completely determined by the nominal trajectory. Moreover, if functions are in $\mathbb{C}^2$, based on Corollary 4 the second-order approximation of the cost function is also dominated by the nominal cost. This leads to the extension of the decoupling of open-loop/closed-loop designs for a multi-agent fully-observed system. That is, under small noise, the multi-agent version of problem* (2) *can be optimally separated into two problems: i) an open-loop optimal control problem to design the nominal trajectories of the system, and ii) a design of the optimal feedback law to track the nominal trajectories of the system.*

**Problem 9 (Nominal Trajectory Design Problem)** *Given an initial joint state $\mathbf{x}_0^{\mathcal{I}}$, solve:*

$$\min_{\mathbf{u}_{0:K_{\mathcal{I}}-1}^{\mathcal{I}}} \sum_{t=0}^{K_{\mathcal{I}}-1} c_t(\mathbf{x}_t^{\mathcal{I}}, \mathbf{u}_t^{\mathcal{I}}) + c_{K_{\mathcal{I}}}(\mathbf{x}_{K_{\mathcal{I}}}^{\mathcal{I}})]$$

$$s.t.\ \mathbf{x}_{t+1}^{\mathcal{I}} = \mathbf{f}^{\mathcal{I}}(\mathbf{x}_t^{\mathcal{I}}, \mathbf{u}_t^{\mathcal{I}}).$$

*Nominal trajectories:* Given the initial state $\mathbf{x}_0^{\mathcal{I}}$, and using the optimized nominal controls of the above problem, $\mathbf{u}_t^{p\mathcal{I}}$, the nominal trajectory of the multi-agent system is defined as:

$$\mathbf{x}_{t+1}^{p_{\mathcal{I}}} = \mathbf{f}^{\mathcal{I}}(\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}), \tag{5.3}$$

where $\mathbf{x}_0^{p_{\mathcal{I}}} := \mathbf{x}_0^{\mathcal{I}}$, and $\mathbf{x}_{t+1}^{p_i} = \mathbf{f}^i(\mathbf{x}_t^{p_i}, \mathbf{u}_t^{p_i})$ for $i \in \mathcal{I}$.

*Linearized system:* Using the result of the previous chapter and using a feedback policy for each agent that depends on the entire system's state, we can write the linearized system for each agent as:

$$\begin{aligned}
\mathbf{x}_{t+1}^{\mathcal{I}} &= \mathbf{x}_{t+1}^{p_{\mathcal{I}}} + \mathbf{A}_t^{\mathcal{I}}(\mathbf{x}_t^{\mathcal{I}} - \mathbf{x}_t^{p_{\mathcal{I}}}) + \mathbf{B}_t^{\mathcal{I}}(\mathbf{u}_t^{\mathcal{I}} - \mathbf{u}_t^{p_{\mathcal{I}}}) + \mathbf{G}_t^{\mathcal{I}}\mathbf{w}_t^{\mathcal{I}} + O(\delta),\\
J_{\boldsymbol{\pi}} &= J^p + \tilde{J}_1 + O(\delta),\\
\tilde{J}_1 &:= \sum_{t=0}^{K_{\mathcal{I}}-1} [\mathbf{C}_t^{\mathbf{x}_{\mathcal{I}}}(\mathbf{x}_t^{\mathcal{I}} - \mathbf{x}_t^{p_{\mathcal{I}}}) + \mathbf{C}_t^{\mathbf{u}_{\mathcal{I}}}(\mathbf{u}_t^{\mathcal{I}} - \mathbf{u}_t^{p_{\mathcal{I}}})] + \mathbf{C}_{K_{\mathcal{I}}}^{\mathbf{x}_{\mathcal{I}}}(\mathbf{x}_{K_{\mathcal{I}}}^{\mathcal{I}} - \mathbf{x}_{K_{\mathcal{I}}}^{p_{\mathcal{I}}}),\\
J^p &:= \sum_{t=0}^{K_{\mathcal{I}}-1} c_t(\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}) + c_{K_{\mathcal{I}}}(\mathbf{x}_{K_{\mathcal{I}}}^{p_{\mathcal{I}}}).
\end{aligned}$$

The Jacobians are:

$$\begin{aligned}
&\mathbf{A}_t^{\mathcal{I}} := \nabla_{\mathbf{x}}\mathbf{f}^{\mathcal{I}}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}}, \mathbf{B}_t^{\mathcal{I}} := \nabla_{\mathbf{u}}\mathbf{f}^{\mathcal{I}}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}}, \mathbf{G}_t^{\mathcal{I}} := \epsilon\boldsymbol{\sigma}^{\mathbf{f}_{\mathcal{I}}}(t),\\
&\mathbf{C}_t^{\mathbf{x}_{\mathcal{I}}} := \nabla_{\mathbf{x}} c_t^{\boldsymbol{\pi}^{\mathcal{I}}}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}}, \mathbf{C}_{K_{\mathcal{I}}}^{\mathbf{x}_{\mathcal{I}}} := \nabla_{\mathbf{x}} c_{K_{\mathcal{I}}}^{\boldsymbol{\pi}^{\mathcal{I}}}(\mathbf{x})|_{\mathbf{x}_t^{p_{\mathcal{I}}}}, \mathbf{C}_t^{\mathbf{u}_{\mathcal{I}}} := \nabla_{\mathbf{u}} c_t^{\boldsymbol{\pi}^{\mathcal{I}}}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}},
\end{aligned}$$

and $\mathbf{u}_t^{\mathcal{I}}$ is obtained by deigning a optimal LQR feedback policy to track the joint nominal trajectory $\mathbf{x}_t^{p_{\mathcal{I}}}$ as:

$$\mathbf{u}_t^{\mathcal{I}} = \mathbf{u}_t^{p_{\mathcal{I}}} - \mathbf{L}_t^{\mathcal{I}}(\mathbf{x}_t^{\mathcal{I}} - \mathbf{x}_t^{p_{\mathcal{I}}}), \tag{5.4}$$

$$\mathbf{L}_t^{\mathcal{I}} = (\mathbf{W}_t^{u_{\mathcal{I}}} + (\mathbf{B}_t^{\mathcal{I}})^T\mathbf{S}_t^{\mathcal{I}}\mathbf{B}_t^{\mathcal{I}})^{-1}(\mathbf{B}_t^{\mathcal{I}})^T\mathbf{S}_t^{\mathcal{I}}\mathbf{A}_t^{\mathcal{I}}. \tag{5.5}$$

$\mathbf{S}_t^{\mathcal{I}}$ is obtained using the backward dynamic Riccati equation with $\mathbf{S}_{K_{\mathcal{I}}}^{\mathcal{I}} = \mathbf{W}_{K_{\mathcal{I}}}^{x_{\mathcal{I}}}$:

$$\mathbf{S}_{t-1}^{\mathcal{I}} = (\mathbf{A}_t^{\mathcal{I}})^T \mathbf{S}_t^{\mathcal{I}} \mathbf{A}_t^{\mathcal{I}} - (\mathbf{A}_t^{\mathcal{I}})^T \mathbf{S}_t^{\mathcal{I}} \mathbf{B}_t^{\mathcal{I}} \mathbf{L}_t^{\mathcal{I}} + \mathbf{W}_t^{x_{\mathcal{I}}}, \tag{5.6}$$

where $\mathbf{W}_t^{x_{\mathcal{I}}} \succeq 0$ and $\mathbf{W}_t^{u_{\mathcal{I}}} \succeq 0$ are two block-diagonal positive semi-definite weight matrices (with blocks of $\mathbf{W}_t^{x_i} \succeq 0$ of dimension $n_x^i \times n_x^i$ and $\mathbf{W}_t^{u_i} \succeq 0$ of dimension $n_u^i \times n_u^i$, respectively).

*Structure of feedback:* As shown above, $\mathbf{L}_t^{\mathcal{I}}$ that is designed using the single-agent decoupling principle depends on the entire state. Next, we will analyze the structure of feedback and prove the multi-agent decoupling principle.

*Remark:* Although, we have shown the first-order linearizations above, it should be noted that the second-order linearizations also follows similarly with proper indexing of the single-agent variables. Therefore, we avoid repeating them. Nevertheless, for the rest of the proof only first-order variables suffice.

## 5.2 Decoupling of Feedback Designs

**Proof 13 (proof of Result 2 for a multi-agent system)** *Note* $\mathbf{A}_t^{\mathcal{I}}, \mathbf{B}_t^{\mathcal{I}}, \mathbf{W}_t^{x_{\mathcal{I}}}, \mathbf{W}_t^{u_{\mathcal{I}}}$ *and* $\mathbf{S}_{K_{\mathcal{I}}}^{\mathcal{I}}$ *are block matrices, which goes back to the independent dynamics assumption in* (3.2)*, and* $(\mathbf{W}_t^{u_{\mathcal{I}}} + (\mathbf{B}_t^{\mathcal{I}})^T \mathbf{S}_t^{\mathcal{I}} \mathbf{B}_t^{\mathcal{I}})$ *is a block-diagonal square matrix. Since the operations of matrix summation, multiplication of block matrices and inverse of square block-diagonal matrices preserve the block structure,* $\mathbf{S}_t^{\mathcal{I}}, t \geq 0$ *is a block-diagonal square matrix with blocks of* $n_x^i \times n_x^i$ *dimensions, and can be written as* $\mathbf{S}_t^{\mathcal{I}} := \text{blockdiag}(\mathbf{S}_t^1, \cdots, \mathbf{S}_t^m)$*. Importantly, no element of agent j's variable is involved in the calculation of the* $\mathbf{S}_t^i$ *block for* $j \neq i$*. Thus, the* $mn_u^i \times mn_x^i$*-dimensional matrix* $\mathbf{L}_t^{\mathcal{I}}$ *is a block diagonal matrix, as well. Also, the ith block of the* $\mathbf{L}_t^{\mathcal{I}}$ *(denoted by* $\mathbf{L}_t^i$ *which is an* $n_u^i \times mn_x^i$*-dimensional matrix) consists of non-zero elements only in its* $n_u^i \times n_x^i$*-th block. Further, no element of agent j's variable is involved in the* *calculation of these non-zero elements for $i \neq j$.*

**Remark 2** *Result 2 proves that under the conditions of Result 1 and the independence of the dynamics, the feedback gain of the agent $i$ can be optimally calculated decoupled from the agent $j$, and states that the Riccati equation of* (5.6) *breaks up into $m$ separate Riccati equations. As a result, the dimension of the optimal linear feedback gain for agent $i$ reduces to $n_u^i \times n_x^i$, which is the same as an LQR design to track the fully-observed nominal state of agent $i$. This design leads to a decentralized multi-agent planning approach of Multi-agent T-LQR (MT-LQR), which is near-second-order optimal as $\epsilon \downarrow 0$, and is elaborated next.*

**Remark 3** *Note that in a multi-agent scenario, the cost functions may have a shared cost such as inter-agent collision. In that situation, with a careful design of the nominal trajectory, the shared cost is taken into account in the nominal trajectory design stage with sufficient safety margins such that within the $\delta$ tubes of the agents, the shared cost vanishes to zero. Therefore, the feedback design for each agent becomes the LQG tracking problem within a tube without considering the shared cost. This is addressed in more details for the general partially-observed situation in Chapter 9.*

## 5.3 MT-LQR: Multi-agent Trajectory-optimized LQR

The design approach resulting from Result 2 for a multi-agent system with full state information consists of two steps. The first step is to solve the joint nominal trajectory design problem. The second step is to design $m$ LQR trackers one for each of the agents, separately.

**Problem 10 (MT-LQR Nominal Trajectory Design Problem)** *Given an ini-*

*tial joint state* $\mathbf{x}_0^{\mathcal{I}} =: \mathbf{x}_0^{p_{\mathcal{I}}}$*, solve:*

$$\min_{\mathbf{u}_{0:K_{\mathcal{I}}-1}^{p_{\mathcal{I}}}} \mathbb{E}[\sum_{t=0}^{K_{\mathcal{I}}-1} c_t(\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}) + c_{K_{\mathcal{I}}}(\mathbf{x}_{K_{\mathcal{I}}}^{p_{\mathcal{I}}})]$$

$$s.t.\ \mathbf{x}_{t+1}^{p_i} = \mathbf{f}^i(\mathbf{x}_t^{p_i}, \mathbf{u}_t^{p_i}),\ \ i \in \mathcal{I}.$$

*Control policy:* After solving Problem (10), the control policy for agent $i$ is designed as an LQR policy to track the nominal trajectory of agent $i$, $\mathbf{x}_t^{p_i}$ as:

$$\mathbf{u}_t^i = \mathbf{u}_t^{p_i} - \mathbf{L}_t^i(\mathbf{x}_t^i - \mathbf{x}_t^{p_i}), \tag{5.7}$$

$$\mathbf{L}_t^i = (\mathbf{W}_t^{u_i} + (\mathbf{B}_t^i)^T \mathbf{S}_t^i \mathbf{B}_t^i)^{-1} (\mathbf{B}_t^i)^T \mathbf{S}_t^i \mathbf{A}_t^i, \tag{5.8}$$

where the Jacobians are

$$\mathbf{A}_t^i := \nabla_{\mathbf{x}} \mathbf{f}^i(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_i}, \mathbf{u}_t^{p_i}}, \mathbf{B}_t^i := \nabla_{\mathbf{u}} \mathbf{f}^i(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_i}, \mathbf{u}_t^{p_i}}, \mathbf{G}_t^i := \epsilon \boldsymbol{\sigma}^{\mathbf{f}_i}(t),$$

and $\mathbf{S}_t^i$ is obtained using a single-agent backward dynamic Riccati equation with $\mathbf{S}_{K_i}^i = \mathbf{W}_{K_i}^{x_i}$:

$$\mathbf{S}_{t-1}^i = (\mathbf{A}_t^i)^T \mathbf{S}_t^i \mathbf{A}_t^i - (\mathbf{A}_t^i)^T \mathbf{S}_t^i \mathbf{B}_t^i \mathbf{L}_t^i + \mathbf{W}_t^{x_i}. \tag{5.9}$$

# 6. PARTIALY-OBSERVED SINGLE-AGENT SYSTEM

In this chapter, we extend the results of Chapter 4 to the situations with imperfect measurements of the state to prove the main Result 3. In this case, we assume that the initial state is only known up to a distribution, and that it is subsequently partially observed through a noisy observation process. The outline of this section is parallel to the outline of Chapter 4. Moreover, where necessary we will refer to the previous equations pointing out the minimal changes necessary without restating them.

## 6.1 Case I: The Deterministic Optimal Policy

In this section, we analyze the performance of the deterministic optimal control policy used in the stochastic problem.

**Problem 11 Deterministic Closed-Loop Problem**: *Given an initial state* $\bar{\mathbf{x}}_0$*, we begin by determining a continuously differentiable optimal observation-trajectory-feedback policy for*

$$\min_{\boldsymbol{\pi}} \sum_{t=0}^{K-1} c_t(\mathbf{x}_t, \mathbf{u}_t) + c_K(\mathbf{x}_K)$$

$$s.t.\ \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) \tag{6.1}$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t). \tag{6.2}$$

*Nominal trajectories:* For $0 \leq t \leq K-1$, let $\boldsymbol{\pi}^d$ be the optimal feedback law of the deterministic problem above, let $\mathbf{x}_t^p$ and $\mathbf{z}_t^p$ be the corresponding state and observations, whose evolutions are governed by:

$$\mathbf{u}_t^p := \boldsymbol{\pi}^d(\mathbf{z}_{0:t}^p),\ \mathbf{x}_{t+1}^p := \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p),\ \mathbf{z}_{t+1}^p := \mathbf{h}(\mathbf{x}_{t+1}^p), \tag{6.3}$$

where $\mathbf{x}_0^p := \bar{\mathbf{x}}_0$. We refer to these as the nominal trajectories.

*Linearization of the system equations:* We consider the application of a control $\mathbf{u}_t = \boldsymbol{\pi}^d(\mathbf{z}_{0:t})$ to the stochastic system. Then the resulting trajectory is:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \boldsymbol{\pi}^d(\mathbf{z}_{0:t})) + \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}} \mathbf{w}_t, \tag{6.4}$$

$$\mathbf{z}_{t+1} = \mathbf{h}(\mathbf{x}_{t+1}) + \epsilon \boldsymbol{\sigma}_{t+1}^{\mathbf{h}} \mathbf{v}_{t+1}. \tag{6.5}$$

Let $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}_t^p$, and $\tilde{\mathbf{z}}_t := \mathbf{z}_t - \mathbf{z}_t^p$ denote the state and observation errors, respectively. Then we linearize the drift of the process and observation models around the nominal trajectory. Hence, for $0 \leq t \leq K-1$:

$$\begin{aligned}
\tilde{\mathbf{x}}_{t+1} &= \mathbf{f}(\mathbf{x}_t, \boldsymbol{\pi}^d(\mathbf{z}_{0:t})) - \mathbf{f}(\mathbf{x}_t^p, \boldsymbol{\pi}^d(\mathbf{z}_{0:t}^p)) + \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}} \mathbf{w}_t && \text{(6.6a)}\\
&= \mathbf{A}_t \tilde{\mathbf{x}}_t - \sum_{s=0}^{t} \mathbf{B}_t \mathbf{L}_{s,t} \tilde{\mathbf{z}}_s + \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}} \mathbf{w}_t + o(\|\tilde{\mathbf{x}}_t\| + \sum_{s=0}^{t} \|\tilde{\mathbf{z}}_s\|) && \text{(6.6b)}\\
&= \mathbf{A}_t \tilde{\mathbf{x}}_t - \sum_{s=0}^{t} \mathbf{B}_t \mathbf{L}_{s,t} \tilde{\mathbf{z}}_s + \mathbf{G}_t \mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty), && \text{(6.6c)}\\
\tilde{\mathbf{z}}_{t+1} &= \mathbf{h}(\mathbf{x}_{t+1}) - \mathbf{h}(\mathbf{x}_{t+1}^p) + \epsilon \boldsymbol{\sigma}_{t+1}^{\mathbf{h}} \mathbf{v}_{t+1} && \text{(6.6d)}\\
&= \mathbf{H}_{t+1} \tilde{\mathbf{x}}_{t+1} + \epsilon \boldsymbol{\sigma}_{t+1}^{\mathbf{h}} \mathbf{v}_{t+1} + o(\|\tilde{\mathbf{x}}_{t+1}\|) && \text{(6.6e)}\\
&= \mathbf{H}_{t+1} \tilde{\mathbf{x}}_{t+1} + \mathbf{M}_{t+1} \mathbf{v}_{t+1} + o(\|\tilde{\mathbf{x}}\|_\infty) && \text{(6.6f)}
\end{aligned}$$

as $(\|\tilde{\mathbf{x}}\|_\infty + (\|\tilde{\mathbf{z}}\|_\infty) \downarrow 0$, where we have:

- $\mathbf{A}_t := \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{B}_t := \nabla_{\mathbf{u}} \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{L}_{s,t} := -\nabla_{\mathbf{z}_s} \boldsymbol{\pi}^d(\mathbf{z}_{0:s})|_{\mathbf{z}_{0:t}^p}$, $\mathbf{G}_t := \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}}$;
- $\mathbf{H}_t := \nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x})|_{\mathbf{x}_t^p}$, $\mathbf{M}_t := \epsilon \boldsymbol{\sigma}_t^{\mathbf{h}}$;

- $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}_0^p$, and $\tilde{\mathbf{z}}_0 = \mathbf{z}_0 - \mathbf{z}_0^p$.

Therefore, we have:

$$\begin{aligned}
\tilde{\mathbf{x}}_{t+1} &= \mathbf{A}_t\tilde{\mathbf{x}}_t - \sum_{s=0}^{t} \mathbf{B}_t\mathbf{L}_{s,t}(\mathbf{H}_s\tilde{\mathbf{x}}_s + \mathbf{M}_s\mathbf{v}_s) + \mathbf{G}_t\mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty) && (6.7a)\\
&= \sum_{s=0}^{t}(\mathbf{U}_{s,t}\tilde{\mathbf{x}}_s + \mathbf{V}_{s,t}\mathbf{v}_s) + \mathbf{G}_t\mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty), && (6.7b)
\end{aligned}$$

where $\mathbf{U}_{s,t} := \mathbf{A}_s - \mathbf{B}_t\mathbf{L}_{s,t}\mathbf{H}_s, s = t$, $\mathbf{U}_{s,t} := -\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{H}_s, 0 \leq s \leq t-1$, and $\mathbf{V}_{s,t} := -\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{M}_s, 0 \leq s \leq t$. The above linear recursive equation can be solved. In particular, there exists matrices $\mathbf{U}_t^{\mathbf{x}_0}, 0 \leq t \leq K-1$, $\mathbf{V}_{s,t}^{\mathbf{v}}, 0 \leq s \leq t, 0 \leq t \leq K-1$, and $\mathbf{W}_{s,t}^{\mathbf{w}}, 0 \leq s \leq t, 0 \leq t \leq K-1$ such that

$$\begin{aligned}
\tilde{\mathbf{x}}_{t+1} &= \mathbf{U}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t}(\mathbf{V}_{s,t}^{\mathbf{v}}\mathbf{v}_s + \mathbf{W}_{s,t}^{\mathbf{w}}\mathbf{w}_t) + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty), && (6.8a)\\
\tilde{\mathbf{z}}_{t+1} &= \mathbf{H}_{t+1}(\mathbf{U}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t}(\mathbf{V}_{s,t}^{\mathbf{v}}\mathbf{v}_s + \mathbf{W}_{s,t}^{\mathbf{w}}\mathbf{w}_t)) + \mathbf{M}_{t+1}\mathbf{v}_{t+1} + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty)\\
&= \mathbf{U}_t^{\mathbf{z},\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t+1}\mathbf{V}_{s,t}^{\mathbf{z},\mathbf{v}}\mathbf{v}_s + \sum_{s=0}^{t}\mathbf{W}_{s,t}^{\mathbf{z},\mathbf{w}}\mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty), && (6.8b)
\end{aligned}$$

where for $0 \leq t \leq K-1$, we have $\mathbf{U}_t^{\mathbf{z},\mathbf{x}_0} := \mathbf{H}_{t+1}\mathbf{U}_t^{\mathbf{x}_0}$, $\mathbf{V}_{s,t}^{\mathbf{z},\mathbf{v}} := \mathbf{H}_{t+1}\mathbf{V}_{s,t}^{\mathbf{v}}, 0 \leq s \leq t$, $\mathbf{V}_{s,t}^{\mathbf{z},\mathbf{v}} := \mathbf{M}_{t+1}, s = t+1$ and $\mathbf{W}_{s,t}^{\mathbf{z},\mathbf{w}} := \mathbf{H}_{t+1}\mathbf{W}_{s,t}^{\mathbf{w}}, 0 \leq s \leq t$.

*The exactly linear l-system:* From the above system of (6.8), we remove the $o(\cdot)$ terms, and define an exactly linear system:

$$\begin{aligned}
\tilde{\mathbf{x}}_{t+1}^l &:= \mathbf{U}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0^l + \sum_{s=0}^{t}(\mathbf{V}_{s,t}^{\mathbf{v}}\mathbf{v}_s + \mathbf{W}_{s,t}^{\mathbf{w}}\mathbf{w}_t), && (6.9a)\\
\tilde{\mathbf{z}}_{t+1}^l &:= \mathbf{U}_t^{\mathbf{z},\mathbf{x}_0}\tilde{\mathbf{x}}_0^l + \sum_{s=0}^{t+1}\mathbf{V}_{s,t}^{\mathbf{z},\mathbf{v}}\mathbf{v}_s + \sum_{s=0}^{t}\mathbf{W}_{s,t}^{\mathbf{z},\mathbf{w}}\mathbf{w}_t, && (6.9b)
\end{aligned}$$

where $\tilde{\mathbf{x}}_0^l := \tilde{\mathbf{x}}_0$, and $\tilde{\mathbf{z}}_0^l := \mathbf{H}_0\tilde{\mathbf{x}}_0^l + \mathbf{M}_0\mathbf{v}_0$.

*The difference d-system:* We denote the difference between the two systems of (6.8) and (6.9) by a superscript $d$, and define for $0 \leq t \leq K-1$, $\tilde{\mathbf{x}}^d_{t+1} := \tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}^l_{t+1}$, and $\tilde{\mathbf{z}}^d_{t+1} := \tilde{\mathbf{z}}_{t+1} - \tilde{\mathbf{z}}^l_{t+1}$, where $\tilde{\mathbf{x}}^d_0 = \tilde{\mathbf{x}}_0 - \tilde{\mathbf{x}}^l_0 = \mathbf{0}$, and $\tilde{\mathbf{z}}^d_0 = \tilde{\mathbf{z}}_0 - \tilde{\mathbf{z}}^l_0 = \mathbf{0}$. Therefore,

$$\tilde{\mathbf{x}}^d_{t+1} = o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty), \tag{6.10}$$

$$\tilde{\mathbf{z}}^d_{t+1} = o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty), \tag{6.11}$$

Hence,

$$O(\|\tilde{\mathbf{x}}^l\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty) + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty), \tag{6.12}$$

$$O(\|\tilde{\mathbf{z}}^l\|_\infty) = O(\|\tilde{\mathbf{z}}\|_\infty) + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty), \tag{6.13}$$

This means that all the errors in the original system, the $l$-system, and the $d$-system are of the order of $O(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty)$. Moreover, $O(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty)$ is itself $O(\|\tilde{\mathbf{x}}^l\|_\infty)$, which we calculate next.

*Large deviations:* The $l$-system is a linear Gaussian system and in fact $\tilde{\mathbf{x}}^l_t$ is a linear combination of independent Gaussian random variables. Hence, $\tilde{\mathbf{x}}^l_t$ is also a Gaussian variable, for which we use the large deviations result of Lemma 1 with some modifications. In particular, for the sake of simplicity, let us replace $\mathbf{w}_t$ with $\tilde{\mathbf{w}}_t = \epsilon \mathbf{w}_t$, and replace $\mathbf{v}_t$ with $\tilde{\mathbf{v}}_t = \epsilon \mathbf{v}_t$. Then, $\tilde{\mathbf{w}}_t \sim \mathcal{N}(0, \epsilon^2 \boldsymbol{\Sigma}_{\mathbf{w}})$ and $\tilde{\mathbf{v}}_t \sim \mathcal{N}(0, \epsilon^2 \boldsymbol{\Sigma}_{\mathbf{v}})$. Also redefine, $\mathbf{G}_t := \boldsymbol{\sigma}^{\mathbf{f}}_t$ and $\mathbf{M}_t := \boldsymbol{\sigma}^{\mathbf{h}}_t$. Then, by redefining $\alpha_{i,t}$ in Lemma 1 as

$$\alpha_{i,t} := \sum_{j=1}^{n_x} [((\mathbf{U}^{\mathbf{x}_0}_t \boldsymbol{\Sigma}_{\mathbf{x}_0})^{ij})^2 + \sum_{s=0}^{t} ((\mathbf{V}^{\mathbf{v}}_{s,t} \boldsymbol{\Sigma}_{\mathbf{v}})^{i,j})^2 + \sum_{s=0}^{t} ((\mathbf{W}^{\mathbf{w}}_{s,t} \boldsymbol{\Sigma}_{\mathbf{w}})^{ij})^2], \tag{6.14}$$

where $(\cdot)^{ij}$ shows the $ij$-th element of the corresponding matrices. Hence, we get the probability $P\{\max_{0 \leq t \leq K} \|\tilde{\mathbf{x}}^l_t\| \geq \delta\} = o(\epsilon)$ for each finite $\delta \geq 0$.

Let $\Omega(\delta)$ be the set where $\max_{0\leq t\leq K}\|\tilde{\mathbf{x}}_t^l\| \leq \delta$. Then, $P(\Omega(\delta)) \geq 1 - o(\epsilon)$ and for $\omega \in \Omega(\delta)$, $\|\tilde{\mathbf{x}}^l\|_\infty = O(\delta)$. Therefore, from the calculations above, we have that $O(\|\tilde{\mathbf{x}}\|_\infty) = O(\delta)$, and hence all the other errors are also $O(\delta)$ for $\omega \in \Omega(\delta)$.

Then for $\omega \in \Omega(\delta)$ and for all $0 \leq t \leq K-1$,

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^p + \tilde{\mathbf{x}}_{t+1}^l + O(\delta), \tag{6.15a}$$

$$\mathbf{z}_{t+1} = \mathbf{z}_{t+1}^p + \tilde{\mathbf{z}}_{t+1}^l + O(\delta), \tag{6.15b}$$

which means that the linear Gaussian stochastic $\tilde{(\cdot)}^l$-system along with the deterministic $p$-system can be used to control and estimate the original system given the $O(\delta)$ approximations hold (with probability of at least $1-o(\epsilon)$). In another interpretation, the original system can be approximated for all $0\leq t\leq K-1$ as:

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^l + O(\delta), \tag{6.16a}$$

$$\mathbf{z}_{t+1} = \mathbf{z}_{t+1}^l + O(\delta). \tag{6.16b}$$

### *6.1.1 Analysis of the Cost*

Next, we use the more general definition of the cost function directly in terms of the state, and try to approximate the cost function of the original system in terms of the cost of the $l$-system.

*Linearization of the cost function:* We consider the general cost function:

$$J_{\boldsymbol{\pi}} := \sum_{t=0}^{K-1} c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t) + c_K^{\boldsymbol{\pi}}(\mathbf{x}_K), \tag{6.17}$$

and linearize it around the nominal system:

$$J = J^p + \tilde{J}_1 + o(\sum_{t=1}^{K}(\|\tilde{\mathbf{x}}_t\| + \|\tilde{\mathbf{z}}_t\|)) \tag{6.18a}$$

$$= J^p + \tilde{J}_1 + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{z}}\|_\infty), \tag{6.18b}$$

from the assumption that the cost function is continuously differentiable and bounded. That is $|c_t| \leq M$ and $|c_K| \leq M$ for some $M > 0$. Moreover,

- $\mathbf{C}_t^{\mathbf{x}} = \nabla_{\mathbf{x}} c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{u}} = \nabla_{\mathbf{u}} c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_K^{\mathbf{x}} = \nabla_{\mathbf{x}} c_K(\mathbf{x})|_{\mathbf{x}_K^p}$;
- $J^p := \sum_{t=0}^{K-1} c_t(\mathbf{x}_t^p, \mathbf{u}_t^p) + c_K(\mathbf{x}_K^p)$ denotes the nominal cost;
- $\tilde{J}_1 := \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t - \sum_{s=0}^{t} \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_s\tilde{\mathbf{z}}_s) + \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K$ is the first order error in the cost; and
- $J_1 := J^p + \tilde{J}_1$ is the first order approximation of the cost function.

Therefore, for $\omega \in \Omega(\delta)$, and

$$J = J^p + \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t - \sum_{s=0}^{t} \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s) + \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K + O(\delta) \tag{6.19a}$$

$$= J^p + \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t^l - \sum_{s=0}^{t} \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s^l) + \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K^l + O(\delta). \tag{6.19b}$$

The above calculations show that the cost of the original system is close to the cost of the $l$-system. Moreover, $J - J_1 = O(\delta)$ for $\omega \in \Omega(\delta)$.

Next, we provide the main result regarding the expected first order error of the cost function.

**Theorem 6 First-Order Cost Function Error for a Partially-Observed System with a Deterministic Policy:** *Given that process noises are zero mean i.i.d. Gaussian, the initial error is zero mean Gaussian, and all the functions are in* $\mathbb{C}^1$*, under a first-order approximation for the small noise paradigm, the stochastic cost* *function is dominated by the nominal part of the cost function, and the expected first-order error is* $O(\delta)$*. That is,*

$$\mathbb{E}[\tilde{J}_1] = O(\delta), \text{ and } \mathbb{E}[J] = J^p + O(\delta).$$

*Moreover, by choosing* $\delta = \sqrt{\log(\frac{1}{\epsilon})\epsilon}$*, we have*

$$\mathbb{E}[\tilde{J}_1] = O(\epsilon^{1-\gamma}), \text{ and } \mathbb{E}[J] = J^p + O(\epsilon^{1-\gamma}),$$

*for some* $0 < \gamma \ll 1$*, which shows that this error tends to zero with a near-first-order rate as* $\epsilon \downarrow 0$*.*

***Proof 14*** *Let* $\tilde{J}_1^l{:=}\sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t^l-\sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s^l){+}\mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K^l$*. Also note* $\mathbb{E}[\tilde{\mathbf{x}}_0^l] = \mathbb{E}[\tilde{\mathbf{x}}_0] = \mathbb{E}[\mathbf{x}_0 - \hat{\mathbf{x}}_0] = \mathbf{0}$*,* $\mathbb{E}[\tilde{\mathbf{z}}_0^l] = \mathbb{E}[\mathbf{H}_0\tilde{\mathbf{x}}_0^l + \mathbf{M}_0\mathbf{v}_0] = \mathbf{0}$*, and* $\mathbb{E}[\mathbf{w}_t] = \mathbb{E}[\mathbf{v}_t] = 0$ *for all* $t$*. Then, we use* (6.9)*. First, we calculate* $\mathbb{E}[\tilde{\mathbf{x}}_{t+1}^l]$ *and* $\mathbb{E}[\tilde{\mathbf{z}}_{t+1}^l]$ *for* $0 \leq t \leq K-1$*:*

$$\begin{aligned}
\mathbb{E}[\tilde{\mathbf{x}}_{t+1}^l] &= \mathbf{U}_t^{\mathbf{x}_0}\mathbb{E}[\tilde{\mathbf{x}}_0^l] + \sum_{s=0}^{t}(\mathbf{V}_{s,t}^{\mathbf{v}}\mathbb{E}[\mathbf{v}_s] + \mathbf{W}_{s,t}^{\mathbf{w}}\mathbb{E}[\mathbf{w}_t]) = \mathbf{0},\\
\mathbb{E}[\tilde{\mathbf{z}}_{t+1}^l] &= \mathbf{U}_t^{\mathbf{z},\mathbf{x}_0}\mathbb{E}[\tilde{\mathbf{x}}_0^l] + \sum_{s=0}^{t+1}\mathbf{V}_{s,t}^{\mathbf{z},\mathbf{v}}\mathbb{E}[\mathbf{v}_s] + \sum_{s=0}^{t}\mathbf{W}_{s,t}^{\mathbf{z},\mathbf{w}}\mathbb{E}[\mathbf{w}_t] = \mathbf{0}.
\end{aligned}$$

*Therefore, we have:*

$$\mathbb{E}[\tilde{J}_1^l]{=}\sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\mathbb{E}[\tilde{\mathbf{x}}_t^l] - \sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbb{E}[\tilde{\mathbf{z}}_s^l]){+}\mathbf{C}_K^{\mathbf{x}}\mathbb{E}[\tilde{\mathbf{x}}_K^l] = 0.$$

*Now, we take expectation on both sides of* (6.19b)*. Since, for* $\omega \notin \Omega(\delta)$*,* $J \leq M$*, then*

$$\mathbb{E}[J - J^p] = P(\Omega(\delta))(\mathbb{E}[\tilde{J}_1^l] + O(\delta)) + M(1 - P(\Omega(\delta)))$$

$$= P(\Omega(\delta))O(\delta) + M(1 - P(\Omega(\delta))) \tag{6.20}$$

*Now, the last expression is the same as* (4.19)*. Although* $\Omega(\delta)$ *is not the same as in Theorem 2,* $P(\Omega(\delta))$ *is still the same. In the proof of Theorem 2 while we discussed on the probabilistic argument and choosing the proper* $\delta$*, we showed that by choosing* $\delta := \sqrt{-\log(\epsilon)}\epsilon$*, the* $\mathbb{E}[J - J^p] = O(\epsilon^{1-\gamma})$*. The same argument follows through and this theorem is proved.*

Hence, the expected stochastic cost is equal to the nominal cost with a very high probability as $\epsilon \downarrow 0$. The result is summarized below:

**Corollary 5 Near-First-Order Optimality of the Deterministic Optimal Policy for the Stochastic Partially-Observed System Under Small Noise.** *Based on Theorem 6, for a partially-observed system where the function are in* $\mathbb{C}^1$ *under the small noise paradigm, as* $\epsilon \downarrow 0$*, the deterministic optimal control law becomes* $O(\epsilon^{1-\gamma})$*-optimal with* $0 < \gamma \ll 1$ *for the stochastic problem.*

***Proof 15*** *Using Theorem 8, for* $\omega \in \Omega(\delta)$ *we have* $\mathbb{E}[J] = J^p + O(\epsilon^{1-\gamma})$*, which is the cost of applying policy* $\boldsymbol{\pi}^d$ *to the stochastic system. Now suppose* $\boldsymbol{\pi}^*$ *is the optimal stochastic policy. By assumption it is continuously differentiable. Therefore, by modifying the definition of* $\mathbf{L}_{s,t}$ *as* $\mathbf{L}_{s,t} = -\nabla_{\mathbf{z}_s}\boldsymbol{\pi}^*_t(\mathbf{z}_{0:s})|_{\mathbf{z}^{*p}_{0:t}}$*, defining* $\mathbf{u}^{*p}_t = \boldsymbol{\pi}^*_t(\mathbf{z}^{*p}_{0:t})$ *and replacing* $p$ *with* $*p$ *in* (4.23)*, we have* $\boldsymbol{\pi}^*_t(\mathbf{z}_{0:t}) = \mathbf{u}^{*p}_t - \sum_{s=0}^{t}\mathbf{L}_{s,t}(\mathbf{z}_s - \mathbf{z}^{*p}_s) + o(\|\tilde{\mathbf{z}}\|_\infty)$*. Similarly, by using appropriate modifications, the entire calculations of this section hold for this policy. Hence, using Theorem 8 for this system, the cost function of policy* $\boldsymbol{\pi}^*$ *can be written as* $\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{1-\gamma})$*. Now, by construction* $J^p \leq J^{*p}$*, and*

$$\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{1-\gamma}) \geq J^p + O(\epsilon^{1-\gamma}) = \mathbb{E}[J_{\boldsymbol{\pi}^d}] + O(\epsilon^{1-\gamma})$$

*As a result, policy $\boldsymbol{\pi}^d$ is within $O(\epsilon^{1-\gamma})$ of the optimal stochastic policy.*

## 6.2 Case II: T-LQG

*Stochastic system:* Given $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \epsilon^2 \boldsymbol{\Sigma}_{\mathbf{x}_0})$, consider the following problem:

$$\min_{\boldsymbol{\pi}} \mathbb{E}[\sum_{t=0}^{K-1} c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t) + c_K^{\boldsymbol{\pi}}(\mathbf{x}_K)]$$

$$s.t.\ \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}} \mathbf{w}_t, \tag{6.21a}$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \epsilon \boldsymbol{\sigma}_t^{\mathbf{h}} \mathbf{v}_t, \tag{6.21b}$$

*Deterministic System:* Given $\bar{\mathbf{x}}_0$, consider the following deterministic problem:

$$\min_{\mathbf{u}_{0:K-1}} \sum_{t=0}^{K-1} c_t(\mathbf{x}_t, \mathbf{u}_t) + c_K(\mathbf{x}_K)$$

$$s.t.\ \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t), \tag{6.22a}$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t). \tag{6.22b}$$

*Nominal trajectories:* For $0 \leq t \leq K-1$, let $\mathbf{u}_t^p$ be the optimal open-loop solution of the deterministic problem above, and let $\mathbf{x}_t^p$ and $\mathbf{z}_t^p$ be the corresponding state and observations, whose evolutions are governed by:

$$\mathbf{x}_{t+1}^p := \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p),\ \mathbf{z}_{t+1}^p := \mathbf{h}(\mathbf{x}_{t+1}^p), \tag{6.23}$$

where $\mathbf{x}_0^p := \bar{\mathbf{x}}_0$. We refer to the $p$-system as the nominal trajectories

*Linearization of the system equations:* We apply the control $\mathbf{u}_t = \mathbf{u}_t^p + \tilde{\mathbf{u}}_t$ to the stochastic system. Then, the resulting trajectory is $\mathbf{x}_t = \mathbf{x}_t^p + \tilde{\mathbf{x}}_t$ and $\mathbf{z}_t = \mathbf{z}_t^p + \tilde{\mathbf{z}}_t$, where $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}_t^p$, and $\tilde{\mathbf{z}}_t := \mathbf{z}_t - \mathbf{z}_t^p$ denote the state and observation errors, respectively. Then,

$$\mathbf{x}^p_{t+1} + \tilde{\mathbf{x}}_{t+1} = \mathbf{f}(\mathbf{x}^p_t + \tilde{\mathbf{x}}_t, \mathbf{u}^p_t + \tilde{\mathbf{u}}_t) + \epsilon \boldsymbol{\sigma}^{\mathbf{f}}_t \mathbf{w}_t, \tag{6.24}$$

$$\mathbf{z}^p_{t+1} + \tilde{\mathbf{z}}_{t+1} = \mathbf{h}(\mathbf{x}^p_{t+1} + \tilde{\mathbf{x}}_{t+1}) + \epsilon \boldsymbol{\sigma}^{\mathbf{h}}_{t+1} \mathbf{v}_{t+1}, \tag{6.25}$$

Then we linearize the drifts of the processes around their nominal counterparts. Then for $0 \leq t \leq K-1$:

$$\begin{aligned} \tilde{\mathbf{x}}_{t+1} &= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \mathbf{G}_t \mathbf{w}_t + o(\|\tilde{\mathbf{x}}_t\| + \|\tilde{\mathbf{u}}_t\|) && \text{(6.26a)} \\ &= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \mathbf{G}_t \mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{u}}\|_\infty), && \text{(6.26b)} \\ \tilde{\mathbf{z}}_{t+1} &= \mathbf{H}_{t+1} \tilde{\mathbf{x}}_{t+1} + \mathbf{M}_{t+1} \mathbf{v}_{t+1} + o(\|\tilde{\mathbf{x}}_{t+1}\|) && \text{(6.26c)} \\ &= \mathbf{H}_{t+1} \tilde{\mathbf{x}}_{t+1} + \mathbf{M}_{t+1} \mathbf{v}_{t+1} + o(\|\tilde{\mathbf{x}}\|_\infty), && \text{(6.26d)} \end{aligned}$$

as $\epsilon \downarrow 0$, where we have:

- $\mathbf{A}_t := \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}^p_t, \mathbf{u}^p_t}$, $\mathbf{B}_t := \nabla_{\mathbf{u}} \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}^p_t, \mathbf{u}^p_t}$, $\mathbf{G}_t := \epsilon \boldsymbol{\sigma}^{\mathbf{f}}_t$;
- $\mathbf{H}_t := \nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x})|_{\mathbf{x}^p_t}$, $\mathbf{M}_t := \epsilon \boldsymbol{\sigma}^{\mathbf{h}}(t)$; and
- $\tilde{\mathbf{u}}_0 = \mathbf{u}_0 - \mathbf{u}^p_0 = \mathbf{0}$, and $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}^p_0$.

*The exactly linear l-system:* Based on the linearized system of (10.2), and removing the $o(\cdot)$ terms, we define a set of exactly linear systems:

$$\tilde{\mathbf{x}}^l_{t+1} := \mathbf{A}_t \tilde{\mathbf{x}}^l_t + \mathbf{B}_t \tilde{\mathbf{u}}^l_t + \mathbf{G}_t \mathbf{w}_t, \tag{6.27a}$$

$$\tilde{\mathbf{z}}^l_{t+1} := \mathbf{H}_{t+1} \tilde{\mathbf{x}}^l_{t+1} + \mathbf{M}_{t+1} \mathbf{v}_{t+1}, \tag{6.27b}$$

where, $\tilde{\mathbf{x}}^l_0 := \tilde{\mathbf{x}}_0$.

**Theorem 7 (Separation of Estimation and Control)** *The design of a stochastic controller for a partially observed linear system can be formulated as two separate designs of an estimator and a controller with perfect state information.*

Proof of this theorem can be found in [2]. As a result of Theorem 7, the control law for the linear Gaussian system ($l$-system) can be designed to track the nominal trajectory of the system with perfect measurement of estimate's trajectory. Moreover, the estimation effort can be performed separately using a KF, which is brought next.

*Kalman Filter:* The estimates of the $l$-system can be obtained using the KF equations:

$$\hat{\tilde{\mathbf{x}}}^l_{t+1} := \mathbf{A}_t\hat{\tilde{\mathbf{x}}}^l_t + \mathbf{B}_t\tilde{\mathbf{u}}^l_t + \mathbf{K}_{t+1}(\tilde{\mathbf{z}}^l_{t+1} - \mathbf{H}_{t+1}(\mathbf{A}_t\hat{\tilde{\mathbf{x}}}^l_t + \mathbf{B}_t\tilde{\mathbf{u}}^l_t)), \tag{6.28a}$$

$$\bar{\mathbf{P}}_{t+1} := \mathbf{A}_t\mathbf{P}^l_t\mathbf{A}^T_t + \mathbf{G}_t\boldsymbol{\Sigma}_{\mathbf{w}}\mathbf{G}^T_t, \tag{6.28b}$$

$$\boldsymbol{\Sigma}^{\boldsymbol{\nu}}_{t+1} := \mathbf{H}_{t+1}\bar{\mathbf{P}}_{t+1}(\mathbf{H}_{t+1})^T + \mathbf{M}_{t+1}\boldsymbol{\Sigma}_{\mathbf{v}}(\mathbf{M}_{t+1})^T, \tag{6.28c}$$

$$\mathbf{K}_{t+1} := \bar{\mathbf{P}}_{t+1}\mathbf{H}^T_{t+1}(\boldsymbol{\Sigma}^{\boldsymbol{\nu}}_{t+1})^{-1}, \tag{6.28d}$$

$$\mathbf{P}^l_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H}_{t+1})\bar{\mathbf{P}}_{t+1}. \tag{6.28e}$$

where $\mathbf{P}^l_0 := \epsilon^2\boldsymbol{\Sigma}_{\mathbf{x}_0}$ and $\hat{\tilde{\mathbf{x}}}^l_0 := \mathbf{0}$.

*LQG policy:* Let us design the LQG policy for the $l$-system with the cost:

$$\min_{\boldsymbol{\pi}} \ \mathbb{E}[\sum_{t=0}^{K-1}(\tilde{\mathbf{x}}^l_t)^T\mathbf{W}^x_t\tilde{\mathbf{x}}^l_t + (\tilde{\mathbf{u}}^l_t)^T\mathbf{W}^u_t\tilde{\mathbf{u}}^l_t]. \tag{6.29}$$

This problem is solved using the control theory's separation principle and the resulting policy is $\tilde{\mathbf{u}}^l_t = -\mathbf{L}_t\hat{\tilde{\mathbf{x}}}^l_t$. Now, since $\tilde{\mathbf{z}}^l_{t+1}$ is unobserved, we modify the mean equation by replacing it with $\tilde{\mathbf{z}}_{t+1}$:

$$\hat{\tilde{\mathbf{x}}}_{t+1} := \mathbf{A}_t\hat{\tilde{\mathbf{x}}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{K}_{t+1}(\tilde{\mathbf{z}}_{t+1} - \mathbf{H}_{t+1}(\mathbf{A}_t\hat{\tilde{\mathbf{x}}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t)), \tag{6.30}$$

where $\hat{\tilde{\mathbf{x}}}_0 := \mathbf{0}$, and use $\tilde{\mathbf{u}}_t = -\mathbf{L}_t \hat{\tilde{\mathbf{x}}}_t$ in the original system. Hence, (10.2) is rewritten as:

$$\begin{aligned}
\tilde{\mathbf{x}}_{t+1} &= \mathbf{A}_t \tilde{\mathbf{x}}_t - \mathbf{B}_t \mathbf{L}_t \hat{\tilde{\mathbf{x}}}_t + \mathbf{G}_t \mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\tilde{\mathbf{u}}\|_\infty), && \text{(6.31a)}\\
&= \mathbf{A}_t \tilde{\mathbf{x}}_t - \mathbf{B}_t \mathbf{L}_t \hat{\tilde{\mathbf{x}}}_t + \mathbf{G}_t \mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty), && \text{(6.31b)}\\
\tilde{\mathbf{z}}_{t+1} &= \mathbf{H}_{t+1} \tilde{\mathbf{x}}_{t+1} + \mathbf{M}_{t+1} \mathbf{v}_{t+1} + o(\|\tilde{\mathbf{x}}\|_\infty). && \text{(6.31c)}
\end{aligned}$$

Also, the $l$-system becomes as follows:

$$\begin{aligned}
\tilde{\mathbf{x}}^l_{t+1} &= \mathbf{A}_t \tilde{\mathbf{x}}^l_t - \mathbf{B}_t \mathbf{L}_t \hat{\tilde{\mathbf{x}}}^l_t + \mathbf{G}_t \mathbf{w}_t, && \text{(6.32a)}\\
\tilde{\mathbf{z}}^l_{t+1} &= \mathbf{H}_{t+1} \tilde{\mathbf{x}}^l_{t+1} + \mathbf{M}_{t+1} \mathbf{v}_{t+1}. && \text{(6.32b)}
\end{aligned}$$

*The difference d-system:* We denote the difference between the two systems of (6.31) and (6.32) by $d$ superscript, and define for $0 \le t \le K-1$:

$$\begin{aligned}
&\tilde{\mathbf{u}}^d_t := \tilde{\mathbf{u}}_t - \tilde{\mathbf{u}}^l_t, && \tilde{\mathbf{u}}^d_t = -\mathbf{L}_t(\hat{\tilde{\mathbf{x}}}_t - \hat{\tilde{\mathbf{x}}}^l_t), && \text{(6.33a)}\\
&\tilde{\mathbf{x}}^d_{t+1} := \tilde{\mathbf{x}}_{t+1} - \tilde{\mathbf{x}}^l_{t+1}, && \tilde{\mathbf{x}}^d_{t+1} = \mathbf{A}_t \tilde{\mathbf{x}}^d_t + \mathbf{B}_t \tilde{\mathbf{u}}^d_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty), && \text{(6.33b)}\\
&\tilde{\mathbf{z}}^d_{t+1} := \tilde{\mathbf{z}}_{t+1} - \tilde{\mathbf{z}}^l_{t+1}, && \tilde{\mathbf{z}}^d_{t+1} = \mathbf{H}_{t+1} \tilde{\mathbf{x}}^d_{t+1} + o(\|\tilde{\mathbf{x}}\|_\infty), && \text{(6.33c)}\\
&\hat{\tilde{\mathbf{x}}}^d_{t+1} := \hat{\tilde{\mathbf{x}}}_{t+1} - \hat{\tilde{\mathbf{x}}}^l_{t+1}, && \hat{\tilde{\mathbf{x}}}^d_{t+1} = \mathbf{A}_t \hat{\tilde{\mathbf{x}}}^d_t + \mathbf{B}_t \tilde{\mathbf{u}}^d_t + \mathbf{K}_{t+1}(\tilde{\mathbf{z}}^d_{t+1} - \mathbf{H}_{t+1}(\mathbf{A}_t \hat{\tilde{\mathbf{x}}}^d_t + \mathbf{B}_t \tilde{\mathbf{u}}^d_t)), && \text{(6.33d)}
\end{aligned}$$

where $\tilde{\mathbf{u}}^d_0 = \tilde{\mathbf{u}}_0 - \tilde{\mathbf{u}}^l_0 = \mathbf{0}$, $\tilde{\mathbf{x}}^d_0 = \tilde{\mathbf{x}}_0 - \tilde{\mathbf{x}}^l_0 = \mathbf{0}$, and $\hat{\tilde{\mathbf{x}}}^d_0 := \hat{\tilde{\mathbf{x}}}_0 - \hat{\tilde{\mathbf{x}}}^l_0 = \mathbf{0}$. Let us simplify the above equations:

$$\begin{aligned}
\tilde{\mathbf{u}}^d_t &= -\mathbf{L}_t \hat{\tilde{\mathbf{x}}}^d_t, && \text{(6.34a)}\\
\tilde{\mathbf{x}}^d_{t+1} &= \mathbf{A}_t \tilde{\mathbf{x}}^d_t - \mathbf{B}_t \mathbf{L}_t \hat{\tilde{\mathbf{x}}}^d_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty), && \text{(6.34b)}\\
\tilde{\mathbf{z}}^d_{t+1} &= \mathbf{H}_{t+1} \tilde{\mathbf{x}}^d_{t+1} + o(\|\tilde{\mathbf{x}}\|_\infty), && \text{(6.34c)}
\end{aligned}$$

$$\begin{aligned}\hat{\tilde{\mathbf{x}}}^d_{t+1} &= (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H}_{t+1})(\mathbf{A}_t - \mathbf{B}_t\mathbf{L}_t)\hat{\tilde{\mathbf{x}}}^d_t + \mathbf{K}_{t+1}\tilde{\mathbf{z}}^d_{t+1} &(6.34\text{d})\\ &= (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H}_{t+1})(\mathbf{A}_t - \mathbf{B}_t\mathbf{L}_t)\hat{\tilde{\mathbf{x}}}^d_t + \mathbf{K}_{t+1}\mathbf{H}_{t+1}\tilde{\mathbf{x}}^d_{t+1} + o(\|\tilde{\mathbf{x}}\|_\infty) &(6.34\text{e})\\ &= (\mathbf{A}_t - \mathbf{B}_t\mathbf{L}_t - \mathbf{K}_{t+1}\mathbf{H}_{t+1}\mathbf{A}_t)\hat{\tilde{\mathbf{x}}}^d_t + \mathbf{K}_{t+1}\mathbf{H}_{t+1}\mathbf{A}_t\tilde{\mathbf{x}}^d_t + o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty). &(6.34\text{f})\end{aligned}$$

Next, we can regroup the above equations as follows:

$$\mathbf{A}^d_t := \begin{pmatrix} \mathbf{A}_t, & -\mathbf{B}_t\mathbf{L}_t \\ \mathbf{K}_{t+1}\mathbf{H}_{t+1}\mathbf{A}_t, & \mathbf{A}_t - \mathbf{B}_t\mathbf{L}_t - \mathbf{K}_{t+1}\mathbf{H}_{t+1}\mathbf{A}_t \end{pmatrix}, \tag{6.35}$$

$$\begin{pmatrix} \tilde{\mathbf{x}}^d_{t+1} \\ \hat{\tilde{\mathbf{x}}}^d_{t+1} \end{pmatrix} = \mathbf{A}^d_t \begin{pmatrix} \tilde{\mathbf{x}}^d_t \\ \hat{\tilde{\mathbf{x}}}^d_t \end{pmatrix} + \begin{pmatrix} o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty) \\ o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty) \end{pmatrix}. \tag{6.36}$$

Define $\tilde{\mathbf{A}}^d_{t_1:t_2} := \Pi^{t_2}_{t=t_1}\mathbf{A}^d_t, t_2 \geq t_1 \geq 0$, otherwise, it is the identity matrix. Then,

$$\begin{pmatrix} \tilde{\mathbf{x}}^d_{t+1} \\ \hat{\tilde{\mathbf{x}}}^d_{t+1} \end{pmatrix} = \tilde{\mathbf{A}}^d_{0:t} \begin{pmatrix} \tilde{\mathbf{x}}^d_0 \\ \hat{\tilde{\mathbf{x}}}^d_0 \end{pmatrix} + \begin{pmatrix} o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty) \\ o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty) \end{pmatrix} = \begin{pmatrix} o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty) \\ o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty) \end{pmatrix} \tag{6.37}$$

where we used the fact that $((\tilde{\mathbf{x}}^d_t)^T, (\hat{\tilde{\mathbf{x}}}^d_t)^T)^T = \mathbf{0}$. This leads to $\|\tilde{\mathbf{x}}^d\|_\infty = o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty)$ and $\|\hat{\tilde{\mathbf{x}}}^d\|_\infty = o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty)$. Hence,

$$O(\|\tilde{\mathbf{x}}^l\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty) + o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty), \tag{6.38}$$

$$O(\|\hat{\tilde{\mathbf{x}}}^l\|_\infty) = O(\|\hat{\tilde{\mathbf{x}}}\|_\infty) + o(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty). \tag{6.39}$$

Also using (6.34a), (6.34c), and the definition of $\tilde{\mathbf{u}}_t$ we have:

$$O(\|\tilde{\mathbf{u}}^d\|_\infty) = O(\|\hat{\tilde{\mathbf{x}}}^d\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty), \tag{6.40}$$

$$O(\|\tilde{\mathbf{z}}^d\|_\infty) = O(\|\tilde{\mathbf{x}}^d\|_\infty) + o(\|\tilde{\mathbf{x}}\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty) \tag{6.41}$$

$$O(\|\tilde{\mathbf{u}}\|_\infty) = O(\|\hat{\tilde{\mathbf{x}}}\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty), \tag{6.42}$$

Also, using (6.28a),the definition of $\tilde{\mathbf{u}}_t^l$ and (6.41), we have:

$$O(\|\tilde{\mathbf{z}}\|_\infty) = O(\|\tilde{\mathbf{z}}^d\|_\infty) + O(\|\tilde{\mathbf{z}}^l\|_\infty) \tag{6.43}$$

$$= O(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty) + O(\|\tilde{\mathbf{x}}^l\|_\infty) = O(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty). \tag{6.44}$$

This means that all the errors in the original system, the $l$-system, and the $d$-system are in the order of $O(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty)$. That is, all these errors are in the same order. Moreover, $O(\|\tilde{\mathbf{x}}\|_\infty + \|\hat{\tilde{\mathbf{x}}}\|_\infty)$ is itself written in terms of $O(\|\tilde{\mathbf{x}}^l\|_\infty)$ and $O(\|\hat{\tilde{\mathbf{x}}}^l\|_\infty)$, which we calculate next.

*Innovation process:* It is established that the innovation process of a least square estimation for a linear Gaussian system, defined as $\boldsymbol{\nu}_{t+1} := \tilde{\mathbf{z}}_{t+1}^l - \mathbf{H}_{t+1}(\mathbf{A}_t(\hat{\mathbf{x}}_t^l - \mathbf{x}_t^p) + \mathbf{B}_t\tilde{\mathbf{u}}_t^l)$ for the $l$-system, is a Gaussian white noise [2], i.e., $\mathbb{E}[\boldsymbol{\nu}_t\boldsymbol{\nu}_s^T] = 0, s \neq t$, and $\mathbb{E}[\boldsymbol{\nu}_t\boldsymbol{\nu}_t^T] = \boldsymbol{\Sigma}_t^{\boldsymbol{\nu}}, 1 \leq t \leq K$, which is proportional to $\epsilon^2$ (proven next). This is also referred to as the whitening property of the KF. Therefore, (6.28a) can be written as:

$$\hat{\tilde{\mathbf{x}}}_{t+1}^l = (\mathbf{A}_t - \mathbf{B}_t\mathbf{L}_t)\hat{\tilde{\mathbf{x}}}_t^l + \mathbf{K}_{t+1}(\boldsymbol{\Sigma}_{t+1}^{\boldsymbol{\nu}}(\epsilon))^{\frac{1}{2}}\boldsymbol{\nu}_{t+1}. \tag{6.45}$$

**Lemma 5** **$\epsilon^2$-Dependence of Innovation Process's Variance** *Using the KF, the innovation process's variance is proportional to $\epsilon^2$.*

**Proof 16** *We prove that $\boldsymbol{\Sigma}_t^{\boldsymbol{\nu}}$, $\bar{\mathbf{P}}_t$, and $\mathbf{P}_t$ are all proportional to $\epsilon^2$. This is done by mathematical induction and using the covariance equation of* (6.28b)-(6.28e). *First, note $\mathbf{P}_0^l = \epsilon^2\boldsymbol{\Sigma}_{\mathbf{x}_0}$, then,*

$$\begin{aligned}\bar{\mathbf{P}}_1 =& \mathbf{A}_0\mathbf{P}_0^l\mathbf{A}_0^T + \mathbf{G}_0\boldsymbol{\Sigma}_{\mathbf{w}}\mathbf{G}_0^T = \epsilon^2(\mathbf{A}_0\boldsymbol{\Sigma}_{\mathbf{x}_0}\mathbf{A}_0^T + \boldsymbol{\sigma}_0^{\mathbf{f}}\boldsymbol{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_0^{\mathbf{f}})^T),\\ \boldsymbol{\Sigma}_1^{\boldsymbol{\nu}} =& \mathbf{H}_1\bar{\mathbf{P}}_1\mathbf{H}_1^T + \mathbf{M}_1\boldsymbol{\Sigma}_{\mathbf{v}}\mathbf{M}_1^T\end{aligned}$$

$$
\begin{aligned}
&= \epsilon^2(\mathbf{H}_1(\mathbf{A}_0\mathbf{\Sigma}_{\mathbf{x}_0}\mathbf{A}_0^T + \boldsymbol{\sigma}_0^{\mathbf{f}}\mathbf{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_0^{\mathbf{f}})^T)\mathbf{H}_1^T + \boldsymbol{\sigma}_1^{\mathbf{h}}\mathbf{\Sigma}_{\mathbf{v}}(\boldsymbol{\sigma}_1^{\mathbf{h}})^T),\\
\mathbf{K}_1 &= \bar{\mathbf{P}}_1\mathbf{H}_1^T(\mathbf{\Sigma}_1^{\boldsymbol{\nu}})^{-1}\\
&= (\epsilon^2(\mathbf{A}_0\mathbf{\Sigma}_{\mathbf{x}_0}\mathbf{A}_0^T + \boldsymbol{\sigma}_0^{\mathbf{f}}\mathbf{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_0^{\mathbf{f}})^T))\mathbf{H}_1^T\\
&\quad \times (\epsilon^2(\mathbf{H}_1(\mathbf{A}_0\mathbf{\Sigma}_{\mathbf{x}_0}\mathbf{A}_0^T + \boldsymbol{\sigma}_0^{\mathbf{f}}\mathbf{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_0^{\mathbf{f}})^T)\mathbf{H}_1^T + \boldsymbol{\sigma}_1^{\mathbf{h}}\mathbf{\Sigma}_{\mathbf{v}}(\boldsymbol{\sigma}_1^{\mathbf{h}})^T))^{-1}\\
&= (\mathbf{A}_0\mathbf{\Sigma}_{\mathbf{x}_0}\mathbf{A}_0^T + \boldsymbol{\sigma}_0^{\mathbf{f}}\mathbf{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_0^{\mathbf{f}})^T)\mathbf{H}_1^T\\
&\quad \times ((\mathbf{H}_1(\mathbf{A}_0\mathbf{\Sigma}_{\mathbf{x}_0}\mathbf{A}_0^T + \boldsymbol{\sigma}_0^{\mathbf{f}}\mathbf{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_0^{\mathbf{f}})^T)\mathbf{H}_1^T + \boldsymbol{\sigma}_1^{\mathbf{h}}\mathbf{\Sigma}_{\mathbf{v}}(\boldsymbol{\sigma}_1^{\mathbf{h}})^T))^{-1}.
\end{aligned}
$$

*Therefore,* $\mathbf{K}_1$ *does not have* $\epsilon$*-dependence, and we will not replace for its expanded equation. Now,*

$$
\mathbf{P}_1^l = (\mathbf{I} - \mathbf{K}_1\mathbf{H}_1)\bar{\mathbf{P}}_1 = \epsilon^2(\mathbf{I} - \mathbf{K}_1\mathbf{H}_1)(\mathbf{A}_0\mathbf{\Sigma}_{\mathbf{x}_0}\mathbf{A}_0^T + \boldsymbol{\sigma}_0^{\mathbf{f}}\mathbf{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_0^{\mathbf{f}})^T),
$$

*which shows that since* $\mathbf{P}_0^l$, $\mathbf{G}_0$ *is proportional to* $\epsilon^2$*, and* $\mathbf{M}_0$ *are proportional to* $\epsilon$*,* $\mathbf{\Sigma}_1^{\boldsymbol{\nu}}$*,* $\bar{\mathbf{P}}_1$*, and* $\mathbf{P}_1$ *are also proportional to* $\epsilon^2$*. Similarly, we can show that* $\bar{\mathbf{P}}_2$ *and* $\mathbf{\Sigma}_2^{\boldsymbol{\nu}}$ *are also proportional to* $\epsilon^2$*. Therefore, the first step of the mathematical induction is proven this way. The* $k$*-th step is also similarly proven by changing the index of* $0$ *to* $k$ *and* $1$ *to* $k+1$ *in the above equations, which is provided next. For this purpose, we only need to assume that* $\mathbf{P}_k$ *(for some* $0 \leq k \leq K-1$*) is proportional to* $\epsilon^2$ *(i.e., assume that* $\mathbf{P}_k = \epsilon^2\mathbf{\Sigma}_{\mathbf{x}_k}$ *where we have* $\mathbf{\Sigma}_{\mathbf{x}_k} := (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)(\mathbf{A}_{k-1}\mathbf{\Sigma}_{\mathbf{x}_{k-1}}\mathbf{A}_{k-1}^T + \boldsymbol{\sigma}_{k-1}^{\mathbf{f}}\mathbf{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_{k-1}^{\mathbf{f}})^T)$ *for* $1 \leq k \leq K$*), then we show that* $\mathbf{\Sigma}_{k+1}^{\boldsymbol{\nu}}$*,* $\bar{\mathbf{P}}_{k+1}$*, and* $\mathbf{P}_{k+1}$ *are also proportional to* $\epsilon^2$*:*

$$
\begin{aligned}
\bar{\mathbf{P}}_{k+1} &= \mathbf{A}_k\mathbf{P}_k^l\mathbf{A}_k^T + \mathbf{G}_k\mathbf{\Sigma}_{\mathbf{w}}\mathbf{G}_k^T = \epsilon^2(\mathbf{A}_k\mathbf{\Sigma}_{\mathbf{x}_k}\mathbf{A}_k^T + \boldsymbol{\sigma}_k^{\mathbf{f}}\mathbf{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_k^{\mathbf{f}})^T),\\
\mathbf{\Sigma}_{k+1}^{\boldsymbol{\nu}} &= \mathbf{H}_{k+1}\bar{\mathbf{P}}_{k+1}\mathbf{H}_{k+1}^T + \mathbf{M}_{k+1}\mathbf{\Sigma}_{\mathbf{v}}\mathbf{M}_{k+1}^T\\
&= \epsilon^2(\mathbf{H}_{k+1}(\mathbf{A}_k\mathbf{\Sigma}_{\mathbf{x}_k}\mathbf{A}_k^T + \boldsymbol{\sigma}_k^{\mathbf{f}}\mathbf{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_k^{\mathbf{f}})^T)\mathbf{H}_{k+1}^T + \boldsymbol{\sigma}_{k+1}^{\mathbf{h}}\mathbf{\Sigma}_{\mathbf{v}}(\boldsymbol{\sigma}_{k+1}^{\mathbf{h}})^T),
\end{aligned}
$$

$$\begin{aligned}\mathbf{K}_{k+1} =&\bar{\mathbf{P}}_{k+1}\mathbf{H}_{k+1}^T(\boldsymbol{\Sigma}_{k+1}^{\boldsymbol{\nu}})^{-1}\\ =&(\epsilon^2(\mathbf{A}_k\boldsymbol{\Sigma}_{\mathbf{x}_k}\mathbf{A}_k^T+\boldsymbol{\sigma}_k^{\mathbf{f}}\boldsymbol{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_k^{\mathbf{f}})^T))\mathbf{H}_{k+1}^T\\ &\times(\epsilon^2(\mathbf{H}_{k+1}(\mathbf{A}_k\boldsymbol{\Sigma}_{\mathbf{x}_k}\mathbf{A}_k^T+\boldsymbol{\sigma}_k^{\mathbf{f}}\boldsymbol{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_k^{\mathbf{f}})^T)\mathbf{H}_{k+1}^T+\boldsymbol{\sigma}_{k+1}^{\mathbf{h}}\boldsymbol{\Sigma}_{\mathbf{v}}(\boldsymbol{\sigma}_{k+1}^{\mathbf{h}})^T))^{-1}\\ =&(\mathbf{A}_k\boldsymbol{\Sigma}_{\mathbf{x}_k}\mathbf{A}_k^T+\boldsymbol{\sigma}_k^{\mathbf{f}}\boldsymbol{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_k^{\mathbf{f}})^T)\mathbf{H}_{k+1}^T\\ &\times((\mathbf{H}_{k+1}(\mathbf{A}_k\boldsymbol{\Sigma}_{\mathbf{x}_k}\mathbf{A}_k^T+\boldsymbol{\sigma}_k^{\mathbf{f}}\boldsymbol{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_k^{\mathbf{f}})^T)\mathbf{H}_{k+1}^T+\boldsymbol{\sigma}_{k+1}^{\mathbf{h}}\boldsymbol{\Sigma}_{\mathbf{v}}(\boldsymbol{\sigma}_{k+1}^{\mathbf{h}})^T))^{-1}.\end{aligned}$$

*Therefore, $\mathbf{K}_{k+1}$ does not have $\epsilon$-dependence, and we will not replace for its expanded equation. Now,*

$$\begin{aligned}\mathbf{P}_{k+1}^l =&(\mathbf{I}-\mathbf{K}_{k+1}\mathbf{H}_{k+1})\bar{\mathbf{P}}_{k+1}=\epsilon^2(\mathbf{I}-\mathbf{K}_{k+1}\mathbf{H}_{k+1})(\mathbf{A}_k\boldsymbol{\Sigma}_{\mathbf{x}_k}\mathbf{A}_k^T+\boldsymbol{\sigma}_k^{\mathbf{f}}\boldsymbol{\Sigma}_{\mathbf{w}}(\boldsymbol{\sigma}_k^{\mathbf{f}})^T)\\ =&\epsilon^2\boldsymbol{\Sigma}_{\mathbf{x}_{k+1}}.\end{aligned}$$

*This shows that $\boldsymbol{\Sigma}_t^{\boldsymbol{\nu}}$, $\bar{\mathbf{P}}_t$, and $\mathbf{P}_t$ for $1\le t\le K$ are always proportional to $\epsilon^2$. Moreover, we can observe that the Kalman gain is independent from the choice of $\epsilon$ for our system.*

Now, we can regroup the above equation and (6.32a) as:

$$\begin{pmatrix}\tilde{\mathbf{x}}_{t+1}^l\\ \hat{\tilde{\mathbf{x}}}_{t+1}^l\end{pmatrix}=\begin{pmatrix}\mathbf{A}_t, & -\mathbf{B}_t\mathbf{L}_t\\ \mathbf{0}, & \mathbf{A}_t-\mathbf{B}_t\mathbf{L}_t\end{pmatrix}\begin{pmatrix}\tilde{\mathbf{x}}_t^l\\ \hat{\tilde{\mathbf{x}}}_t^l\end{pmatrix}+\begin{pmatrix}\mathbf{G}_t, & \mathbf{0}\\ \mathbf{0}, & \mathbf{K}_{t+1}(\boldsymbol{\Sigma}_{t+1}^{\boldsymbol{\nu}}(\epsilon))^{\frac{1}{2}}\end{pmatrix}\begin{pmatrix}\mathbf{w}_t\\ \boldsymbol{\nu}_{t+1}\end{pmatrix}, \qquad (6.46)$$

which is a linear Gaussian system with additive noise. Now, using large deviations for the above system, for each finite $\delta\ge 0$, we have $P\{\max_{0\le t\le K}\|\tilde{\mathbf{x}}_t^l+\hat{\tilde{\mathbf{x}}}_t^l\|\ge\delta\}=o(\epsilon)$.

Let $\Omega(\delta)$ be the set where $\max_{0\le t\le K}\|\tilde{\mathbf{x}}_t^l+\hat{\tilde{\mathbf{x}}}_t^l\|\le\delta$. Then, $P(\Omega(\delta))\ge 1-o(\epsilon)$ and for $\omega\in\Omega(\delta)$, $\|\tilde{\mathbf{x}}^l+\hat{\tilde{\mathbf{x}}}^l\|_\infty=O(\delta)$. Therefore, from the calculations above, we have that $O(\|\tilde{\mathbf{x}}\|_\infty+\|\hat{\tilde{\mathbf{x}}}\|_\infty)=O(\delta)$, and hence all the other errors are also $O(\delta)$ for for $\omega\in\Omega(\delta)$.

Then for $\omega \in \Omega(\delta)$ and for all $0 \le t \le K-1$,

$$\mathbf{u}_t = \mathbf{u}_t^p + \tilde{\mathbf{u}}_t^l + O(\delta), \tag{6.47a}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^p + \tilde{\mathbf{x}}_{t+1}^l + O(\delta), \tag{6.47b}$$

$$\mathbf{z}_{t+1} = \mathbf{z}_{t+1}^p + \tilde{\mathbf{z}}_{t+1}^l + O(\delta), \tag{6.47c}$$

which means that the linear Gaussian stochastic $\tilde{(\cdot)}^l$-system along with the deterministic $p$-system can be used to control and estimate the original system given the $O(\delta)$ approximations hold. In another interpretation, the original system can be approximated for all $0 \le t \le K-1$ as:

$$\mathbf{u}_t = \mathbf{u}_t^l + O(\delta), \tag{6.48a}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^l + O(\delta), \tag{6.48b}$$

$$\mathbf{z}_{t+1} = \mathbf{z}_{t+1}^l + O(\delta). \tag{6.48c}$$

*Revisiting the mean update:* For simplicity of notations, let us define $\tilde{\mathbf{e}}_t^l := \hat{\mathbf{x}}_t^l - \mathbf{x}_t^p$ for $0 \le t \le K$ as the mean of the $l$-system's belief error, where $\tilde{\mathbf{e}}_0^l := \hat{\mathbf{x}}_0^l - \mathbf{x}_0^p = \mathbf{0}$. Then, we can rewrite the KF mean update of (6.28) as the following linear system for $0 \le t \le K-1$:

$$\tilde{\mathbf{e}}_{t+1}^l = \mathbf{T}_t^{\mathbf{e}} \tilde{\mathbf{e}}_t^l + \mathbf{T}_t^{\mathbf{u}} \tilde{\mathbf{u}}_t^l + \mathbf{T}_t^{\mathbf{z}} \tilde{\mathbf{z}}_{t+1}^l, \tag{6.49}$$

where $\mathbf{T}_t^{\mathbf{e}} := \mathbf{A}_t - \mathbf{K}_{t+1}\mathbf{H}_{t+1}\mathbf{A}_t$, $\mathbf{T}_t^{\mathbf{u}} := \mathbf{B}_t - \mathbf{K}_{t+1}\mathbf{H}_{t+1}\mathbf{B}_t$, and $\mathbf{T}_t^{\mathbf{z}} := \mathbf{K}_{t+1}$.

**Lemma 6 State Error Propagation** *For the l-system of* (6.27) *and* (6.49)*, the* *state error* $\tilde{\mathbf{x}}^l_t = \mathbf{x}^l_t - \mathbf{x}^p_t$ *for* $t \geq 1$ *can be written as follows:*

$$\tilde{\mathbf{x}}^l_{t+1} = \tilde{\mathbf{A}}^{\mathbf{x}_0}_t \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t} \tilde{\mathbf{A}}^{\mathbf{w}}_{s,t} \mathbf{w}_s + \sum_{s=0}^{t} \tilde{\mathbf{A}}^{\tilde{\mathbf{b}}}_{s,t} \tilde{\mathbf{e}}^l_s, \tag{6.50}$$

*where we have:*

- $\tilde{\mathbf{A}}_{t_1:t_2} := \Pi_{t=t_1}^{t_2} \mathbf{A}_t, t_2 \geq t_1 \geq 0$, *otherwise, it is the identity matrix;*
- $\tilde{\mathbf{A}}^{\mathbf{x}_0}_t := \tilde{\mathbf{A}}_{0:t}, t \geq 0$*;*
- $\tilde{\mathbf{A}}^{\mathbf{w}}_{s,t} := \tilde{\mathbf{A}}_{s+1:t} \mathbf{G}_s, t \geq 0, t \geq s \geq 0$*; and*
- $\tilde{\mathbf{A}}^{\tilde{\mathbf{b}}}_{s,t} := -\tilde{\mathbf{A}}_{s+1:t} \mathbf{B}_s \mathbf{L}_s, t \geq 0, t \geq s \geq 0.$

***Proof 17*** *For simplicity of notation and for the sake of the proof only, we omit the l-superscript. The reader should note that these calculations are for the l-system and not for the original system. Using the fact that* $\tilde{\mathbf{u}}_t = -\mathbf{L}_t \tilde{\mathbf{e}}_t$*, the calculations of* $\tilde{\mathbf{x}}_{t+1}$ *for* $t \geq 0$ *are as follows:*

$$\begin{aligned} \tilde{\mathbf{x}}_{t+1} &= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \mathbf{G}_t \mathbf{w}_t = \mathbf{A}_t \tilde{\mathbf{x}}_t - \mathbf{B}_t \mathbf{L}_t \tilde{\mathbf{e}}_t + \mathbf{G}_t \mathbf{w}_t \\ &=: \tilde{\mathbf{A}}_{0:t} \tilde{\mathbf{x}}_0 - \sum_{r=0}^{t} \tilde{\mathbf{A}}_{r+1:t} [\mathbf{B}_r \mathbf{L}_r \tilde{\mathbf{e}}_r - \mathbf{G}_r \mathbf{w}_r] =: \tilde{\mathbf{A}}^{\mathbf{x}_0}_t \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t} \tilde{\mathbf{A}}^{\mathbf{w}}_{s,t} \mathbf{w}_s + \sum_{s=0}^{t} \tilde{\mathbf{A}}^{\tilde{\mathbf{b}}}_{s,t} \tilde{\mathbf{e}}_s. \end{aligned}$$

**Lemma 7 Observation Error Propagation** *For the l-system of* (6.27) *and* (6.49)*, the observation error* $\tilde{\mathbf{z}}^l_t = \mathbf{z}^l_t - \mathbf{z}^p_t$ *for* $t \geq 1$ *can be written as follows:*

$$\tilde{\mathbf{z}}^l_{t+1} = \tilde{\mathbf{H}}^{\mathbf{x}_0}_{t+1} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t} \tilde{\mathbf{H}}^{\mathbf{w}}_{s,t+1} \mathbf{w}_s + \sum_{s=0}^{t} \tilde{\mathbf{H}}^{\tilde{\mathbf{b}}}_{s,t+1} \tilde{\mathbf{e}}^l_s + \mathbf{M}_{t+1} \mathbf{v}_{t+1}, \tag{6.51}$$

*where we have:*

- $\tilde{\mathbf{H}}^{\mathbf{x}_0}_{t+1} := \mathbf{H}_{t+1} \tilde{\mathbf{A}}^{\mathbf{x}_0}_t, t \geq 0$*;*
- $\tilde{\mathbf{H}}^{\mathbf{w}}_{s,t+1} := \mathbf{H}_{t+1} \tilde{\mathbf{A}}^{\mathbf{w}}_{s,t}, t \geq 0, t \geq s \geq 0$*; and*

- $\tilde{\mathbf{H}}^{\tilde{\mathbf{b}}}_{s,t+1} := \mathbf{H}_{t+1}\tilde{\mathbf{A}}^{\tilde{\mathbf{b}}}_{s,t}, t \geq 0, t \geq s \geq 0.$

***Proof 18*** *For simplicity of notation and for the sake of the proof only, we omit the $l$-superscript. The reader should note that these calculations are for the $l$-system and not for the original system.*

$$\begin{aligned}\tilde{\mathbf{z}}_{t+1} =& \mathbf{H}_{t+1}\tilde{\mathbf{x}}_{t+1} + \mathbf{M}_{t+1}\mathbf{v}_{t+1} = \mathbf{H}_{t+1}(\tilde{\mathbf{A}}^{\mathbf{x}_0}_t\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t}\tilde{\mathbf{A}}^{\mathbf{w}}_{s,t}\mathbf{w}_s + \sum_{s=0}^{t}\tilde{\mathbf{A}}^{\tilde{\mathbf{b}}}_{s,t}\tilde{\mathbf{e}}_s) + \mathbf{M}_{t+1}\mathbf{v}_{t+1}\\ &=: \tilde{\mathbf{H}}^{\mathbf{x}_0}_{t+1}\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t}\tilde{\mathbf{H}}^{\mathbf{w}}_{s,t+1}\mathbf{w}_s + \sum_{s=0}^{t}\tilde{\mathbf{H}}^{\tilde{\mathbf{b}}}_{s,t+1}\tilde{\mathbf{e}}_s + \mathbf{M}_{t+1}\mathbf{v}_{t+1}.\end{aligned}$$

*Solving a recursion:* In order to prove the next lemma, we first need to solve a similar linear recursion that will be used here and later in the chapter. Let us define the recursion equation as:

$$\mathbf{x}_{t+1} = \sum_{s=0}^{t}\mathbf{U}_{s,t}\mathbf{x}_s + \mathbf{f}_t, \tag{6.52}$$

for some given $\mathbf{f}_s, 0 \leq s \leq t$ and $\mathbf{x}_0$. We first write a few iterations to obtain the general formula:

$$\begin{aligned}\mathbf{x}_{t+1} =& \sum_{s=0}^{t}\mathbf{U}_{s,t}\mathbf{x}_s + \mathbf{f}_t = \mathbf{U}_{t,t}\mathbf{x}_t + \sum_{s=0}^{t-1}\mathbf{U}_{s,t}\mathbf{x}_s + \mathbf{f}_t\\ =&\mathbf{U}_{t,t}(\sum_{s=0}^{t-1}\mathbf{U}_{s,t-1}\mathbf{x}_s + \mathbf{f}_{t-1}) + \sum_{s=0}^{t-1}\mathbf{U}_{s,t}\mathbf{x}_s + \mathbf{f}_t\\ =&\sum_{s=0}^{t-1}(\mathbf{U}_{s,t} + \mathbf{U}_{t,t}\mathbf{U}_{s,t-1})\mathbf{x}_s + \mathbf{U}_{t,t}\mathbf{f}_{t-1} + \mathbf{f}_t\\ =&(\mathbf{U}_{t-1,t} + \mathbf{U}_{t,t}\mathbf{U}_{t-1,t-1})\mathbf{x}_{t-1} + \sum_{s=0}^{t-2}(\mathbf{U}_{s,t} + \mathbf{U}_{t,t}\mathbf{U}_{s,t-1})\mathbf{x}_s + \mathbf{U}_{t,t}\mathbf{f}_{t-1} + \mathbf{f}_t\\ =&(\mathbf{U}_{t-1,t} + \mathbf{U}_{t,t}\mathbf{U}_{t-1,t-1})(\sum_{s=0}^{t-2}\mathbf{U}_{s,t-2}\mathbf{x}_s + \mathbf{f}_{t-2})\\ &+ \sum_{s=0}^{t-2}(\mathbf{U}_{s,t} + \mathbf{U}_{t,t}\mathbf{U}_{s,t-1})\mathbf{x}_s + \mathbf{U}_{t,t}\mathbf{f}_{t-1} + \mathbf{f}_t\end{aligned}$$

$$\begin{aligned}
&= \sum_{s=0}^{t-2}(\mathbf{U}_{t-1,t} + \mathbf{U}_{t,t}\mathbf{U}_{t-1,t-1})\mathbf{U}_{s,t-2}\mathbf{x}_s + (\mathbf{U}_{t-1,t} + \mathbf{U}_{t,t}\mathbf{U}_{t-1,t-1})\mathbf{f}_{t-2} \\
&\quad + \sum_{s=0}^{t-2}(\mathbf{U}_{s,t} + \mathbf{U}_{t,t}\mathbf{U}_{s,t-1})\mathbf{x}_s + \mathbf{U}_{t,t}\mathbf{f}_{t-1} + \mathbf{f}_t \\
&= \sum_{s=0}^{t-2}((\mathbf{U}_{t-1,t} + \mathbf{U}_{t,t}\mathbf{U}_{t-1,t-1})\mathbf{U}_{s,t-2} + (\mathbf{U}_{s,t} + \mathbf{U}_{t,t}\mathbf{U}_{s,t-1}))\mathbf{x}_s \\
&\quad + (\mathbf{U}_{t-1,t} + \mathbf{U}_{t,t}\mathbf{U}_{t-1,t-1})\mathbf{f}_{t-2} + \mathbf{U}_{t,t}\mathbf{f}_{t-1} + \mathbf{f}_t.
\end{aligned}$$

From these few iterations, we can define the following polynomial for $1 \leq s \leq t$:

$$\mathbf{Q}_s := \sum_{r=0}^{s-1} \mathbf{Q}_r \mathbf{U}_{t-s+1,t-r}, \tag{6.53}$$

where $\mathbf{Q}_0 := 1$. Then, (6.52) can be written as:

$$\begin{aligned}
\mathbf{x}_{t+1} &= (\sum_{s=0}^{t} \mathbf{Q}_s \mathbf{U}_{0,t-s})\mathbf{x}_0 + \sum_{r=0}^{t} \mathbf{Q}_r \mathbf{f}_{t-r} \\
&= (\sum_{s=0}^{t} \mathbf{Q}_s \mathbf{U}_{0,t-s})\mathbf{x}_0 + \sum_{s=0}^{t} \mathbf{Q}_{t-s} \mathbf{f}_s, \qquad (6.54)
\end{aligned}$$

where we used $s = t - r$ in the last equation. This can easily be proven with the mathematical induction. Instead, we provide the intuitive process that led to the formula. First, note that the last there terms in the last iteration written above hint at the construction of the polynomial $\mathbf{Q}_s$ as:

$$\begin{aligned}
\mathbf{Q}_0 &:= 1, \\
\mathbf{Q}_1 &:= \mathbf{Q}_0 \mathbf{U}_{t,t}, \\
\mathbf{Q}_2 &:= \mathbf{Q}_0 \mathbf{U}_{t-1,t} + \mathbf{Q}_1 \mathbf{U}_{t-1,t-1}, \\
\mathbf{Q}_3 &:= \mathbf{Q}_0 \mathbf{U}_{t-2,t} + \mathbf{Q}_1 \mathbf{U}_{t-2,t-1} + \mathbf{Q}_2 \mathbf{U}_{t-2,t-2},
\end{aligned}$$

from which the formula for $\mathbf{Q}_s$ can be derived as

$$\mathbf{Q}_s = \mathbf{Q}_0\mathbf{U}_{t-s+1,t} + \mathbf{Q}_1\mathbf{U}_{t-s+1,t-1} + \cdots + \mathbf{Q}_{s-1}\mathbf{U}_{t-s+1,t-s+1},$$

Also the wighted summation of the $\mathbf{f}_{t-s}$ can be derived, as stated above. Next, to derive the weighted summation formula for the $\mathbf{x}_s$ in the last iterative equation, we note that the weight of $\mathbf{x}_s$ can be written as:

$$\mathbf{Q}_0\mathbf{U}_{s,t} + \mathbf{Q}_1\mathbf{U}_{s,t-1} + \mathbf{Q}_2\mathbf{U}_{s,t-2}.$$

Hence, that summation can be rewritten as

$$\sum_{s=0}^{t-2}(\mathbf{Q}_0\mathbf{U}_{s,t} + \mathbf{Q}_1\mathbf{U}_{s,t-1} + \mathbf{Q}_2\mathbf{U}_{s,t-2})\mathbf{x}_s,$$

where the goal is to eliminate the summation to only the term $s = 0$. Therefore, from the pattern above we reach to the following formula:

$$(\mathbf{Q}_0\mathbf{U}_{s,t} + \mathbf{Q}_1\mathbf{U}_{s,t-1} + \mathbf{Q}_2\mathbf{U}_{s,t-2} + \cdots + \mathbf{Q}_t\mathbf{U}_{s,t-t})\mathbf{x}_0,$$

where we set $s = 0$ and reach to the formula stated above.

*A summation exchange formula:* We also need the following summation exchange formula for the next lemma:

$$\sum_{s=0}^{t}\sum_{r=0}^{s} f_{s,r}x_r = \sum_{r=0}^{t}\sum_{s=r}^{t} f_{s,r}x_r = \sum_{r=0}^{t}(\sum_{s=r}^{t} f_{s,r})x_r = \sum_{s=0}^{t}(\sum_{r=s}^{t} f_{r,s})x_s.$$

**Lemma 8 Mean Error Propagation** *For the $l$-system of* (6.27) *and* (6.49)*, the mean error $\tilde{\mathbf{e}}_t^l = \hat{\mathbf{x}}_t^l - \mathbf{x}_t^p$ for $t \geq 0$ 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:*

$$\tilde{\mathbf{e}}^l_{t+1} = \tilde{\mathbf{T}}^{\mathbf{x}_0}_t \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t} \tilde{\mathbf{T}}^{\mathbf{w}}_{s,t} \mathbf{w}_s + \sum_{s=0}^{t} \tilde{\mathbf{T}}^{\mathbf{v}}_{s,t} \mathbf{v}_{s+1}, \tag{6.55}$$

*where we have:*

- $\mathbf{T}^{\mathbf{L}}_{s,t} := \mathbf{T}^{\mathbf{z}}_t \tilde{\mathbf{H}}^{\tilde{\mathbf{b}}}_{s,t+1}, 0 \leq s \leq t-1, 0 \leq t \leq K-1$*;*
- $\mathbf{T}^{\mathbf{L}}_{s,t} := \mathbf{T}^{\mathbf{e}}_t - \mathbf{T}^{\mathbf{u}}_t \mathbf{L}_t + \mathbf{T}^{\mathbf{z}}_t \tilde{\mathbf{H}}^{\tilde{\mathbf{b}}}_{t,t+1}, s = t, 0 \leq t \leq K-1$*;*
- $\mathbf{Q}_s := \sum_{r=0}^{s-1} \mathbf{Q}_r \mathbf{T}^{\mathbf{L}}_{t-s+1,t-r}, 1 \leq s \leq t, 0 \leq t \leq K-1$*, and* $\mathbf{Q}_0 = 1$*;*
- $\tilde{\mathbf{T}}^{\mathbf{x}_0}_t := \sum_{s=0}^{t} \mathbf{Q}_{t-s} \mathbf{T}^{\mathbf{z}}_s \tilde{\mathbf{H}}^{\mathbf{x}_0}_{s+1}, 0 \leq t \leq K-1$*;*
- $\tilde{\mathbf{T}}^{\mathbf{w}}_{s,t} := \sum_{r=s}^{t} \mathbf{Q}_{t-r} \mathbf{T}^{\mathbf{z}}_r \tilde{\mathbf{H}}^{\mathbf{w}}_{s,r+1}, 0 \leq s \leq t, 0 \leq t \leq K-1$*; and*
- $\tilde{\mathbf{T}}^{\mathbf{v}}_{s,t} := \mathbf{Q}_{t-s} \mathbf{T}^{\mathbf{z}}_s \mathbf{M}_{s+1}, 0 \leq s \leq t, 0 \leq t \leq K-1.$

***Proof 19*** *For simplicity of notation and for the sake of the proof only, we omit the* $l$*-superscript. The reader should note that these calculations are for the* $l$*-system and not for the original system. First note* $\tilde{\mathbf{e}}_0 = 0$*. Now, we can rewrite the mean error for* $t \geq 0$ *as:*

$$\begin{aligned}
\tilde{\mathbf{e}}_{t+1} =& \mathbf{T}^{\mathbf{e}}_t \tilde{\mathbf{e}}_t + \mathbf{T}^{\mathbf{u}}_t \tilde{\mathbf{u}}_t + \mathbf{T}^{\mathbf{z}}_t \tilde{\mathbf{z}}_{t+1} \\
=& (\mathbf{T}^{\mathbf{e}}_t - \mathbf{T}^{\mathbf{u}}_t \mathbf{L}_t) \tilde{\mathbf{e}}_t + \mathbf{T}^{\mathbf{z}}_t (\tilde{\mathbf{H}}^{\mathbf{x}_0}_{t+1} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t} \tilde{\mathbf{H}}^{\mathbf{w}}_{s,t+1} \mathbf{w}_s + \sum_{s=0}^{t} \tilde{\mathbf{H}}^{\tilde{\mathbf{b}}}_{s,t+1} \tilde{\mathbf{e}}_s + \mathbf{M}_{t+1} \mathbf{v}_{t+1}) \\
=& (\mathbf{T}^{\mathbf{e}}_t - \mathbf{T}^{\mathbf{u}}_t \mathbf{L}_t) \tilde{\mathbf{e}}_t + \mathbf{T}^{\mathbf{z}}_t \tilde{\mathbf{H}}^{\tilde{\mathbf{b}}}_{t,t+1} \tilde{\mathbf{e}}_t \\
& + \mathbf{T}^{\mathbf{z}}_t \tilde{\mathbf{H}}^{\mathbf{x}_0}_{t+1} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t} \mathbf{T}^{\mathbf{z}}_t \tilde{\mathbf{H}}^{\mathbf{w}}_{s,t+1} \mathbf{w}_s + \sum_{s=0}^{t-1} \mathbf{T}^{\mathbf{z}}_t \tilde{\mathbf{H}}^{\tilde{\mathbf{b}}}_{s,t+1} \tilde{\mathbf{e}}_s + \mathbf{T}^{\mathbf{z}}_t \mathbf{M}_{t+1} \mathbf{v}_{t+1} \\
=:& \sum_{s=0}^{t} \mathbf{T}^{\mathbf{L}}_{s,t} \tilde{\mathbf{e}}_s + \mathbf{T}^{\mathbf{z}}_t \tilde{\mathbf{H}}^{\mathbf{x}_0}_{t+1} \tilde{\mathbf{x}}_0 + \sum_{s=0}^{t} \mathbf{T}^{\mathbf{z}}_t \tilde{\mathbf{H}}^{\mathbf{w}}_{s,t+1} \mathbf{w}_s + \mathbf{T}^{\mathbf{z}}_t \mathbf{M}_{t+1} \mathbf{v}_{t+1} \\
=& (\sum_{s=0}^{t} \mathbf{Q}_s \mathbf{T}^{\mathbf{L}}_{0,t-s}) \tilde{\mathbf{e}}_0 + (\sum_{s=0}^{t} \mathbf{Q}_{t-s} \mathbf{T}^{\mathbf{z}}_s \tilde{\mathbf{H}}^{\mathbf{x}_0}_{s+1}) \tilde{\mathbf{x}}_0
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{s=0}^{t}\sum_{r=0}^{s}\mathbf{Q}_{t-s}\mathbf{T}^{\mathbf{z}}_{s}\tilde{\mathbf{H}}^{\mathbf{w}}_{r,s+1}\mathbf{w}_{r}+\sum_{s=0}^{t}\mathbf{Q}_{t-s}\mathbf{T}^{\mathbf{z}}_{s}\mathbf{M}_{s+1}\mathbf{v}_{s+1}\\
=&(\sum_{s=0}^{t}\mathbf{Q}_{t-s}\mathbf{T}^{\mathbf{z}}_{s}\tilde{\mathbf{H}}^{\mathbf{x}_0}_{s+1})\tilde{\mathbf{x}}_0+\sum_{s=0}^{t}\sum_{r=s}^{t}(\mathbf{Q}_{t-r}\mathbf{T}^{\mathbf{z}}_{r}\tilde{\mathbf{H}}^{\mathbf{w}}_{s,r+1})\mathbf{w}_{s}+\sum_{s=0}^{t}\mathbf{Q}_{t-s}\mathbf{T}^{\mathbf{z}}_{s}\mathbf{M}_{s+1}\mathbf{v}_{s+1}\\
=:&\ \tilde{\mathbf{T}}^{\mathbf{x}_0}_{t}\tilde{\mathbf{x}}_0+\sum_{s=0}^{t}\tilde{\mathbf{T}}^{\mathbf{w}}_{s,t}\mathbf{w}_{s}+\sum_{s=0}^{t}\tilde{\mathbf{T}}^{\mathbf{v}}_{s,t}\mathbf{v}_{s+1}.
\end{aligned}
$$

### *6.2.1 Analysis of the Cost*

In this section, we use the more general definition of the cost function directly in terms of the state, and try to approximate the cost function of the original system in terms of the cost of the $l$-system.

*Cost function:* We consider the most general cost function:

$$
J_{\boldsymbol{\pi}} := \sum_{t=0}^{K-1} c_t^{\boldsymbol{\pi}}(\mathbf{x}_t, \mathbf{u}_t) + c_K^{\boldsymbol{\pi}}(\mathbf{x}_K), \tag{6.56}
$$

which we linearize around the nominal system:

$$
\begin{aligned}
J &= J^p + \tilde{J}_1 + o(\sum_{t=1}^{K-1}(\|\tilde{\mathbf{x}}_t\| + \|\tilde{\mathbf{u}}_t\|) + \|\tilde{\mathbf{x}}_K\|) && \text{(6.57a)}\\
&= J^p + \tilde{J}_1 + o(\|\tilde{\mathbf{x}}\|_\infty). && \text{(6.57b)}
\end{aligned}
$$

We assume that the cost function is continuously differentiable and bounded. Let $|c_t| \leq M$ and $|c_K| \leq M$ for some $M > 0$. Moreover,

- $\mathbf{C}^{\mathbf{x}}_t = \nabla_{\mathbf{x}} c_t(\mathbf{x},\mathbf{u})|_{\mathbf{x}^p_t,\mathbf{u}^p_t}$, $\mathbf{C}^{\mathbf{u}}_t = \nabla_{\mathbf{u}} c_t(\mathbf{x},\mathbf{u})|_{\mathbf{x}^p_t,\mathbf{u}^p_t}$, $\mathbf{C}^{\mathbf{x}}_K = \nabla_{\mathbf{x}} c_K(\mathbf{x})|_{\mathbf{x}^p_K}$;
- $J^p := \sum_{t=0}^{K-1} c_t(\mathbf{x}^p_t, \mathbf{u}^p_t) + c_K(\mathbf{x}^p_K)$ denotes the nominal cost;
- $\tilde{J}_1 := \sum_{t=0}^{K-1}(\mathbf{C}^{\mathbf{x}}_t\tilde{\mathbf{x}}_t + \mathbf{C}^{\mathbf{u}}_t\tilde{\mathbf{u}}_t) + \mathbf{C}^{\mathbf{x}}_K\tilde{\mathbf{x}}_K$ is the first order error in the cost; and
- $J_1 := J^p + \tilde{J}_1$ is the first order approximation of the cost function.

Therefore, for $\omega \in \Omega(\delta)$, and

$$J = J^p + \sum_{t=0}^{K-1} (\mathbf{C}_t^{\mathbf{x}} \tilde{\mathbf{x}}_t + \mathbf{C}_t^{\mathbf{u}} \tilde{\mathbf{u}}_t) + \mathbf{C}_K^{\mathbf{x}} \tilde{\mathbf{x}}_K + O(\delta) \tag{6.58a}$$

$$= J^p + \sum_{t=0}^{K-1} (\mathbf{C}_t^{\mathbf{x}} \tilde{\mathbf{x}}_t^l + \mathbf{C}_t^{\mathbf{u}} \tilde{\mathbf{u}}_t^l) + \mathbf{C}_K^{\mathbf{x}} \tilde{\mathbf{x}}_K^l + O(\delta). \tag{6.58b}$$

The above calculations show that the cost of the original system is close to the cost of the $l$-system. Moreover, $J - J_1 = O(\delta)$ for $\omega \in \Omega(\delta)$.

Next, we provide the main result regarding the expected first order error of the cost function.

**Theorem 8 First-Order Cost Function Error for a Partially-observed System with T-LQG Policy:** *Given that process and observation noises are zero mean i.i.d. Gaussian, the initial error is zero mean Gaussian, and all the functions are in $\mathbb{C}^1$, under a first-order approximation for the small noise paradigm, the stochastic cost function is dominated by the nominal part of the cost function, and the expected first-order error is $O(\delta)$. That is,*

$$\mathbb{E}[\tilde{J}_1] = O(\delta), \ \textit{and}\ \mathbb{E}[J] = J^p + O(\delta).$$

*Moreover, by choosing $\delta = \sqrt{\log(\frac{1}{\epsilon})}\epsilon$, we have*

$$\mathbb{E}[\tilde{J}_1] = O(\epsilon^{1-\gamma}), \ \textit{and}\ \mathbb{E}[J] = J^p + O(\epsilon^{1-\gamma}),$$

*for some $0 < \gamma \ll 1$, which shows that this error tends to zero with a near-first-order rate as $\epsilon \downarrow 0$.*

***Proof 20*** *Let* $\tilde{J}_1^l{:=}\sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t^l{+}\mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t^l)+\mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K^l$. *Then,*

$$\tilde{J}_1^l := \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t^l{+}\mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t^l)+\mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K^l = \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t^l{-}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_t\tilde{\mathbf{e}}_t^l)+\mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K^l.$$

*Also note* $\mathbb{E}[\tilde{\mathbf{x}}_0^l] = \mathbb{E}[\tilde{\mathbf{x}}_0] = \mathbb{E}[\mathbf{x}_0 - \hat{\mathbf{x}}_0] = \mathbf{0}$, $\tilde{\mathbf{e}}_0^l = \mathbf{0}$, *and* $\mathbb{E}[\mathbf{w}_t] = \mathbb{E}[\mathbf{v}_t] = 0$ *for all* $t$. *Then, we use Lemmas 6, 7, and 8. First, we calculate* $\mathbb{E}[\tilde{\mathbf{e}}_t^l], 1 \leq t \leq K$*:*

$$\mathbb{E}[\tilde{\mathbf{e}}_t^l] = \tilde{\mathbf{T}}_t^{\mathbf{x}_0}\mathbb{E}[\tilde{\mathbf{x}}_0] + \sum_{s=0}^{t}\tilde{\mathbf{T}}_{s,t}^{\mathbf{w}}\mathbb{E}[\mathbf{w}_s] + \sum_{s=0}^{t}\tilde{\mathbf{T}}_{s,t}^{\mathbf{v}}\mathbb{E}[\mathbf{v}_{s+1}] = \mathbf{0}.$$

*Then, we calculate* $\mathbb{E}[\tilde{\mathbf{x}}_{t+1}^l], 0 \leq t \leq K-1$*:*

$$\mathbb{E}[\tilde{\mathbf{x}}_{t+1}^l] = \tilde{\mathbf{A}}_t^{\mathbf{x}_0}\mathbb{E}[\tilde{\mathbf{x}}_0] + \sum_{s=0}^{t}\tilde{\mathbf{A}}_{s,t}^{\mathbf{w}}\mathbb{E}[\mathbf{w}_s] + \sum_{s=0}^{t}\tilde{\mathbf{A}}_{s,t}^{\tilde{\mathbf{b}}}\mathbb{E}[\tilde{\mathbf{e}}_s] = \mathbf{0}.$$

*Therefore, we have:*

$$\mathbb{E}[\tilde{J}_1^l]{=}\sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\mathbb{E}[\tilde{\mathbf{x}}_t^l] - \mathbf{C}_t^{\mathbf{u}}\mathbf{L}_t\mathbb{E}[\tilde{\mathbf{e}}_t^l]){+}\mathbf{C}_K^{\mathbf{x}}\mathbb{E}[\tilde{\mathbf{x}}_K^l] = 0.$$

*Now, we take expectation of both sides of* (6.58b). *Since, for* $\omega \notin \Omega(\delta)$, $J \leq M$, *then*

$$\begin{aligned}\mathbb{E}[J - J^p] &= P(\Omega(\delta))(\mathbb{E}[\tilde{J}_1^l] + O(\delta)) + M(1 - P(\Omega(\delta)))\\ &= P(\Omega(\delta))O(\delta) + M(1 - P(\Omega(\delta)))\end{aligned} \tag{6.59}$$

*Now, the last expression is the same as* (4.19)*. Although* $\Omega(\delta)$ *is not the same as in Theorem 2,* $P(\Omega(\delta))$ *is still the same. In the proof of Theorem 2 while we discussed on the probabilistic argument and choosing the proper* $\delta$*, we showed that by choosing* $\delta := \sqrt{-\log(\epsilon)}\epsilon$*, the* $\mathbb{E}[J - J^p] = O(\epsilon^{1-\gamma})$*. The same argument follows through and this theorem is proved.*

Hence, the expected stochastic cost is equal to the nominal cost with a high probability as $\epsilon \downarrow 0$. Therefore, it follows that the open-loop nominal design can be done decoupled from the closed-loop design, summarized below:

**Corollary 6 Decoupling Principle: Decoupling of the Open-Loop and Closed-Loop Designs Under Small Noise.** *Based on Theorem 8, for a partially-observed system where the function are in $\mathbb{C}^1$ under the small noise paradigm, as $\epsilon \downarrow 0$, the design of the feedback law can be decoupled from the design of the open-loop optimized trajectory.If the functions are in $\mathbb{C}^1$, this result is $O(\epsilon^{1-\gamma})$-optimal for $0 < \gamma \ll 1$ as $\epsilon \downarrow 0$.*

***Proof 21*** *Using Theorem 8, for $\omega \in \Omega(\delta)$ we have $\mathbb{E}[J] = J^p + O(\epsilon^{1-\gamma})$, which is the cost of applying policy $\boldsymbol{\pi}_t(\mathbf{z}_{0:t}) = \mathbf{u}_t^p - \mathbf{L}_t(\hat{\mathbf{x}}_t - \mathbf{x}_t^p)$ to the stochastic system (note that $\hat{\mathbf{x}}_t$ is a function of $\mathbf{z}_{0:t}$). Now, suppose $\boldsymbol{\pi}^*$ is the optimal stochastic policy. We showed in the proof of Corollary 5 that for this policy, we have $\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{1-\gamma})$. Now, by construction $J^p \leq J^{*p}$, and*

$$\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{1-\gamma}) \geq J^p + O(\epsilon^{1-\gamma}) = \mathbb{E}[J_{\boldsymbol{\pi}}] + O(\epsilon^{1-\gamma})$$

*As a result, policy $\boldsymbol{\pi}$ is within $O(\epsilon^{1-\gamma})$ of the optimal stochastic policy.*

## 6.3 Near-Second-Order Optimality of The Deterministic Law

In this section, we provide a second-order analysis of the deterministic feedback law and show that applying the optimal feedback law of the deterministic problem to the stochastic problem results in a near-second-order optimality as well. Therefore, we improve the results of Section 6.1.

*Assumptions:* Other than the assumptions of Section 6.1, we assume for the analysis of this section that all the functions (including the dynamics and observation models, feedback law, and the cost functions) are in $\mathbb{C}^2$, i.e., they are continuously differentiable to the second-order.

*Second-order expansion of the control law:* Here, we will use the same policy $\mathbf{u}_t = \boldsymbol{\pi}_t^d(\mathbf{z}_{0:t})$ defined in Section 4.4. However, as opposed to that section, for the analysis of this section we expand this law to the second-order. Let us define $\mathbf{u}_t^p := \boldsymbol{\pi}_t^d(\mathbf{z}_{0:t}^p)$, $\tilde{\mathbf{u}}_t := \mathbf{u}_t - \mathbf{u}_t^p$ and $\tilde{\mathbf{x}}_t$ and $\tilde{\mathbf{z}}_t$ as before. Then,

$$\begin{aligned}\tilde{\mathbf{u}}_t &= \boldsymbol{\pi}_t^d(\mathbf{z}_{0:t}) - \boldsymbol{\pi}_t^d(\mathbf{z}_{0:t}^p) && \text{(6.60a)}\\ &= -\sum_{s=0}^{t} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s + \sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t} \tilde{\mathbf{z}}_i^T \mathbf{H}_t^{\pi^{kij}} \tilde{\mathbf{z}}_j \mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}_t\|^2 + \sum_{s=0}^{t}\|\tilde{\mathbf{z}}_s\|^2) && \text{(6.60b)}\\ &= -\sum_{s=0}^{t} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s + \sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t} \tilde{\mathbf{z}}_i^T \mathbf{H}_t^{\pi^{kij}} \tilde{\mathbf{z}}_j \mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{z}}\|_\infty^2), && \text{(6.60c)}\end{aligned}$$

as $(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{z}}\|_\infty^2) \downarrow 0$, where we have:

- $\mathbf{L}_{s,t} := -\nabla_{\mathbf{z}_s}\boldsymbol{\pi}_t^d(\mathbf{z}_{0:t})|_{\mathbf{z}_{0:t}^p}$;
- $\boldsymbol{\pi}_t^d(\mathbf{z}_{0:t}) = (\pi^{d_k}(\mathbf{z}_{0:t})), 1 \leq k \leq n_u$;
- $\mathbf{H}_t^{\pi^{kij}} := \frac{1}{2}\nabla^2_{\mathbf{z}_i\mathbf{z}_j}\boldsymbol{\pi}_t^{d_k}(\mathbf{z}_{0:t})|_{\mathbf{z}_{0:t}^p}$;
- $\tilde{\mathbf{x}}_0 = \mathbf{x}_0 - \mathbf{x}_0^p$, and $\tilde{\mathbf{z}}_0 = \mathbf{z}_0 - \mathbf{z}_0^p$.

Also note the simplified from of the second-order terms comes from the fact that we can simplify the following expression:

$$\begin{pmatrix}\tilde{\mathbf{z}}_0\\ \vdots\\ \tilde{\mathbf{z}}_t\end{pmatrix}^T \begin{pmatrix}\mathbf{H}_t^{\pi^{k00}}, \mathbf{H}_t^{\pi^{k01}}, \cdots, \mathbf{H}_t^{\pi^{k0t}}\\ \vdots\\ \mathbf{H}_t^{\pi^{kt0}}, \mathbf{H}_t^{\pi^{kt1}}, \cdots, \mathbf{H}_t^{\pi^{ktt}}\end{pmatrix} \begin{pmatrix}\tilde{\mathbf{z}}_0\\ \vdots\\ \tilde{\mathbf{z}}_t\end{pmatrix} = \begin{pmatrix}\tilde{\mathbf{z}}_0\\ \vdots\\ \tilde{\mathbf{z}}_t\end{pmatrix}^T \begin{pmatrix}\sum_{j=0}^t \mathbf{H}_t^{\pi^{k0j}}\tilde{\mathbf{z}}_j\\ \vdots\\ \sum_{j=0}^t \mathbf{H}_t^{\pi^{ktj}}\tilde{\mathbf{z}}_j\end{pmatrix} = \sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{z}}_i^T\mathbf{H}_t^{\pi^{kij}}\tilde{\mathbf{z}}_j.$$

Therefore, the second-order term is indeed the following:

$$\begin{pmatrix} \sum_{i=0}^{t}\sum_{j=0}^{t} \tilde{\mathbf{z}}_i \mathbf{H}_t^{\pi^{00j}} \tilde{\mathbf{z}}_j \\ \vdots \\ \sum_{i=0}^{t}\sum_{j=0}^{t} \tilde{\mathbf{z}}_i \mathbf{H}_t^{\pi^{n_u tj}} \tilde{\mathbf{z}}_j \end{pmatrix} = \sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t} \tilde{\mathbf{z}}_i^T \mathbf{H}_t^{\pi^{kij}} \tilde{\mathbf{z}}_j \mathbf{e}_k^{n_u}.$$

*Second-order expansion of the system equations:* We obtain the second-order expansion of the process model around the nominal trajectory, for $0 \leq t \leq K-1$:

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) - \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p) + \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}} \mathbf{w}_t \tag{6.61a}$$

$$= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}} \mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,1} & \mathbf{F}_{\mathbf{xu}}^{t,1} \\ \mathbf{F}_{\mathbf{ux}}^{t,1} & \mathbf{F}_{\mathbf{uu}}^{t,1} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,n_x} & \mathbf{F}_{\mathbf{xu}}^{t,n_x} \\ \mathbf{F}_{\mathbf{ux}}^{t,n_x} & \mathbf{F}_{\mathbf{uu}}^{t,n_x} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}_t\|^2 + \|\tilde{\mathbf{u}}_t\|^2) \tag{6.61b}$$

$$= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \mathbf{G}_t \mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,1} & \mathbf{F}_{\mathbf{xu}}^{t,1} \\ \mathbf{F}_{\mathbf{ux}}^{t,1} & \mathbf{F}_{\mathbf{uu}}^{t,1} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,n_x} & \mathbf{F}_{\mathbf{xu}}^{t,n_x} \\ \mathbf{F}_{\mathbf{ux}}^{t,n_x} & \mathbf{F}_{\mathbf{uu}}^{t,n_x} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2), \tag{6.61c}$$

$$\tilde{\mathbf{z}}_{t+1} = \mathbf{h}(\mathbf{x}_{t+1}) - \mathbf{h}(\mathbf{x}_{t+1}^p) + \epsilon \boldsymbol{\sigma}_{t+1}^{\mathbf{h}} \mathbf{v}_{t+1} \tag{6.61d}$$

$$= \mathbf{H}_{t+1} \tilde{\mathbf{x}}_{t+1} + \epsilon \boldsymbol{\sigma}_{t+1}^{\mathbf{h}} \mathbf{v}_{t+1} + \sum_{j=1}^{n_z} (\tilde{\mathbf{x}}_{t+1}^T \mathbf{H}_{t+1}^{h^j} \tilde{\mathbf{x}}_{t+1}) \mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}_{t+1}\|^2) \tag{6.61e}$$

$$= \mathbf{H}_{t+1} \tilde{\mathbf{x}}_{t+1} + \mathbf{M}_{t+1} \mathbf{v}_{t+1} + \sum_{j=1}^{n_z} (\tilde{\mathbf{x}}_{t+1}^T \mathbf{H}_{t+1}^{h^j} \tilde{\mathbf{x}}_{t+1}) \mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}\|_\infty^2), \tag{6.61f}$$

as ($\|\tilde{\mathbf{x}}\|_\infty + (\|\tilde{\mathbf{u}}\|_\infty) \downarrow 0$, where we have:

- $\mathbf{A}_t := \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{B}_t := \nabla_{\mathbf{u}} \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{G}_t := \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}}$;
- $\mathbf{H}_t := \nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x})|_{\mathbf{x}_t^p}$, $\mathbf{M}_t := \epsilon \boldsymbol{\sigma}_t^{\mathbf{h}}$.
- $\mathbf{f}(\mathbf{x}, \mathbf{u}) = (f^j(\mathbf{x}, \mathbf{u})), 1 \leq j \leq n_x$;
- $\mathbf{F}_{\mathbf{xx}}^{t,j} := \frac{1}{2} \nabla^2_{\mathbf{xx}} f^j(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}, \mathbf{F}_{\mathbf{xu}}^{t,j} := \frac{1}{2} \nabla^2_{\mathbf{xu}} f^j(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}, \mathbf{F}_{\mathbf{ux}}^{t,j} := \frac{1}{2} \nabla^2_{\mathbf{ux}} f^j(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, and $\mathbf{F}_{\mathbf{uu}}^{t,j} := \frac{1}{2} \nabla^2_{\mathbf{uu}} f^j(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$;
- $\mathbf{h}(\mathbf{x}) = (h^j(\mathbf{x})), 1 \leq j \leq n_z$;
- $\mathbf{H}_t^{h^j} := \frac{1}{2} \nabla^2_{\mathbf{xx}} h^j(\mathbf{x})|_{\mathbf{x}_t^p}$.

*Feedback compensation:* Next, we replace the feedback law of (6.60c) into (6.61c). Note that after th feedback compensation, the first-order terms of (6.61c) which are linear in $\tilde{\mathbf{u}}_t$, result in both first-order and second-order expressions in $\tilde{\mathbf{x}}_t$. That is because, according to (6.61f), the observations can be written in terms of $\tilde{\mathbf{x}}_t$. On the other hand, replacing the second-order terms of the feedback law into the second-order terms of the dynamics in (4.43c) results in second-, third- and fourth-order expressions in $\tilde{\mathbf{x}}_t$. However, since the error term in (6.61c) includes $o(\|\tilde{\mathbf{x}}\|^2_\infty)$, the third- and fourth-order terms can be ignored. As a result, just like the fully-observed case of (4.44), we replace those terms with $o(\|\tilde{\mathbf{x}}\|^2_\infty)$.

Next, we simplify the second-order expansion of the control error:

$$\begin{aligned}
\tilde{\mathbf{u}}_t &= -\sum_{s=0}^{t} \mathbf{L}_{s,t} \tilde{\mathbf{z}}_s + \sum_{k=1}^{n_u} \sum_{i=0}^{t} \sum_{j=0}^{t} \tilde{\mathbf{z}}_i^T \mathbf{H}_t^{\pi^{kij}} \tilde{\mathbf{z}}_j \mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}\|^2_\infty + \|\tilde{\mathbf{z}}\|^2_\infty) \\
&= -\sum_{s=0}^{t} \mathbf{L}_{s,t} (\mathbf{H}_s \tilde{\mathbf{x}}_s + \mathbf{M}_s \mathbf{v}_s + \sum_{j=1}^{n_z} (\tilde{\mathbf{x}}_s^T \mathbf{H}_s^{h^j} \tilde{\mathbf{x}}_s) \mathbf{e}_j^{n_z}) \\
&\quad + \sum_{k=1}^{n_u} \sum_{i=0}^{t} \sum_{j=0}^{t} (\tilde{\mathbf{z}}_i^T \mathbf{H}_t^{\pi^{kij}} \tilde{\mathbf{z}}_j) \mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}\|^2_\infty + \|\tilde{\mathbf{z}}\|^2_\infty) \\
&= -\sum_{s=0}^{t} \mathbf{L}_{s,t} \mathbf{H}_s \tilde{\mathbf{x}}_s - \sum_{s=0}^{t} \mathbf{L}_{s,t} \mathbf{M}_s \mathbf{v}_s - \sum_{s=0}^{t} \sum_{j=1}^{n_z} (\tilde{\mathbf{x}}_s^T \mathbf{H}_s^{h^j} \tilde{\mathbf{x}}_s) \mathbf{L}_{s,t} \mathbf{e}_j^{n_z}
\end{aligned}$$

$$+\sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j + 2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j$$

$$+ \mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}\|_\infty^2), \tag{6.62}$$

where we have used the fact that for $1 \le k \le n_u$ and $0 \le i, j \le t$, we can evaluate the following scalar value

$$\begin{aligned}\tilde{\mathbf{z}}_i^T\mathbf{H}_t^{\pi^{kij}}\tilde{\mathbf{z}}_j =&(\mathbf{H}_i\tilde{\mathbf{x}}_i + \mathbf{M}_i\mathbf{v}_i)^T\mathbf{H}_t^{\pi^{kij}}(\mathbf{H}_j\tilde{\mathbf{x}}_j + \mathbf{M}_j\mathbf{v}_j) + o(\|\tilde{\mathbf{x}}\|_\infty^2)\\ =&\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j + \tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j + \mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j\\ &+ \mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j + o(\|\tilde{\mathbf{x}}\|_\infty^2)\\ =&\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j + 2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j + \mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j + o(\|\tilde{\mathbf{x}}\|_\infty^2).\end{aligned}$$

Note that the error in the above expression is in fact $O(\|\tilde{\mathbf{x}}\|_\infty^4)$. Similar terms in the next equations also will be treated the same as long as there is an $o(\|\tilde{\mathbf{x}}\|_\infty^2)$ error in the overall expression.

Now, we can simplify the second-order expansion of the dynamics:

$$\tilde{\mathbf{x}}_{t+1} =\mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{G}_t\mathbf{w}_t + \begin{pmatrix} \begin{pmatrix}\tilde{\mathbf{x}}_t\\ \tilde{\mathbf{u}}_t\end{pmatrix}^T \begin{pmatrix}\mathbf{F}_{\mathbf{xx}}^{t,1} & \mathbf{F}_{\mathbf{xu}}^{t,1}\\ \mathbf{F}_{\mathbf{ux}}^{t,1} & \mathbf{F}_{\mathbf{uu}}^{t,1}\end{pmatrix} \begin{pmatrix}\tilde{\mathbf{x}}_t\\ \tilde{\mathbf{u}}_t\end{pmatrix}\\ \vdots\\ \begin{pmatrix}\tilde{\mathbf{x}}_t\\ \tilde{\mathbf{u}}_t\end{pmatrix}^T \begin{pmatrix}\mathbf{F}_{\mathbf{xx}}^{t,n_x} & \mathbf{F}_{\mathbf{xu}}^{t,n_x}\\ \mathbf{F}_{\mathbf{ux}}^{t,n_x} & \mathbf{F}_{\mathbf{uu}}^{t,n_x}\end{pmatrix} \begin{pmatrix}\tilde{\mathbf{x}}_t\\ \tilde{\mathbf{u}}_t\end{pmatrix}\end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2) \tag{6.63a}$$

$$=\mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t(-\sum_{s=0}^{t}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s + \sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{z}}_i^T\mathbf{H}_t^{\pi^{kij}}\tilde{\mathbf{z}}_j\mathbf{e}_k^{n_u}) + \mathbf{G}_t\mathbf{w}_t$$

$$
+\begin{pmatrix}
\begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\sum\limits_{s=0}^{t} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s \end{pmatrix}^T
\begin{pmatrix} \mathbf{F}^{t,1}_{\mathbf{xx}} & \mathbf{F}^{t,1}_{\mathbf{xu}} \\ \mathbf{F}^{t,1}_{\mathbf{ux}} & \mathbf{F}^{t,1}_{\mathbf{uu}} \end{pmatrix}
\begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\sum\limits_{s=0}^{t} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s \end{pmatrix} \\
\vdots \\
\begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\sum\limits_{s=0}^{t} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s \end{pmatrix}^T
\begin{pmatrix} \mathbf{F}^{t,n_x}_{\mathbf{xx}} & \mathbf{F}^{t,n_x}_{\mathbf{xu}} \\ \mathbf{F}^{t,n_x}_{\mathbf{ux}} & \mathbf{F}^{t,n_x}_{\mathbf{uu}} \end{pmatrix}
\begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\sum\limits_{s=0}^{t} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s \end{pmatrix}
\end{pmatrix}
+ o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{z}}\|_\infty^2)
\tag{6.63b}
$$

$$
\begin{aligned}
=&\mathbf{A}_t\tilde{\mathbf{x}}_t - \sum_{s=0}^{t}\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{H}_s\tilde{\mathbf{x}}_s - \sum_{s=0}^{t}\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s - \sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} \\
&+\sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j+2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j+\mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{B}_t\mathbf{e}_k^{n_u} \\
&+\sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{x}}_i^T\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j\mathbf{e}_k^{n_x} + \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{x}}_i^T\mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} \\
&+\sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\mathbf{v}_i^T\mathbf{H}_t^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} + \mathbf{G}_t\mathbf{w}_t + o(\|\tilde{\mathbf{x}}\|_\infty^2)
\end{aligned}
\tag{6.63c}
$$

$$
\begin{aligned}
=&\sum_{s=0}^{t}\mathbf{U}_{s,t}\tilde{\mathbf{x}}_s + \sum_{s=0}^{t}\mathbf{V}_{s,t}\mathbf{v}_s + \mathbf{G}_t\mathbf{w}_t - \sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} \\
&+\sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j+2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j+\mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{B}_t\mathbf{e}_k^{n_u} \\
&+\sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{x}}_i^T\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j\mathbf{e}_k^{n_x} + \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{x}}_i^T\mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} \\
&+\sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\mathbf{v}_i^T\mathbf{H}_t^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} + o(\|\tilde{\mathbf{x}}\|_\infty^2),
\end{aligned}
\tag{6.63d}
$$

where $\mathbf{U}_{s,t} := \mathbf{A}_s - \mathbf{B}_t\mathbf{L}_{s,t}\mathbf{H}_s, s = t$, $\mathbf{U}_{s,t} := -\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{H}_s, 0 \le s \le t-1$, and $\mathbf{V}_{s,t} := -\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{M}_s, 0 \le s \le t$ defined as before. Also, in (6.63c) we have used the fact that for $1 \le k \le n_x$, we can evaluate the following scalar value, and define the related matrices, such that

$$\begin{aligned}
&\begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\sum\limits_{s=0}^{t} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,k}_{\mathbf{xx}} & \mathbf{F}^{t,k}_{\mathbf{xu}} \\ \mathbf{F}^{t,k}_{\mathbf{ux}} & \mathbf{F}^{t,k}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\sum\limits_{s=0}^{t} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s \end{pmatrix} \\
&= \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} (\sum_{s=0}^{t} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s) - (\sum_{s=0}^{t} \tilde{\mathbf{z}}_s^T \mathbf{L}_{s,t}^T) \mathbf{F}^{t,k}_{\mathbf{ux}} \tilde{\mathbf{x}}_t + (\sum_{s=0}^{t} \tilde{\mathbf{z}}_s^T \mathbf{L}_{s,t}^T) \mathbf{F}^{t,k}_{\mathbf{uu}} (\sum_{s=0}^{t} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s) \\
&= \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \tilde{\mathbf{x}}_t - \sum_{s=0}^{t} \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s - \sum_{s=0}^{t} \tilde{\mathbf{z}}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{ux}} \tilde{\mathbf{x}}_t + (\sum_{s=0}^{t} \tilde{\mathbf{z}}_s^T \mathbf{L}_{s,t}^T) \mathbf{F}^{t,k}_{\mathbf{uu}} (\sum_{i=0}^{t} \mathbf{L}_{i,t}\tilde{\mathbf{z}}_i) \\
&= \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \tilde{\mathbf{x}}_t - \sum_{s=0}^{t} \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \mathbf{L}_{s,t}\tilde{\mathbf{z}}_s - \sum_{s=0}^{t} \tilde{\mathbf{z}}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{ux}} \tilde{\mathbf{x}}_t + \sum_{s=0}^{t} \sum_{i=0}^{t} \tilde{\mathbf{z}}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}\tilde{\mathbf{z}}_i \\
&= \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \tilde{\mathbf{x}}_t - \sum_{s=0}^{t} \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \mathbf{L}_{s,t}(\mathbf{H}_s\tilde{\mathbf{x}}_s + \mathbf{M}_s\mathbf{v}_s) - \sum_{s=0}^{t} (\mathbf{H}_s\tilde{\mathbf{x}}_s + \mathbf{M}_s\mathbf{v}_s)^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{ux}} \tilde{\mathbf{x}}_t \\
&+ \sum_{s=0}^{t} \sum_{i=0}^{t} (\mathbf{H}_s\tilde{\mathbf{x}}_s + \mathbf{M}_s\mathbf{v}_s)^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}(\mathbf{H}_i\tilde{\mathbf{x}}_i + \mathbf{M}_i\mathbf{v}_i) + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
&= \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \tilde{\mathbf{x}}_t - \sum_{s=0}^{t} \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \mathbf{L}_{s,t}\mathbf{H}_s\tilde{\mathbf{x}}_s - \sum_{s=0}^{t} \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s \\
&- \sum_{s=0}^{t} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{ux}} \tilde{\mathbf{x}}_t - \sum_{s=0}^{t} \mathbf{v}_s^T \mathbf{M}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{ux}} \tilde{\mathbf{x}}_t \\
&+ \sum_{s=0}^{t} \sum_{i=0}^{t} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}\mathbf{H}_i\tilde{\mathbf{x}}_i + \sum_{s=0}^{t} \sum_{i=0}^{t} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}\mathbf{M}_i\mathbf{v}_i \\
&+ \sum_{s=0}^{t} \sum_{i=0}^{t} \mathbf{v}_s^T \mathbf{M}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}\mathbf{H}_i\tilde{\mathbf{x}}_i + \sum_{s=0}^{t} \sum_{i=0}^{t} \mathbf{v}_s^T \mathbf{M}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}\mathbf{M}_i\mathbf{v}_i + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
&= \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \tilde{\mathbf{x}}_t - 2\sum_{s=0}^{t} \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \mathbf{L}_{s,t}\mathbf{H}_s\tilde{\mathbf{x}}_s - 2\sum_{s=0}^{t} \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s \\
&+ \sum_{s=0}^{t} \sum_{i=0}^{t} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}\mathbf{H}_i\tilde{\mathbf{x}}_i + \sum_{s=0}^{t} \sum_{i=0}^{t} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}\mathbf{M}_i\mathbf{v}_i \\
&+ \sum_{s=0}^{t} \sum_{i=0}^{t} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T (\mathbf{F}^{t,k}_{\mathbf{uu}})^T \mathbf{L}_{i,t}\mathbf{M}_i\mathbf{v}_i + \sum_{s=0}^{t} \sum_{i=0}^{t} \mathbf{v}_s^T \mathbf{M}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}\mathbf{M}_i\mathbf{v}_i + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
&= \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \tilde{\mathbf{x}}_t - 2\tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \mathbf{L}_{t,t}\mathbf{H}_t\tilde{\mathbf{x}}_t - 2\sum_{s=0}^{t-1} \tilde{\mathbf{x}}_t^T \mathbf{F}^{t,k}_{\mathbf{xx}} \mathbf{L}_{s,t}\mathbf{H}_s\tilde{\mathbf{x}}_s \\
&+ \sum_{s=0}^{t-1} \sum_{i=0}^{t-1} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}\mathbf{H}_i\tilde{\mathbf{x}}_i + \sum_{i=0}^{t-1} \tilde{\mathbf{x}}_t^T \mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}^{t,k}_{\mathbf{uu}} \mathbf{L}_{i,t}\mathbf{H}_i\tilde{\mathbf{x}}_i
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{s=0}^{t-1} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{t,t} \mathbf{H}_t \tilde{\mathbf{x}}_t + \tilde{\mathbf{x}}_t^T \mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{t,t} \mathbf{H}_t \tilde{\mathbf{x}}_t \\
&+ 2 \sum_{s=0}^{t} \sum_{i=0}^{t} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{i,t} \mathbf{M}_i \mathbf{v}_i - 2 \sum_{s=0}^{t} \tilde{\mathbf{x}}_t^T \mathbf{F}_{\mathbf{xx}}^{t,k} \mathbf{L}_{s,t} \mathbf{M}_s \mathbf{v}_s \\
&+ \sum_{s=0}^{t} \sum_{i=0}^{t} \mathbf{v}_s^T \mathbf{M}_s^T \mathbf{L}_{s,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{i,t} \mathbf{M}_i \mathbf{v}_i + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
&= \tilde{\mathbf{x}}_t^T (\mathbf{F}_{\mathbf{xx}}^{t,k} - 2\mathbf{F}_{\mathbf{xx}}^{t,k} \mathbf{L}_{t,t} \mathbf{H}_t + \mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{t,t} \mathbf{H}_t) \tilde{\mathbf{x}}_t \\
&+ \sum_{s=0}^{t-1} \tilde{\mathbf{x}}_t^T (-2\mathbf{F}_{\mathbf{xx}}^{t,k} \mathbf{L}_{s,t} \mathbf{H}_s + 2\mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{s,t} \mathbf{H}_s) \tilde{\mathbf{x}}_s + \sum_{s=0}^{t-1} \sum_{i=0}^{t-1} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{i,t} \mathbf{H}_i \tilde{\mathbf{x}}_i \\
&+ 2 \sum_{s=0}^{t-1} \sum_{i=0}^{t} \tilde{\mathbf{x}}_s^T \mathbf{H}_s^T \mathbf{L}_{s,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{i,t} \mathbf{M}_i \mathbf{v}_i + 2 \sum_{s=0}^{t} \tilde{\mathbf{x}}_t^T \mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{s,t} \mathbf{M}_s \mathbf{v}_s \\
&- 2 \sum_{s=0}^{t} \tilde{\mathbf{x}}_t^T \mathbf{F}_{\mathbf{xx}}^{t,k} \mathbf{L}_{s,t} \mathbf{M}_s \mathbf{v}_s + \sum_{s=0}^{t} \sum_{i=0}^{t} \mathbf{v}_s^T \mathbf{M}_s^T \mathbf{L}_{s,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{i,t} \mathbf{M}_i \mathbf{v}_i + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
&= \tilde{\mathbf{x}}_t^T (\mathbf{F}_{\mathbf{xx}}^{t,k} - 2\mathbf{F}_{\mathbf{xx}}^{t,k} \mathbf{L}_{t,t} \mathbf{H}_t + \mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{t,t} \mathbf{H}_t) \tilde{\mathbf{x}}_t \\
&+ \sum_{j=0}^{t-1} \tilde{\mathbf{x}}_t^T (-2\mathbf{F}_{\mathbf{xx}}^{t,k} \mathbf{L}_{j,t} \mathbf{H}_j + 2\mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{j,t} \mathbf{H}_j) \tilde{\mathbf{x}}_j + \sum_{i=0}^{t-1} \sum_{j=0}^{t-1} \tilde{\mathbf{x}}_i^T \mathbf{H}_i^T \mathbf{L}_{i,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{j,t} \mathbf{H}_j \tilde{\mathbf{x}}_j \\
&+ 2 \sum_{i=0}^{t-1} \sum_{j=0}^{t} \tilde{\mathbf{x}}_i^T \mathbf{H}_i^T \mathbf{L}_{i,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{j,t} \mathbf{M}_j \mathbf{v}_j + \sum_{j=0}^{t} \tilde{\mathbf{x}}_t^T (2\mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{j,t} \mathbf{M}_j - 2\mathbf{F}_{\mathbf{xx}}^{t,k} \mathbf{L}_{j,t} \mathbf{M}_j) \mathbf{v}_j \\
&+ \sum_{i=0}^{t} \sum_{j=0}^{t} \mathbf{v}_i^T \mathbf{M}_i^T \mathbf{L}_{i,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{j,t} \mathbf{M}_j \mathbf{v}_j + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
&=: \sum_{i=0}^{t} \sum_{j=0}^{t} \tilde{\mathbf{x}}_i^T \mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}} \tilde{\mathbf{x}}_j + \sum_{i=0}^{t} \sum_{j=0}^{t} \tilde{\mathbf{x}}_i^T \mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}} \mathbf{v}_j + \sum_{i=0}^{t} \sum_{j=0}^{t} \mathbf{v}_i^T \mathbf{H}_t^{(\mathbf{v},f,\mathbf{v})^{kij}} \mathbf{v}_j + o(\|\tilde{\mathbf{x}}\|_\infty^2),
\end{aligned}
$$

where

- $\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}} := \mathbf{H}_i^T \mathbf{L}_{i,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{j,t} \mathbf{H}_j, 0 \le i \le t-1, 0 \le j \le t-1;$
- $\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}} := (-2\mathbf{F}_{\mathbf{xx}}^{t,k} \mathbf{L}_{j,t} \mathbf{H}_j + 2\mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{j,t} \mathbf{H}_j), i = t, 0 \le j \le t-1;$
- $\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}} := (\mathbf{F}_{\mathbf{xx}}^{t,k} - 2\mathbf{F}_{\mathbf{xx}}^{t,k} \mathbf{L}_{t,t} \mathbf{H}_t + \mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{t,t} \mathbf{H}_t), i = j = t;$
- $\mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}} := 2\mathbf{H}_i^T \mathbf{L}_{i,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{j,t} \mathbf{M}_j, 0 \le i \le t-1, 0 \le j \le t;$
- $\mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}} := (2\mathbf{H}_t^T \mathbf{L}_{t,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{j,t} \mathbf{M}_j - 2\mathbf{F}_{\mathbf{xx}}^{t,k} \mathbf{L}_{j,t} \mathbf{M}_j), i = t, 0 \le j \le t;$ and
- $\mathbf{H}_t^{(\mathbf{v},f,\mathbf{v})^{kij}} := \mathbf{M}_i^T \mathbf{L}_{i,t}^T \mathbf{F}_{\mathbf{uu}}^{t,k} \mathbf{L}_{j,t} \mathbf{M}_j, 0 \le i \le t, 0 \le j \le t.$

Now, in (6.63d), the linear recursion in $\tilde{\mathbf{x}}$ can be solved by defining the $\mathbf{Q}$ polynomial the same as in (6.53) and using (6.54). In particular, there exists matrices $\mathbf{U}_t^{\mathbf{x}_0}, 0 \le t \le K-1$, $\mathbf{V}_{s,t}^{\mathbf{v}}, 0 \le s \le t, 0 \le t \le K-1$, and $\mathbf{W}_{s,t}^{\mathbf{w}}, 0 \le s \le t, 0 \le t \le K-1$ such that

$$
\begin{aligned}
\tilde{\mathbf{x}}_{t+1} =& \sum_{s=0}^{t} \mathbf{U}_{s,t}\tilde{\mathbf{x}}_s + \sum_{s=0}^{t} \mathbf{V}_{s,t}\mathbf{v}_s + \mathbf{G}_t\mathbf{w}_t - \sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{hj}\tilde{\mathbf{x}}_s)\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} \\
&+\sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j+2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j+\mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{B}_t\mathbf{e}_k^{n_u} \\
&+ \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{x}}_i^T\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j\mathbf{e}_k^{n_x} + \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{x}}_i^T\mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} \\
&+ \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\mathbf{v}_i^T\mathbf{H}_t^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} + o(\|\tilde{\mathbf{x}}\|_\infty^2)
\end{aligned} \tag{6.64a}
$$

$$
\begin{aligned}
=&(\sum_{s=0}^{t}\mathbf{Q}_s\mathbf{U}_{0,t-s})\mathbf{x}_0 + \sum_{s=0}^{t}\sum_{r=0}^{t-s}\mathbf{Q}_s\mathbf{V}_{r,t-s}\mathbf{v}_r + \sum_{s=0}^{t}\mathbf{Q}_s\mathbf{G}_{t-s}\mathbf{w}_{t-s} \\
&+ \sum_{s=0}^{t}\sum_{k=1}^{n_u}\sum_{i=0}^{t-s}\sum_{j=0}^{t-s}\Big(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s}^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j + 2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j \\
&+ \mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_{t-s}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j\Big)\mathbf{Q}_s\mathbf{B}_{t-s}\mathbf{e}_k^{n_u} \\
&+ \sum_{s=0}^{t}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s}\sum_{j=0}^{t-s}\Big(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j + \tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j + \mathbf{v}_i^T\mathbf{H}_{t-s}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j\Big)\mathbf{Q}_s\mathbf{e}_k^{n_x} \\
&- \sum_{s=0}^{t}\sum_{r=0}^{t-s}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{hj}\tilde{\mathbf{x}}_r)\mathbf{Q}_s\mathbf{B}_{t-s}\mathbf{L}_{r,t-s}\mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}\|_\infty^2)
\end{aligned} \tag{6.64b}
$$

$$
\begin{aligned}
=&\mathbf{U}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t}(\mathbf{V}_{s,t}^{\mathbf{v}}\mathbf{v}_s + \mathbf{W}_{s,t}^{\mathbf{w}}\mathbf{w}_s) \\
&+ \sum_{s=0}^{t}\sum_{k=1}^{n_u}\sum_{i=0}^{t-s}\sum_{j=0}^{t-s}\Big(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s}^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j + 2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j \\
&+ \mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_{t-s}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j\Big)\mathbf{Q}_s\mathbf{B}_{t-s}\mathbf{e}_k^{n_u} \\
&+ \sum_{s=0}^{t}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s}\sum_{j=0}^{t-s}\Big(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j + \tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j + \mathbf{v}_i^T\mathbf{H}_{t-s}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j\Big)\mathbf{Q}_s\mathbf{e}_k^{n_x} \\
&- \sum_{s=0}^{t}\sum_{r=0}^{t-s}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{hj}\tilde{\mathbf{x}}_r)\mathbf{Q}_s\mathbf{B}_{t-s}\mathbf{L}_{r,t-s}\mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}\|_\infty^2),
\end{aligned} \tag{6.64c}
$$

where $\mathbf{U}_t^{\mathbf{x}_0} := (\sum_{s=0}^t \mathbf{Q}_s \mathbf{U}_{0,t-s})$, $\mathbf{V}_{s,t}^{\mathbf{v}} := \sum\limits_{r=0}^{t-s} \mathbf{Q}_r \mathbf{V}_{s,t-r}, 0 \leq s \leq t$, and $\mathbf{W}_{s,t}^{\mathbf{w}} := \mathbf{Q}_{s-t}\mathbf{G}_s, 0 \leq s \leq t$. Note in the last equation, we used the following summation exchange formula (which can be easily proven by writing expanding and collecting the terms)

$$\sum_{s=0}^{t}\sum_{r=0}^{t-s} f_{s,r}x_r = \sum_{r=0}^{t}\sum_{s=0}^{t-r} f_{s,r}x_r = \sum_{r=0}^{t}(\sum_{s=0}^{t-r} f_{s,r})x_r,$$

for some $x_r$ and $f_{s,r}$. Therefore, we wrote the following

$$\sum_{s=0}^{t}\sum_{r=0}^{t-s}\mathbf{Q}_s\mathbf{V}_{r,t-s}\mathbf{v}_r = \sum_{r=0}^{t}(\sum_{s=0}^{t-r}\mathbf{Q}_s\mathbf{V}_{r,t-s})\mathbf{v}_r = \sum_{s=0}^{t}(\sum_{r=0}^{t-s}\mathbf{Q}_r\mathbf{V}_{s,t-r})\mathbf{v}_s = \sum_{s=0}^{t}\mathbf{V}_{s,t}^{\mathbf{v}}\mathbf{v}_s.$$

Finally, note that we simplified the following expression (by redefining $y = t - s$ and the relabeling):

$$\sum_{s=0}^{t}\mathbf{Q}_s\mathbf{G}_{t-s}\mathbf{w}_{t-s} = \sum_{y=t}^{0}\mathbf{Q}_{y-t}\mathbf{G}_y\mathbf{w}_y = \sum_{s=0}^{t}\mathbf{Q}_{s-t}\mathbf{G}_s\mathbf{w}_s.$$

*Validity region:* Similar to the fully-observed situation, the definition of $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}_t^p$. Therefore, the properties of $O(\|\tilde{\mathbf{x}}_t\|_\infty)$ that we have proven in Section 6.1 for a deterministic feedback design still hold for the above Taylor expansion, as well. Particularly, we proved that for $\boldsymbol{\pi}^d$ design, $O(\|\tilde{\mathbf{x}}_t\|_\infty) = O(\delta)$ in a set $\Omega(\delta)$ properly defined as before with probability $1 - o(\epsilon)$. Hence, for $\omega \in \Omega(\delta)$, $O(\|\tilde{\mathbf{x}}_t\|_\infty^2) = O(\delta^2)$. Thus, for $\omega \in \Omega(\delta)$ (the same set and with the same probability), we have:

$$\begin{aligned}\tilde{\mathbf{x}}_{t+1} =& \mathbf{U}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t}(\mathbf{V}_{s,t}^{\mathbf{v}}\mathbf{v}_s + \mathbf{W}_{s,t}^{\mathbf{w}}\mathbf{w}_s) + \sum_{s=0}^{t}\sum_{k=1}^{n_u}\sum_{i=0}^{t-s}\sum_{j=0}^{t-s}\Big(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s}^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j \\ &+ 2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j + \mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_{t-s}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j\Big)\mathbf{Q}_s\mathbf{B}_{t-s}\mathbf{e}_k^{n_u}\end{aligned}$$

$$+\sum_{s=0}^{t}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s}\sum_{j=0}^{t-s}\Big(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}_{t-s}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j\Big)\mathbf{Q}_s\mathbf{e}_k^{n_x}$$
$$-\sum_{s=0}^{t}\sum_{r=0}^{t-s}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{h^j}\tilde{\mathbf{x}}_r)\mathbf{Q}_s\mathbf{B}_{t-s}\mathbf{L}_{r,t-s}\mathbf{e}_j^{n_z}+O(\delta^2). \tag{6.65}$$

*Second-order expansion of the cost function:* Similarly, we obtain the second-order Taylor series expansion of the cost function around the nominal trajectory:

$$J = J^p + \tilde{J}_1 + \tilde{J}_2 + o(\sum_{t=1}^{K-1}(\|\tilde{\mathbf{x}}_t\|^2 + \|\tilde{\mathbf{u}}_t\|^2) + \|\tilde{\mathbf{x}}_K\|^2) \tag{6.66a}$$
$$= J^p + \tilde{J}_1 + \tilde{J}_2 + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2), \tag{6.66b}$$

as $(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2) \downarrow 0$. Moreover, we have:

- $J^p{:=}\sum_{t=0}^{K-1} c_t(\mathbf{x}_t^p, \mathbf{u}_t^p) + c_K(\mathbf{x}_K^p)$ denotes the nominal cost;
- $\tilde{J}_1{:=}\sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t+\mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t)+\mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K$ is the first order cost error;
- $\tilde{J}_2 := \sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xx}}\tilde{\mathbf{x}}_t+\frac{1}{2}\tilde{\mathbf{u}}_t^T\mathbf{C}_t^{\mathbf{uu}}\tilde{\mathbf{u}}_t+\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xu}}\tilde{\mathbf{u}}_t)+\frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}_K^{\mathbf{xx}}\tilde{\mathbf{x}}_K$ is the second order cost error.
- $J_2 := J^p + \tilde{J}_1 + \tilde{J}_2$ is the second order approximation of the cost function;
- $\mathbf{C}_t^{\mathbf{xx}} = \nabla_{\mathbf{xx}}^2 c_t(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^p,\mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{uu}} = \nabla_{\mathbf{uu}}^2 c_t(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^p,\mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{xu}} = \nabla_{\mathbf{xu}}^2 c_t(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^p,\mathbf{u}_t^p}$, and $\mathbf{C}_K^{\mathbf{xx}} = \nabla_{\mathbf{xx}}^2 c_K(\mathbf{x})|_{\mathbf{x}_K^p}$, where we have used the fact that $c_t \in \mathbb{C}^2$.

Next, we provide the main result regarding the expected second order error of the cost function.

**Theorem 9 Second-Order Cost Function Error for a Partially-Observed System with a Deterministic Policy:** *Given that process noises are zero mean i.i.d. Gaussian, the initial error is zero mean Gaussian, and all the functions are in $\mathbb{C}^2$, under a first-order approximation for the small noise paradigm, the stochastic cost function is dominated by the nominal part of the cost function, and the expected* *first-order error is $O(\delta^2)$. That is,*

$$\mathbb{E}[\tilde{J}_1] = O(\delta^2), \text{ and } \mathbb{E}[J] = J^p + O(\delta^2).$$

*Moreover, by choosing $\delta = \sqrt{2\log(\frac{1}{\epsilon})}\epsilon$, we have*

$$\mathbb{E}[\tilde{J}_1] = O(\epsilon^{2-\gamma}), \text{ and } \mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma}),$$

*for some $0 < \gamma \ll 1$, which shows that this error tends to zero with a near-first-order rate as $\epsilon \downarrow 0$.*

***Proof 22*** *First, let us simply the first order cost error:*

$$\begin{aligned}
\tilde{J}_1 = &\sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t + \mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t) + \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K \\
=&\sum_{t=0}^{K-1}\Bigg(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t - \sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{H}_s\tilde{\mathbf{x}}_s - \sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s - \sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} \\
&+\sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j + 2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j + \mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{C}_t^{\mathbf{u}}\mathbf{e}_k^{n_u}\Bigg) \\
&+ \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
=&\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{x}}_t - \sum_{t=0}^{K-1}\sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s - \sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} \\
&+ \sum_{t=0}^{K-1}\sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j + 2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j \\
&+ \mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{C}_t^{\mathbf{u}}\mathbf{e}_k^{n_u} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
=&\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\Bigg(\mathbf{U}_{t-1}^{\mathbf{x}_0}\tilde{\mathbf{x}}_0 + \sum_{s=0}^{t-1}(\mathbf{V}_{s,t-1}^{\mathbf{v}}\mathbf{v}_s + \mathbf{W}_{s,t-1}^{\mathbf{w}}\mathbf{w}_s) + \sum_{s=0}^{t-1}\sum_{k=1}^{n_u}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j \\
&+ 2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j + \mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{e}_k^{n_u} \\
&+\sum_{s=0}^{t-1}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j + \tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j + \mathbf{v}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j)\mathbf{Q}_s\mathbf{e}_k^{n_x}
\end{aligned}$$

$$
\begin{aligned}
&-\sum_{s=0}^{t-1}\sum_{r=0}^{t-s-1}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{hj}\tilde{\mathbf{x}}_r)\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{L}_{r,t-s-1}\mathbf{e}_j^{n_z}\Bigg)\\
&-\sum_{t=0}^{K-1}\sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s-\sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{hj}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z}\\
&+\sum_{t=0}^{K-1}\sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j+2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j\\
&+\mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{C}_t^{\mathbf{u}}\mathbf{e}_k^{n_u}+o(\|\tilde{\mathbf{x}}\|_\infty^2)\\
=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}})\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t\\
&+\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_u}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j\\
&+2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j+\mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{e}_k^{n_u}\\
&+\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{e}_k^{n_x}\\
&-\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{r=0}^{t-s-1}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{hj}\tilde{\mathbf{x}}_r)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{L}_{r,t-s-1}\mathbf{e}_j^{n_z}\\
&-\sum_{t=0}^{K-1}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t-\sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{hj}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z}\\
&+\sum_{t=0}^{K-1}\sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j+2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j\\
&+\mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{C}_t^{\mathbf{u}}\mathbf{e}_k^{n_u}+o(\|\tilde{\mathbf{x}}\|_\infty^2)\\
=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t\\
&+\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{e}_k^{n_x}\\
&+\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_u}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j\\
&+2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j+\mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{e}_k^{n_u}\\
&+\sum_{t=0}^{K-1}\sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j+2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j+\mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{C}_t^{\mathbf{u}}\mathbf{e}_k^{n_u}
\end{aligned}
$$

$$
\begin{aligned}
&-\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{r=0}^{t-s-1}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{h^j}\tilde{\mathbf{x}}_r)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{L}_{r,t-s-1}\mathbf{e}_j^{n_z}\\
&-\sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z}+o(\|\tilde{\mathbf{x}}\|_\infty^2),
\end{aligned}
$$

*where* $\mathbf{C}_t^{\mathbf{L}} := \mathbf{C}_t^{\mathbf{x}} - \sum\limits_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{H}_t, 0 \le t \le K-1,$ *and* $\mathbf{C}_K^{\mathbf{L}} = \mathbf{C}_K^{\mathbf{x}}$*. Note that in evaluating the above expression, we used the following summation exchange formula*

$$
\sum_{t=0}^{K-1}\sum_{s=0}^{t}f_{t,s}x_s=\sum_{s=0}^{K-1}\sum_{t=s}^{K-1}f_{t,s}x_s=\sum_{s=0}^{K-1}(\sum_{t=s}^{K-1}f_{t,s})x_s=\sum_{t=0}^{K-1}(\sum_{s=t}^{K-1}f_{s,t})x_t.
$$

*For instance, we simplified the following expression:*

$$
\sum_{t=0}^{K-1}\sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{H}_s\tilde{\mathbf{x}}_s=\sum_{t=0}^{K-1}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{H}_t)\tilde{\mathbf{x}}_t.
$$

*Next, we simplify the second order cost error. Once again, we ignore the second order feedback terms and replace them with* $o(\|\tilde{\mathbf{x}}\|_\infty^2)$

$$
\begin{aligned}
\tilde{J}_2&=\sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xx}}\tilde{\mathbf{x}}_t+\frac{1}{2}\tilde{\mathbf{u}}_t^T\mathbf{C}_t^{\mathbf{uu}}\tilde{\mathbf{u}}_t+\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xu}}\tilde{\mathbf{u}}_t)+\frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}_K^{\mathbf{xx}}\tilde{\mathbf{x}}_K\\
&=\sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xx}}\tilde{\mathbf{x}}_t+\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{L}_t^T\mathbf{C}_t^{\mathbf{uu}}\mathbf{L}_t\tilde{\mathbf{x}}_t+\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xu}}\mathbf{L}_t\tilde{\mathbf{x}}_t)+\frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}_K^{\mathbf{xx}}\tilde{\mathbf{x}}_K+o(\|\tilde{\mathbf{x}}\|_\infty^2)\\
&=\sum_{t=0}^{K}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{LL}}\tilde{\mathbf{x}}_t+o(\|\tilde{\mathbf{x}}\|_\infty^2),
\end{aligned}
$$

*where* $\mathbf{C}_t^{\mathbf{LL}} := \frac{1}{2}\mathbf{C}_t^{\mathbf{xx}} + \frac{1}{2}\mathbf{L}_t^T\mathbf{C}_t^{\mathbf{uu}}\mathbf{L}_t + \mathbf{C}_t^{\mathbf{xu}}\mathbf{L}_t, 0 \le t \le K-1,$ *and* $\mathbf{C}_K^{\mathbf{LL}} := \frac{1}{2}\mathbf{C}_K^{\mathbf{xx}}$*. Hence, we have:*

$$
\begin{aligned}
\tilde{J}_1+\tilde{J}_2=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t\\
&+\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{e}_k^{n_x}
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_u}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j\\
&+2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j+\mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_{t-s-1}^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{e}_k^{n_u}\\
&+\sum_{t=0}^{K-1}\sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{H}_j\tilde{\mathbf{x}}_j+2\tilde{\mathbf{x}}_i^T\mathbf{H}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j+\mathbf{v}_i^T\mathbf{M}_i^T\mathbf{H}_t^{\pi^{kij}}\mathbf{M}_j\mathbf{v}_j)\mathbf{C}_t^{\mathbf{u}}\mathbf{e}_k^{n_u}\\
&-\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{r=0}^{t-s-1}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{h^j}\tilde{\mathbf{x}}_r)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{L}_{r,t-s-1}\mathbf{e}_j^{n_z}\\
&-\sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z}+\sum_{t=0}^{K}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{LL}}\tilde{\mathbf{x}}_t+o(\|\tilde{\mathbf{x}}\|_\infty^2)\\
=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t\\
&+O(\|\tilde{\mathbf{x}}\|_\infty^2)+o(\|\tilde{\mathbf{x}}\|_\infty^2)\\
=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t\\
&+O(\|\tilde{\mathbf{x}}\|_\infty^2)
\end{aligned}
$$

*Hence, for* $\omega\in\Omega(\delta)$*,*

$$
\begin{aligned}
\tilde{J}_1+\tilde{J}_2=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t\\
&+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t+O(\delta^2).
\end{aligned}
$$

*As a result, using* (6.75b)*, for* $\omega\in\Omega(\delta)$*, we have:*

$$
\begin{aligned}
J=&J^p+(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t\\
&+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t+O(\delta^2).
\end{aligned}
$$

*Next, note* $\mathbb{E}[\tilde{\mathbf{x}}_0] = \mathbb{E}[\mathbf{x}_0 - \hat{\mathbf{x}}_0] = \mathbf{0}$, *and* $\mathbb{E}[\mathbf{w}_t] = \mathbb{E}[\mathbf{v}_t] = 0$ *for all* $t$. *Therefore,*

$$\begin{aligned}
&\mathbb{E}[(\sum_{t=0}^{K} \mathbf{C}^{\mathbf{L}}_t \mathbf{U}^{\mathbf{x}_0}_{t-1})\tilde{\mathbf{x}}_0 + \sum_{t=0}^{K}(\sum_{s=t}^{K-1} \mathbf{C}^{\mathbf{L}}_s \mathbf{V}^{\mathbf{v}}_{t,s-1} - \mathbf{C}^{\mathbf{u}}_s \mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t + \sum_{t=0}^{K}(\sum_{s=t}^{K-1} \mathbf{C}^{\mathbf{L}}_s \mathbf{W}^{\mathbf{w}}_{t,s-1})\mathbf{w}_t] \\
&= (\sum_{t=0}^{K} \mathbf{C}^{\mathbf{L}}_t \mathbf{U}^{\mathbf{x}_0}_{t-1})\mathbb{E}[\tilde{\mathbf{x}}_0] + \sum_{t=0}^{K}(\sum_{s=t}^{K-1} \mathbf{C}^{\mathbf{L}}_s \mathbf{V}^{\mathbf{v}}_{t,s-1} - \mathbf{C}^{\mathbf{u}}_s \mathbf{L}_{t,s}\mathbf{M}_t)\mathbb{E}[\mathbf{v}_t] \\
&\quad + \sum_{t=0}^{K}(\sum_{s=t}^{K-1} \mathbf{C}^{\mathbf{L}}_s \mathbf{W}^{\mathbf{w}}_{t,s-1})\mathbb{E}[\mathbf{w}_t] = 0.
\end{aligned}$$

*Noting that for* $\omega \notin \Omega(\delta)$, $J \leq M$, *the expected value of* $J$ *is Now, since for* $\omega \notin \Omega(\delta)$, $J \leq M$, *then*

$$\begin{aligned}
\mathbb{E}[J - J^p] &= P(\Omega(\delta))(0 + O(\delta^2)) + M(1 - P(\Omega(\delta))) \\
&= P(\Omega(\delta))O(\delta) + M(1 - P(\Omega(\delta))) \qquad (6.67)
\end{aligned}$$

*Note the last expression is the same as* (4.47). *Although* $\Omega(\delta)$ *is not the same as in Theorem 4,* $P(\Omega(\delta))$ *is still the same. In the proof of Theorem 4 while we discussed on the probabilistic argument and choosing the proper* $\delta$, *we showed that by choosing* $\delta := \sqrt{-2\log(\epsilon)}\epsilon$, *the* $\mathbb{E}[J - J^p] = O(\epsilon^{2-\gamma})$. *Thus,* $\mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma})$. *Similarly,* $\mathbb{E}[\tilde{J}_1 + \tilde{J}_2] = O(\epsilon^{2-\gamma})$. *The same argument follows through and this theorem is proved.*

Hence, when the functions are in $\mathbb{C}^2$, the expected stochastic cost is equal to the nominal cost with a higher probability as $\epsilon \downarrow 0$. Therefore, it follows that the deterministic policy is near-second-order optimal, summarized below:

**Corollary 7 Near-Second-Order Optimality of the Deterministic Optimal Policy for the Stochastic Partially-Observed System Under Small Noise.** *Based on Theorem 9, for a partially-observed system where the function are in* $\mathbb{C}^2$ *under the small noise paradigm, as* $\epsilon \downarrow 0$, *the deterministic optimal control law becomes* $O(\epsilon^{2-\gamma})$*-optimal with some* $0 < \gamma \ll 1$ *for the stochastic problem.*

***Proof 23*** *Using Theorem 9, for $\omega \in \Omega(\delta)$ we have $\mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma})$, which is the cost of applying policy $\boldsymbol{\pi}^d$ to the stochastic system. Now, suppose $\boldsymbol{\pi}^*$ is the optimal stochastic policy. By assumption $\boldsymbol{\pi}^*$ is in $\mathbb{C}^2$. Therefore, by modifying the definition of $\mathbf{L}_{s,t}$ as $\mathbf{L}_{s,t} := -\nabla_{\mathbf{z}_s}\boldsymbol{\pi}^*_t(\mathbf{z}_{0:t})|_{\mathbf{z}^{*p}_{0:t}}$ and modifying $\mathbf{H}^{\pi^k}_t$ as $\mathbf{H}^{\pi^{kij}}_t := \frac{1}{2}\nabla^2_{\mathbf{z}_i\mathbf{z}_j}\pi^{*k}_t(\mathbf{z}_{0:t})|_{\mathbf{z}^{*p}_{0:t}}$, defining $\mathbf{u}^{*p}_t = \boldsymbol{\pi}^*_t(\mathbf{z}^{*p}_{0:t})$ and replacing $p$ with $*p$ in (4.23), we have $\boldsymbol{\pi}^*_t(\mathbf{z}_{0:t}) = \mathbf{u}^{*p}_t - \sum\limits_{s=0}^{t}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s + \sum_{k=1}^{n_u}\sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{z}}^T_i\mathbf{H}^{\pi^{kij}}_t\tilde{\mathbf{z}}_j\mathbf{e}^{n_u}_k + o(\|\tilde{\mathbf{x}}\|^2_\infty + \|\tilde{\mathbf{z}}\|^2_\infty)$ (where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{z}}$ are also modified to denote $(\mathbf{x}_t - \mathbf{x}^{*p}_t)$ and $(\mathbf{z}_t - \mathbf{z}^{*p}_t)$, respectively). Similarly, by using appropriate modifications the entire calculations of this section hold for this policy. Hence, using Theorem 9 for this system, the cost function of policy $\boldsymbol{\pi}^*$ can be written as $\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{2-\gamma})$. Now, by construction $J^p \leq J^{*p}$, and*

$$\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{2-\gamma}) \geq J^p + O(\epsilon^{2-\gamma}) = \mathbb{E}[J_{\boldsymbol{\pi}^d}] + O(\epsilon^{2-\gamma}).$$

*As a result, policy $\boldsymbol{\pi}^d$ is within $O(\epsilon^{2-\gamma})$ of the optimal stochastic policy.*

*Similarly, using the results of Theorem 9, we can write*

$$\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{2-\gamma}) \geq J^p + O(\epsilon^{2-\gamma}) = \mathbb{E}[J_{\boldsymbol{\pi}^d}] + O(\epsilon^{2-\gamma}).$$

*As a result, policy $\boldsymbol{\pi}^d$ is within $O(\epsilon^{2-\gamma})$ of the optimal stochastic policy.*

## 6.4 Near-Second-Order Optimality of T-LQG

In this section, we provide a second-order analysis of the deterministic feedback law and show that applying the optimal feedback law of the deterministic problem to the stochastic problem results in a near-second-order optimality as well. Therefore, we improve the results of Section 6.2.

*Assumptions:* Similar to the previous section,other than the assumptions of Section 6.2, we assume for the analysis of this section that all the functions (including the dynamics and observation models, feedback law, and the cost functions) are in $\mathbb{C}^2$.

*Second-order expansion of the system equations:* We obtain the second-order expansion of the process model around the nominal trajectory, for $0 \leq t \leq K-1$:

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) - \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p) + \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}} \mathbf{w}_t \tag{6.68a}$$

$$= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}} \mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,1} & \mathbf{F}_{\mathbf{xu}}^{t,1} \\ \mathbf{F}_{\mathbf{ux}}^{t,1} & \mathbf{F}_{\mathbf{uu}}^{t,1} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,n_x} & \mathbf{F}_{\mathbf{xu}}^{t,n_x} \\ \mathbf{F}_{\mathbf{ux}}^{t,n_x} & \mathbf{F}_{\mathbf{uu}}^{t,n_x} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}_t\|^2 + \|\tilde{\mathbf{u}}_t\|^2) \tag{6.68b}$$

$$= \mathbf{A}_t \tilde{\mathbf{x}}_t + \mathbf{B}_t \tilde{\mathbf{u}}_t + \mathbf{G}_t \mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,1} & \mathbf{F}_{\mathbf{xu}}^{t,1} \\ \mathbf{F}_{\mathbf{ux}}^{t,1} & \mathbf{F}_{\mathbf{uu}}^{t,1} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}_{\mathbf{xx}}^{t,n_x} & \mathbf{F}_{\mathbf{xu}}^{t,n_x} \\ \mathbf{F}_{\mathbf{ux}}^{t,n_x} & \mathbf{F}_{\mathbf{uu}}^{t,n_x} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2), \tag{6.68c}$$

$$\tilde{\mathbf{z}}_{t+1} = \mathbf{h}(\mathbf{x}_{t+1}) - \mathbf{h}(\mathbf{x}_{t+1}^p) + \epsilon \boldsymbol{\sigma}_{t+1}^{\mathbf{h}} \mathbf{v}_{t+1} \tag{6.68d}$$

$$= \mathbf{H}_{t+1} \tilde{\mathbf{x}}_{t+1} + \epsilon \boldsymbol{\sigma}_{t+1}^{\mathbf{h}} \mathbf{v}_{t+1} + \sum_{j=1}^{n_z} (\tilde{\mathbf{x}}_{t+1}^T \mathbf{H}_{t+1}^{h^j} \tilde{\mathbf{x}}_{t+1}) \mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}_{t+1}\|^2) \tag{6.68e}$$

$$= \mathbf{H}_{t+1} \tilde{\mathbf{x}}_{t+1} + \mathbf{M}_{t+1} \mathbf{v}_{t+1} + \sum_{j=1}^{n_z} (\tilde{\mathbf{x}}_{t+1}^T \mathbf{H}_{t+1}^{h^j} \tilde{\mathbf{x}}_{t+1}) \mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}\|_\infty^2), \tag{6.68f}$$

as $(\|\tilde{\mathbf{x}}\|_\infty + (\|\tilde{\mathbf{u}}\|_\infty) \downarrow 0$, where we have:

- $\mathbf{A}_t := \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{B}_t := \nabla_{\mathbf{u}} \mathbf{f}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{G}_t := \epsilon \boldsymbol{\sigma}_t^{\mathbf{f}}$;

- $\mathbf{H}_t := \nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x})|_{\mathbf{x}_t^p}$, $\mathbf{M}_t := \epsilon\boldsymbol{\sigma}_t^{\mathbf{h}}$.
- $\mathbf{f}(\mathbf{x},\mathbf{u}) = (f^j(\mathbf{x},\mathbf{u})), 1 \leq j \leq n_x$;
- $\mathbf{F}_{\mathbf{xx}}^{t,j} := \frac{1}{2}\nabla_{\mathbf{xx}}^2 f^j(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^p,\mathbf{u}_t^p}, \mathbf{F}_{\mathbf{xu}}^{t,j} := \frac{1}{2}\nabla_{\mathbf{xu}}^2 f^j(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^p,\mathbf{u}_t^p}, \mathbf{F}_{\mathbf{ux}}^{t,j} := \frac{1}{2}\nabla_{\mathbf{ux}}^2 f^j(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^p,\mathbf{u}_t^p}$, and $\mathbf{F}_{\mathbf{uu}}^{t,j} := \frac{1}{2}\nabla_{\mathbf{uu}}^2 f^j(\mathbf{x},\mathbf{u})|_{\mathbf{x}_t^p,\mathbf{u}_t^p}$;
- $\mathbf{h}(\mathbf{x}) = (h^j(\mathbf{x})), 1 \leq j \leq n_z$;
- $\mathbf{H}_t^{h^j} := \frac{1}{2}\nabla_{\mathbf{xx}}^2 h^j(\mathbf{x})|_{\mathbf{x}_t^p}$.

*The T-LQG feedback law:* For the analysis of this section, we apply the T-LQG feedback law $\tilde{\mathbf{u}}_t = -\mathbf{L}_t\hat{\hat{\mathbf{x}}}_t$ in the original system, where $\hat{\hat{\mathbf{x}}}_0 := \mathbf{0}$ and the state estimation mean error evolution is given as

$$\hat{\hat{\mathbf{x}}}_{t+1} = \mathbf{A}_t\hat{\hat{\mathbf{x}}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{K}_{t+1}(\tilde{\mathbf{z}}_{t+1} - \mathbf{H}_{t+1}(\mathbf{A}_t\hat{\hat{\mathbf{x}}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t)), \tag{6.69}$$

which is the same as (6.30). Next, we simplify the above equation as:

$$\begin{aligned}
\hat{\hat{\mathbf{x}}}_{t+1} &= \mathbf{A}_t\hat{\hat{\mathbf{x}}}_t - \mathbf{B}_t\mathbf{L}_t\hat{\hat{\mathbf{x}}}_t + \mathbf{K}_{t+1}(\tilde{\mathbf{z}}_{t+1} - \mathbf{H}_{t+1}(\mathbf{A}_t\hat{\hat{\mathbf{x}}}_t - \mathbf{B}_t\mathbf{L}_t\hat{\hat{\mathbf{x}}}_t)) && \text{(6.70a)}\\
&= (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H}_{t+1})(\mathbf{A}_t - \mathbf{B}_t\mathbf{L}_t)\hat{\hat{\mathbf{x}}}_t + \mathbf{K}_{t+1}\tilde{\mathbf{z}}_{t+1} && \text{(6.70b)}\\
&= \mathbf{K}_{t+1}^{\mathbf{L}}\hat{\hat{\mathbf{x}}}_t + \mathbf{K}_{t+1}\tilde{\mathbf{z}}_{t+1} && \text{(6.70c)}\\
&=: \tilde{\mathbf{K}}_{1:t+1}^{\mathbf{L}}\hat{\hat{\mathbf{x}}}_0 + \sum_{s=1}^{t+1}\tilde{\mathbf{K}}_{s+1:t+1}^{\mathbf{L}}\mathbf{K}_s\tilde{\mathbf{z}}_s && \text{(6.70d)}\\
&= \sum_{s=1}^{t+1}\tilde{\mathbf{K}}_{s+1:t+1}^{\mathbf{L}}\mathbf{K}_s\tilde{\mathbf{z}}_s, && \text{(6.70e)}
\end{aligned}$$

where $\mathbf{K}_{t+1}^{\mathbf{L}} := (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H}_{t+1})(\mathbf{A}_t - \mathbf{B}_t\mathbf{L}_t), 0 \leq t \leq K-1$, $\tilde{\mathbf{K}}_{t_1:t_2}^{\mathbf{L}} = \Pi_{t=t_1}^{t_2}\mathbf{K}_t^{\mathbf{L}}, t_2 \geq t_1 \geq 1$, otherwise, it is the identity matrix. Also we solved the following recursion equation:

$$\hat{\hat{\mathbf{x}}}_{t+1} = \mathbf{K}_{t+1}^{\mathbf{L}}\hat{\hat{\mathbf{x}}}_t + \mathbf{K}_{t+1}\tilde{\mathbf{z}}_{t+1}$$

$$
\begin{aligned}
&= \mathbf{K}^{\mathbf{L}}_{t+1}(\mathbf{K}^{\mathbf{L}}_{t}\hat{\tilde{\mathbf{x}}}_{t-1}+\mathbf{K}_{t}\tilde{\mathbf{z}}_{t})+\mathbf{K}_{t+1}\tilde{\mathbf{z}}_{t+1}\\
&= \mathbf{K}^{\mathbf{L}}_{t+1}\mathbf{K}^{\mathbf{L}}_{t}\hat{\tilde{\mathbf{x}}}_{t-1}+\mathbf{K}^{\mathbf{L}}_{t+1}\mathbf{K}_{t}\tilde{\mathbf{z}}_{t}+\mathbf{K}_{t+1}\tilde{\mathbf{z}}_{t+1}\\
&= \mathbf{K}^{\mathbf{L}}_{t+1}\mathbf{K}^{\mathbf{L}}_{t+1-1}\times\cdots\times\mathbf{K}^{\mathbf{L}}_{t+1-t}\hat{\tilde{\mathbf{x}}}_{t-t}+\sum_{r=0}^{t}(\mathbf{K}^{\mathbf{L}}_{t+1}\times\cdots\times\mathbf{K}^{\mathbf{L}}_{t+1-r+1})\mathbf{K}_{t+1-r}\tilde{\mathbf{z}}_{t+1-r}\\
&= \tilde{\mathbf{K}}^{\mathbf{L}}_{1:t+1}\hat{\tilde{\mathbf{x}}}_{0}+\sum_{r=0}^{t}\tilde{\mathbf{K}}^{\mathbf{L}}_{t+2-r:t+1}\mathbf{K}_{t+1-r}\tilde{\mathbf{z}}_{t+1-r}\\
&= \tilde{\mathbf{K}}^{\mathbf{L}}_{1:t+1}\hat{\tilde{\mathbf{x}}}_{0}+\sum_{s=1}^{t+1}\tilde{\mathbf{K}}^{\mathbf{L}}_{s+1:t+1}\mathbf{K}_{s}\tilde{\mathbf{z}}_{s},
\end{aligned}
$$

where in the last equation, we relabeled $s = t + 1 - r$.

*Rewriting the feedback law:* Using the above equation, we can rewrite the T-LQR feedback law as:

$$
\begin{aligned}
\tilde{\mathbf{u}}_t = -\mathbf{L}_t\hat{\tilde{\mathbf{x}}}_t = -\mathbf{L}_t\sum_{s=1}^{t}\tilde{\mathbf{K}}^{\mathbf{L}}_{s+1:t}\mathbf{K}_s\tilde{\mathbf{z}}_s = -\sum_{s=1}^{t}\mathbf{L}_t\tilde{\mathbf{K}}^{\mathbf{L}}_{s+1:t}\mathbf{K}_s\tilde{\mathbf{z}}_s \qquad (6.71a)\\
= -\sum_{s=0}^{t}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s, \qquad (6.71b)
\end{aligned}
$$

where $\mathbf{L}_{s,t} := \mathbf{L}_t\tilde{\mathbf{K}}^{\mathbf{L}}_{s+1:t}\mathbf{K}_s, 1 \leq s \leq t, 0 \leq t \leq K-1$ and $\mathbf{L}_{s,t} := \mathbf{0}, s = 0, 0 \leq t \leq K-1$. Note that this feedback law is similar to the law in the previous section, except that the law does not include second-order terms in $\tilde{\mathbf{z}}$. However, the process and observation models include second-order terms. Also, we will use the following form of the control law in the proofs:

$$
\begin{aligned}
\tilde{\mathbf{u}}_t &= -\sum_{s=0}^{t}\mathbf{L}_{s,t}(\mathbf{H}_s\tilde{\mathbf{x}}_s+\mathbf{M}_s\mathbf{v}_s+\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{e}_j^{n_z})+o(\|\tilde{\mathbf{x}}\|_\infty^2)\\
&= -\sum_{s=0}^{t}\mathbf{L}_{s,t}\mathbf{H}_s\tilde{\mathbf{x}}_s-\sum_{s=0}^{t}\mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s-\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{L}_{s,t}\mathbf{e}_j^{n_z}+o(\|\tilde{\mathbf{x}}\|_\infty^2).
\end{aligned}
$$

Now, we can simplify the second-order expansion of the dynamics:

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{G}_t\mathbf{w}_t + \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,1}_{\mathbf{xx}} & \mathbf{F}^{t,1}_{\mathbf{xu}} \\ \mathbf{F}^{t,1}_{\mathbf{ux}} & \mathbf{F}^{t,1}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,n_x}_{\mathbf{xx}} & \mathbf{F}^{t,n_x}_{\mathbf{xu}} \\ \mathbf{F}^{t,n_x}_{\mathbf{ux}} & \mathbf{F}^{t,n_x}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ \tilde{\mathbf{u}}_t \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2) \tag{6.72a}$$

$$\begin{aligned} =&\mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t(-\sum_{s=0}^{t}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s) + \mathbf{G}_t\mathbf{w}_t \\ &+ \begin{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\sum_{s=0}^{t}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,1}_{\mathbf{xx}} & \mathbf{F}^{t,1}_{\mathbf{xu}} \\ \mathbf{F}^{t,1}_{\mathbf{ux}} & \mathbf{F}^{t,1}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\sum_{s=0}^{t}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\sum_{s=0}^{t}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s \end{pmatrix}^T \begin{pmatrix} \mathbf{F}^{t,n_x}_{\mathbf{xx}} & \mathbf{F}^{t,n_x}_{\mathbf{xu}} \\ \mathbf{F}^{t,n_x}_{\mathbf{ux}} & \mathbf{F}^{t,n_x}_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_t \\ -\sum_{s=0}^{t}\mathbf{L}_{s,t}\tilde{\mathbf{z}}_s \end{pmatrix} \end{pmatrix} + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{z}}\|_\infty^2) \end{aligned} \tag{6.72b}$$

$$\begin{aligned} =&\mathbf{A}_t\tilde{\mathbf{x}}_t - \sum_{s=0}^{t}\mathbf{B}_t\mathbf{L}_{s,t}(\mathbf{H}_s\tilde{\mathbf{x}}_s + \mathbf{M}_s\mathbf{v}_s + \sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{e}_j^{n_z}) + \mathbf{G}_t\mathbf{w}_t \\ &+ \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{x}}_i^T\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j\mathbf{e}_k^{n_x} + \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\tilde{\mathbf{x}}_i^T\mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} \\ &+ \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}\mathbf{v}_i^T\mathbf{H}_t^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \end{aligned} \tag{6.72c}$$

$$\begin{aligned} =&\mathbf{A}_t\tilde{\mathbf{x}}_t - \sum_{s=0}^{t}\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{H}_s\tilde{\mathbf{x}}_s - \sum_{s=0}^{t}\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s + \mathbf{G}_t\mathbf{w}_t \\ &+ \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j\mathbf{e}_k^{n_x} + \tilde{\mathbf{x}}_i^T\mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} + \mathbf{v}_i^T\mathbf{H}_t^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j)\mathbf{e}_k^{n_x} \\ &- \sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \end{aligned} \tag{6.72d}$$

$$
\begin{aligned}
&= \sum_{s=0}^{t} \mathbf{U}_{s,t}\tilde{\mathbf{x}}_s + \sum_{s=0}^{t} \mathbf{V}_{s,t}\mathbf{v}_s + \mathbf{G}_t\mathbf{w}_t \\
&\quad + \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T \mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j\mathbf{e}_k^{n_x} + \tilde{\mathbf{x}}_i^T \mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} + \mathbf{v}_i^T \mathbf{H}_t^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j)\mathbf{e}_k^{n_x} \\
&\quad - \sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T \mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}\|_\infty^2),
\end{aligned}
\tag{6.72e}
$$

where

- $\mathbf{U}_{s,t} := -\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{H}_s, 0 \le s \le t-1;$
- $\mathbf{U}_{s,t} := \mathbf{A}_s - \mathbf{B}_t\mathbf{L}_{s,t}\mathbf{H}_s, s = t;$
- $\mathbf{V}_{s,t} := -\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{M}_s, 0 \le s \le t;$
- $\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}} := \mathbf{H}_i^T\mathbf{L}_{i,t}^T\mathbf{F}_{\mathbf{uu}}^{t,k}\mathbf{L}_{j,t}\mathbf{H}_j, 0 \le i \le t-1, 0 \le j \le t-1;$
- $\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}} := (-2\mathbf{F}_{\mathbf{xx}}^{t,k}\mathbf{L}_{j,t}\mathbf{H}_j + 2\mathbf{H}_t^T\mathbf{L}_{t,t}^T\mathbf{F}_{\mathbf{uu}}^{t,k}\mathbf{L}_{j,t}\mathbf{H}_j), i = t, 0 \le j \le t-1;$
- $\mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}} := (\mathbf{F}_{\mathbf{xx}}^{t,k} - 2\mathbf{F}_{\mathbf{xx}}^{t,k}\mathbf{L}_{t,t}\mathbf{H}_t + \mathbf{H}_t^T\mathbf{L}_{t,t}^T\mathbf{F}_{\mathbf{uu}}^{t,k}\mathbf{L}_{t,t}\mathbf{H}_t), i = j = t;$
- $\mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}} := 2\mathbf{H}_i^T\mathbf{L}_{i,t}^T\mathbf{F}_{\mathbf{uu}}^{t,k}\mathbf{L}_{j,t}\mathbf{M}_j, 0 \le i \le t-1, 0 \le j \le t;$
- $\mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}} := (2\mathbf{H}_t^T\mathbf{L}_{t,t}^T\mathbf{F}_{\mathbf{uu}}^{t,k}\mathbf{L}_{j,t}\mathbf{M}_j - 2\mathbf{F}_{\mathbf{xx}}^{t,k}\mathbf{L}_{j,t}\mathbf{M}_j), i = t, 0 \le j \le t;$ and
- $\mathbf{H}_t^{(\mathbf{v},f,\mathbf{v})^{kij}} := \mathbf{M}_i^T\mathbf{L}_{i,t}^T\mathbf{F}_{\mathbf{uu}}^{t,k}\mathbf{L}_{j,t}\mathbf{M}_j, 0 \le i \le t, 0 \le j \le t.$

are as defined before in Section 6.3.

Next, similar to (6.63d), for (6.72e) we solve the linear recursion in $\tilde{\mathbf{x}}$:

$$
\begin{aligned}
\tilde{\mathbf{x}}_{t+1} &= \sum_{s=0}^{t} \mathbf{U}_{s,t}\tilde{\mathbf{x}}_s + \sum_{s=0}^{t} \mathbf{V}_{s,t}\mathbf{v}_s + \mathbf{G}_t\mathbf{w}_t \\
&\quad + \sum_{k=1}^{n_x}\sum_{i=0}^{t}\sum_{j=0}^{t}(\tilde{\mathbf{x}}_i^T \mathbf{H}_t^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j\mathbf{e}_k^{n_x} + \tilde{\mathbf{x}}_i^T \mathbf{H}_t^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j\mathbf{e}_k^{n_x} + \mathbf{v}_i^T \mathbf{H}_t^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j)\mathbf{e}_k^{n_x} \\
&\quad - \sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T \mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{B}_t\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \qquad (6.73a) \\
&= (\sum_{s=0}^{t}\mathbf{Q}_s\mathbf{U}_{0,t-s})\mathbf{x}_0 + \sum_{s=0}^{t}\sum_{r=0}^{t-s}\mathbf{Q}_s\mathbf{V}_{r,t-s}\mathbf{v}_r + \sum_{s=0}^{t}\mathbf{Q}_s\mathbf{G}_{t-s}\mathbf{w}_{t-s}
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{s=0}^{t}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s}\sum_{j=0}^{t-s}\Big(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}_{t-s}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j\Big)\mathbf{Q}_s\mathbf{e}_k^{n_x}\\
&-\sum_{s=0}^{t}\sum_{r=0}^{t-s}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{h^j}\tilde{\mathbf{x}}_r)\mathbf{Q}_s\mathbf{B}_{t-s}\mathbf{L}_{r,t-s}\mathbf{e}_j^{n_z}+o(\|\tilde{\mathbf{x}}\|_\infty^2) \qquad (6.73b)\\
&=\mathbf{U}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0+\sum_{s=0}^{t}(\mathbf{V}_{s,t}^{\mathbf{v}}\mathbf{v}_s+\mathbf{W}_{s,t}^{\mathbf{w}}\mathbf{w}_s)\\
&+\sum_{s=0}^{t}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s}\sum_{j=0}^{t-s}\Big(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}_{t-s}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j\Big)\mathbf{Q}_s\mathbf{e}_k^{n_x}\\
&-\sum_{s=0}^{t}\sum_{r=0}^{t-s}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{h^j}\tilde{\mathbf{x}}_r)\mathbf{Q}_s\mathbf{B}_{t-s}\mathbf{L}_{r,t-s}\mathbf{e}_j^{n_z}+o(\|\tilde{\mathbf{x}}\|_\infty^2), \qquad (6.73c)
\end{aligned}
$$

where

- $\mathbf{Q}_s := \sum_{r=0}^{s-1}\mathbf{Q}_r\mathbf{U}_{t-s+1,t-r}, 1 \le s \le t, 0 \le t \le K-1$ and $\mathbf{Q}_0 := 1$;
- $\mathbf{U}_t^{\mathbf{x}_0} := (\sum_{s=0}^{t}\mathbf{Q}_s\mathbf{U}_{0,t-s}), 0 \le t \le K-1$;
- $\mathbf{V}_{s,t}^{\mathbf{v}} := \sum\limits_{r=0}^{t-s}\mathbf{Q}_r\mathbf{V}_{s,t-r}, 0 \le s \le t, 0 \le t \le K-1$; and
- $\mathbf{W}_{s,t}^{\mathbf{w}} := \mathbf{Q}_{s-t}\mathbf{G}_s, 0 \le s \le t, 0 \le t \le K-1$.

*Validity region:* Similar to the analysis of Section 6.3, we have defined the state error as $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}_t^p$. Moreover, we have proven the properties of $O(\|\tilde{\mathbf{x}}\|_\infty)$ for a system compensated with the T-LQG law in Section 6.2. Particularly, we proved that for the T-LQG design, $O(\|\tilde{\mathbf{x}}\|_\infty) = O(\delta)$ in a set $\Omega(\delta)$ properly defined as before with probability $1 - o(\epsilon)$. Hence, for $\omega \in \Omega(\delta)$, $O(\|\tilde{\mathbf{x}}_t\|_\infty^2) = O(\delta^2)$. Therefore, for $\omega \in \Omega(\delta)$ (the same set and with the same probability), we have:

$$
\begin{aligned}
\tilde{\mathbf{x}}_{t+1} =&\mathbf{U}_t^{\mathbf{x}_0}\tilde{\mathbf{x}}_0+\sum_{s=0}^{t}(\mathbf{V}_{s,t}^{\mathbf{v}}\mathbf{v}_s+\mathbf{W}_{s,t}^{\mathbf{w}}\mathbf{w}_s)\\
&+\sum_{s=0}^{t}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s}\sum_{j=0}^{t-s}\Big(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}_{t-s}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j\Big)\mathbf{Q}_s\mathbf{e}_k^{n_x}\\
&-\sum_{s=0}^{t}\sum_{r=0}^{t-s}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{h^j}\tilde{\mathbf{x}}_r)\mathbf{Q}_s\mathbf{B}_{t-s}\mathbf{L}_{r,t-s}\mathbf{e}_j^{n_z}+O(\delta^2). \qquad (6.74)
\end{aligned}
$$

*Second-order expansion of the cost function:* Similarly, we obtain the second-order Taylor series expansion of the cost function around the nominal trajectory:

$$J = J^p + \tilde{J}_1 + \tilde{J}_2 + o(\sum_{t=1}^{K-1} (\|\tilde{\mathbf{x}}_t\|^2 + \|\tilde{\mathbf{u}}_t\|^2) + \|\tilde{\mathbf{x}}_K\|^2) \tag{6.75a}$$

$$= J^p + \tilde{J}_1 + \tilde{J}_2 + o(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2), \tag{6.75b}$$

as $(\|\tilde{\mathbf{x}}\|_\infty^2 + \|\tilde{\mathbf{u}}\|_\infty^2) \downarrow 0$. Moreover, we have:

- $J^p{:=}\sum_{t=0}^{K-1} c_t(\mathbf{x}_t^p, \mathbf{u}_t^p) + c_K(\mathbf{x}_K^p)$ denotes the nominal cost;
- $\tilde{J}_1{:=}\sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t + \mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t) + \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K$ is the first order cost error;
- $\tilde{J}_2 := \sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xx}}\tilde{\mathbf{x}}_t + \frac{1}{2}\tilde{\mathbf{u}}_t^T\mathbf{C}_t^{\mathbf{uu}}\tilde{\mathbf{u}}_t + \tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{xu}}\tilde{\mathbf{u}}_t) + \frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}_K^{\mathbf{xx}}\tilde{\mathbf{x}}_K$ is the second order cost error.
- $J_2 := J^p + \tilde{J}_1 + \tilde{J}_2$ is the second order approximation of the cost function;
- $\mathbf{C}_t^{\mathbf{xx}} = \nabla_{\mathbf{xx}}^2 c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{uu}} = \nabla_{\mathbf{uu}}^2 c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, $\mathbf{C}_t^{\mathbf{xu}} = \nabla_{\mathbf{xu}}^2 c_t(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^p, \mathbf{u}_t^p}$, and $\mathbf{C}_K^{\mathbf{xx}} = \nabla_{\mathbf{xx}}^2 c_K(\mathbf{x})|_{\mathbf{x}_K^p}$, where we have used the fact that $c_t \in \mathbb{C}^2$.

Next, we provide the main result regarding the expected second order error of the cost function.

**Theorem 10 Second-Order Cost Function Error for a Partially-Observed System with T-LQG Policy:** *Given that process noises are zero mean i.i.d. Gaussian, the initial error is zero mean Gaussian, and all the functions are in $\mathbb{C}^2$, under a first-order approximation for the small noise paradigm, the stochastic cost function is dominated by the nominal part of the cost function, and the expected first-order error is $O(\delta^2)$. That is,*

$$\mathbb{E}[\tilde{J}_1] = O(\delta^2), \text{ and } \mathbb{E}[J] = J^p + O(\delta^2).$$

*Moreover, by choosing $\delta = \sqrt{2\log(\frac{1}{\epsilon})}\epsilon$, we have*

$$\mathbb{E}[\tilde{J}_1] = O(\epsilon^{2-\gamma}), \text{ and } \mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma}),$$

*for some $0 < \gamma \ll 1$, which shows that this error tends to zero with a near-first-order rate as $\epsilon \downarrow 0$.*

***Proof 24*** *First, let us simply the first order cost error:*

$$\begin{aligned}
\tilde{J}_1 =& \sum_{t=0}^{K-1}(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t+\mathbf{C}_t^{\mathbf{u}}\tilde{\mathbf{u}}_t)+\mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K \\
=&\sum_{t=0}^{K-1}\Big(\mathbf{C}_t^{\mathbf{x}}\tilde{\mathbf{x}}_t - \sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{H}_s\tilde{\mathbf{x}}_s - \sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s - \sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z}\Big) \\
&+ \mathbf{C}_K^{\mathbf{x}}\tilde{\mathbf{x}}_K + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
=&\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\tilde{\mathbf{x}}_t - \sum_{t=0}^{K-1}\sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s - \sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
=&\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\Big(\mathbf{U}_{t-1}^{\mathbf{x}_0}\tilde{\mathbf{x}}_0+\sum_{s=0}^{t-1}(\mathbf{V}_{s,t-1}^{\mathbf{v}}\mathbf{v}_s+\mathbf{W}_{s,t-1}^{\mathbf{w}}\mathbf{w}_s) \\
&+\sum_{s=0}^{t-1}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j)\mathbf{Q}_s\mathbf{e}_k^{n_x} \\
&-\sum_{s=0}^{t-1}\sum_{r=0}^{t-s-1}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{h^j}\tilde{\mathbf{x}}_r)\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{L}_{r,t-s-1}\mathbf{e}_j^{n_z}\Big) \\
&-\sum_{t=0}^{K-1}\sum_{s=0}^{t}\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{M}_s\mathbf{v}_s - \sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}\|_\infty^2) \\
=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}})\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t \\
&+\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{e}_k^{n_x} \\
&-\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{r=0}^{t-s-1}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{h^j}\tilde{\mathbf{x}}_r)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{L}_{r,t-s-1}\mathbf{e}_j^{n_z} \\
&-\sum_{t=0}^{K-1}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t - \sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z} + o(\|\tilde{\mathbf{x}}\|_\infty^2)
\end{aligned}$$

$$
\begin{aligned}
&=(\sum_{t=0}^{K}\mathbf{C}^{\mathbf{L}}_t\mathbf{U}^{\mathbf{x}_0}_{t-1})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}^{\mathbf{L}}_s\mathbf{V}^{\mathbf{v}}_{t,s-1}-\mathbf{C}^{\mathbf{u}}_s\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}^{\mathbf{L}}_s\mathbf{W}^{\mathbf{w}}_{t,s-1})\mathbf{w}_t\\
&\quad+\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}^{(\mathbf{h},f,\mathbf{h})^{kij}}_{t-s-1}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}^{(\mathbf{h},f,\mathbf{v})^{kij}}_{t-s-1}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}^{(\mathbf{v},f,\mathbf{v})^{kij}}_{t-s-1}\mathbf{v}_j)\mathbf{C}^{\mathbf{L}}_t\mathbf{Q}_s\mathbf{e}^{n_x}_k\\
&\quad-\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{r=0}^{t-s-1}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}^{h^j}_r\tilde{\mathbf{x}}_r)\mathbf{C}^{\mathbf{L}}_t\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{L}_{r,t-s-1}\mathbf{e}^{n_z}_j\\
&\quad-\sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}^{h^j}_s\tilde{\mathbf{x}}_s)\mathbf{C}^{\mathbf{u}}_t\mathbf{L}_{s,t}\mathbf{e}^{n_z}_j+o(\|\tilde{\mathbf{x}}\|^2_\infty),
\end{aligned}
$$

*where* $\mathbf{C}^{\mathbf{L}}_t := \mathbf{C}^{\mathbf{x}}_t - \sum\limits_{s=t}^{K-1}\mathbf{C}^{\mathbf{u}}_s\mathbf{L}_{t,s}\mathbf{H}_t, 0 \le t \le K-1,$ *and* $\mathbf{C}^{\mathbf{L}}_K = \mathbf{C}^{\mathbf{x}}_K$. *Note that in evaluating the above expression, we used the following summation exchange formula*

$$
\sum_{t=0}^{K-1}\sum_{s=0}^{t}f_{t,s}x_s=\sum_{s=0}^{K-1}\sum_{t=s}^{K-1}f_{t,s}x_s=\sum_{s=0}^{K-1}(\sum_{t=s}^{K-1}f_{t,s})x_s=\sum_{t=0}^{K-1}(\sum_{s=t}^{K-1}f_{s,t})x_t.
$$

*For instance, we simplified the following expression:*

$$
\sum_{t=0}^{K-1}\sum_{s=0}^{t}\mathbf{C}^{\mathbf{u}}_t\mathbf{L}_{s,t}\mathbf{H}_s\tilde{\mathbf{x}}_s=\sum_{t=0}^{K-1}(\sum_{s=t}^{K-1}\mathbf{C}^{\mathbf{u}}_s\mathbf{L}_{t,s}\mathbf{H}_t)\tilde{\mathbf{x}}_t.
$$

*Next, we simplify the second order cost error. Once again, we ignore the second order feedback terms and replace them with* $o(\|\tilde{\mathbf{x}}\|^2_\infty)$

$$
\begin{aligned}
\tilde{J}_2&=\sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}^{\mathbf{xx}}_t\tilde{\mathbf{x}}_t+\frac{1}{2}\tilde{\mathbf{u}}_t^T\mathbf{C}^{\mathbf{uu}}_t\tilde{\mathbf{u}}_t+\tilde{\mathbf{x}}_t^T\mathbf{C}^{\mathbf{xu}}_t\tilde{\mathbf{u}}_t)+\frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}^{\mathbf{xx}}_K\tilde{\mathbf{x}}_K\\
&=\sum_{t=0}^{K-1}(\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{C}^{\mathbf{xx}}_t\tilde{\mathbf{x}}_t+\frac{1}{2}\tilde{\mathbf{x}}_t^T\mathbf{L}_t^T\mathbf{C}^{\mathbf{uu}}_t\mathbf{L}_t\tilde{\mathbf{x}}_t+\tilde{\mathbf{x}}_t^T\mathbf{C}^{\mathbf{xu}}_t\mathbf{L}_t\tilde{\mathbf{x}}_t)+\frac{1}{2}\tilde{\mathbf{x}}_K^T\mathbf{C}^{\mathbf{xx}}_K\tilde{\mathbf{x}}_K+o(\|\tilde{\mathbf{x}}\|^2_\infty)\\
&=\sum_{t=0}^{K}\tilde{\mathbf{x}}_t^T\mathbf{C}^{\mathbf{LL}}_t\tilde{\mathbf{x}}_t+o(\|\tilde{\mathbf{x}}\|^2_\infty),
\end{aligned}
$$

*where* $\mathbf{C}^{\mathbf{LL}}_t := \frac{1}{2}\mathbf{C}^{\mathbf{xx}}_t + \frac{1}{2}\mathbf{L}_t^T\mathbf{C}^{\mathbf{uu}}_t\mathbf{L}_t + \mathbf{C}^{\mathbf{xu}}_t\mathbf{L}_t, 0 \le t \le K-1,$ *and* $\mathbf{C}^{\mathbf{LL}}_K := \frac{1}{2}\mathbf{C}^{\mathbf{xx}}_K$. *Hence,* *we have:*

$$\begin{aligned}\tilde{J}_1+\tilde{J}_2=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t\\&+\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{h})^{kij}}\tilde{\mathbf{x}}_j+\tilde{\mathbf{x}}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{h},f,\mathbf{v})^{kij}}\mathbf{v}_j+\mathbf{v}_i^T\mathbf{H}_{t-s-1}^{(\mathbf{v},f,\mathbf{v})^{kij}}\mathbf{v}_j)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{e}_k^{n_x}\\&-\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{r=0}^{t-s-1}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T\mathbf{H}_r^{h^j}\tilde{\mathbf{x}}_r)\mathbf{C}_t^{\mathbf{L}}\mathbf{Q}_s\mathbf{B}_{t-s-1}\mathbf{L}_{r,t-s-1}\mathbf{e}_j^{n_z}\\&-\sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T\mathbf{H}_s^{h^j}\tilde{\mathbf{x}}_s)\mathbf{C}_t^{\mathbf{u}}\mathbf{L}_{s,t}\mathbf{e}_j^{n_z}+\sum_{t=0}^{K}\tilde{\mathbf{x}}_t^T\mathbf{C}_t^{\mathbf{LL}}\tilde{\mathbf{x}}_t+o(\|\tilde{\mathbf{x}}\|_\infty^2)\end{aligned}\tag{6.76}$$

*Therefore, we can simplify the above expression as follows:*

$$\begin{aligned}\tilde{J}_1+\tilde{J}_2=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t\\&+O(\|\tilde{\mathbf{x}}\|_\infty^2)+o(\|\tilde{\mathbf{x}}\|_\infty^2)\\=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t\\&+O(\|\tilde{\mathbf{x}}\|_\infty^2)\end{aligned}$$

*Hence, for* $\omega\in\Omega(\delta)$,

$$\begin{aligned}\tilde{J}_1+\tilde{J}_2=&(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t\\&+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t+O(\delta^2).\end{aligned}$$

*As a result, using* (6.75b), *for* $\omega\in\Omega(\delta)$, *we have:*

$$\begin{aligned}J=&J^p+(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t\\&+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t+O(\delta^2).\end{aligned}$$

*Next, note* $\mathbb{E}[\tilde{\mathbf{x}}_0] = \mathbb{E}[\mathbf{x}_0 - \hat{\mathbf{x}}_0] = \mathbf{0}$*, and* $\mathbb{E}[\mathbf{w}_t] = \mathbb{E}[\mathbf{v}_t] = 0$ *for all* $t$*. Therefore,*

$$\begin{aligned}&\mathbb{E}[(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\tilde{\mathbf{x}}_0+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbf{w}_t]\\ &=(\sum_{t=0}^{K}\mathbf{C}_t^{\mathbf{L}}\mathbf{U}_{t-1}^{\mathbf{x}_0})\mathbb{E}[\tilde{\mathbf{x}}_0]+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{V}_{t,s-1}^{\mathbf{v}}-\mathbf{C}_s^{\mathbf{u}}\mathbf{L}_{t,s}\mathbf{M}_t)\mathbb{E}[\mathbf{v}_t]\\ &\quad+\sum_{t=0}^{K}(\sum_{s=t}^{K-1}\mathbf{C}_s^{\mathbf{L}}\mathbf{W}_{t,s-1}^{\mathbf{w}})\mathbb{E}[\mathbf{w}_t]=0.\end{aligned}$$

*Now, since for* $\omega \notin \Omega(\delta)$*,* $J \leq M$*, then*

$$\begin{aligned}\mathbb{E}[J - J^p] &= P(\Omega(\delta))(0 + O(\delta^2)) + M(1 - P(\Omega(\delta)))\\ &= P(\Omega(\delta))O(\delta) + M(1 - P(\Omega(\delta))) \qquad (6.77)\end{aligned}$$

*Note the last expression is the same as* (4.47)*. Although* $\Omega(\delta)$ *is not the same as in Theorem 4,* $P(\Omega(\delta))$ *is still the same. In the proof of Theorem 4 while we discussed on the probabilistic argument and choosing the proper* $\delta$*, we showed that by choosing* $\delta := \sqrt{-2\log(\epsilon)}\epsilon$*, the* $\mathbb{E}[J - J^p] = O(\epsilon^{2-\gamma})$*. Thus,* $\mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma})$*. Similarly,* $\mathbb{E}[\tilde{J}_1 + \tilde{J}_2] = O(\epsilon^{2-\gamma})$*. The same argument follows through and this theorem is proved.*

Hence, when the functions are in $\mathbb{C}^2$, the expected stochastic cost is equal to the nominal cost with a higher probability as $\epsilon \downarrow 0$. Therefore, it follows that the deterministic policy is near-second-order optimal, summarized below:

**Corollary 8 Decoupling Principle: Near-Second-Order Optimality for a Partially-Observed System.** *Based on Theorem 10, for a partially-observed system under the small noise paradigm, as* $\epsilon \downarrow 0$*, the decoupling principle holds with* $O(\epsilon^{2-\gamma})$*-optimality for* $0 < \gamma \ll 1$*. Moreover, the T-LQG approach (s linear policy designed based on this result) is* $O(\epsilon^{2-\gamma})$*-optimal.*

***Proof 25*** *Using Theorem 10, for* $\omega \in \Omega(\delta)$ *we have* $\mathbb{E}[J] = J^p + O(\epsilon^{2-\gamma})$*, which is* *the cost of applying policy* $\boldsymbol{\pi}_t(\mathbf{z}_{0:t}) = \mathbf{u}_t^p - \mathbf{L}_t(\hat{\mathbf{x}}_t - \mathbf{x}_t^p)$ *to the stochastic system. Now, suppose* $\boldsymbol{\pi}^*$ *is the optimal stochastic policy. We showed in the proof of Corollary 7 that for this policy, we have* $\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{2-\gamma})$*. Now, by construction* $J^p \leq J^{*p}$*, and*

$$\mathbb{E}[J_{\boldsymbol{\pi}^*}] = J^{*p} + O(\epsilon^{2-\gamma}) \geq J^p + O(\epsilon^{2-\gamma}) = \mathbb{E}[J_{\boldsymbol{\pi}}] + O(\epsilon^{2-\gamma})$$

*As a result, policy* $\boldsymbol{\pi}$ *is within* $O(\epsilon^{2-\gamma})$ *of the optimal stochastic policy.*

# 7. PARTIALY-OBSERVED MULTI-AGENT SYSTEM

In this chapter, we generalize the single-agent results of Result 3 to a multi-agent partially-observed system. The generalization is done in a manner similar to the fully-observed case where we create a centralized multi-agent system through appropriate concatenations of the variables.

## 7.1 Multi-Agent Decoupling of Open-Loop and Closed-Loop Designs

In this section, we generalize the single-agent results of Result 3 for a multi-agent partially-observed system. The generalization is straightforward by observing the fact that a centralized multi-agent can be considered as one big single-agent system by defining appropriate concatenations of the variables.

*One joint system:* First, we concatenate the equations of control and state evolutions for all agents and consider the entire multi-agent system as one system similar to the fully-observed case. Hence, we have:

$$\mathbf{x}^{\mathcal{I}}_{t+1} = \mathbf{f}^{\mathcal{I}}(\mathbf{x}^{\mathcal{I}}_t, \mathbf{u}^{\mathcal{I}}_t) + \epsilon \boldsymbol{\sigma}^{\mathbf{f}_{\mathcal{I}}}_t \mathbf{w}^{\mathcal{I}}_t, \tag{7.1}$$

$$\mathbf{z}^{\mathcal{I}}_{t} = \mathbf{h}^{\mathcal{I}}(\mathbf{x}^{\mathcal{I}}_t) + \epsilon \boldsymbol{\sigma}^{\mathbf{h}_{\mathcal{I}}}_t \mathbf{v}^{\mathcal{I}}_t. \tag{7.2}$$

**Remark 4** *Corollary 6 states that for the multi-agent system of* (7.1) *with index* $\mathcal{I}$*, if functions are in* $\mathbb{C}^1$*, the first order approximation of the cost function does not depend on the linear feedback gain, rather, it is completely determined by the nominal trajectory. Moreover, if functions are in* $\mathbb{C}^2$*, based on Corollary 8 the second-order approximation of the cost function is also dominated by the nominal cost. This leads to the extension of the decoupling of open-loop/closed-loop deigns for a multi-agent partially-observed system. That is, under small noise, the multi-agent version* *of Problem (5) can be near-optimally separated into two problems: i) an open-loop optimal control problem to design the nominal trajectories of the system, and ii) a design of the optimal feedback law to track the nominal trajectories of the system. This is elaborated next.*

**Problem 12 (Nominal Trajectory Design Problem)** *Given an initial joint state $\bar{\mathbf{x}}_0^{\mathcal{I}}$, solve:*

$$\begin{aligned} \min_{\mathbf{u}^{\mathcal{I}}_{0:K_{\mathcal{I}}-1}} & \sum_{t=0}^{K_{\mathcal{I}}-1} c_t(\mathbf{x}_t^{\mathcal{I}}, \mathbf{u}_t^{\mathcal{I}}) + c_{K_{\mathcal{I}}}(\mathbf{x}_{K_{\mathcal{I}}}^{\mathcal{I}})] \\ s.t.\ & \mathbf{x}_{t+1}^{\mathcal{I}} = \mathbf{f}^{\mathcal{I}}(\mathbf{x}_t^{\mathcal{I}}, \mathbf{u}_t^{\mathcal{I}}) \\ & \mathbf{z}_t^{\mathcal{I}} = \mathbf{h}^{\mathcal{I}}(\mathbf{x}_t^{\mathcal{I}}). \end{aligned}$$

*Nominal trajectories:* Given the initial state $\bar{\mathbf{x}}_0^{\mathcal{I}}$, and using the optimized nominal controls of the above problem, $\mathbf{u}_t^{p\mathcal{I}}$, the nominal trajectory of the multi-agent system is defined as:

$$\mathbf{x}_{t+1}^{p\mathcal{I}} = \mathbf{f}^{\mathcal{I}}(\mathbf{x}_t^{p\mathcal{I}}, \mathbf{u}_t^{p\mathcal{I}}),\ \mathbf{z}_{t+1}^{p\mathcal{I}} = \mathbf{h}^{\mathcal{I}}(\mathbf{x}_t^{p\mathcal{I}}), \tag{7.3}$$

where $\mathbf{x}_0^{p\mathcal{I}} := \mathbf{x}_0^{\mathcal{I}}$, and $\mathbf{x}_{t+1}^{p_i} = \mathbf{f}^i(\mathbf{x}_t^{p_i}, \mathbf{u}_t^{p_i})$, and $\mathbf{h}_t^{p_i} = \mathbf{h}^i(\mathbf{x}_t^{p_i})$ for $i \in \mathcal{I}$.

*Linearized system:* Using the result of the previous chapter and using a feedback policy for each agent that depends on the entire system's mean of estimate, we can write the linearized system for each agent as:

$$\begin{aligned} \mathbf{x}_{t+1}^{\mathcal{I}} =& \mathbf{x}_{t+1}^{p\mathcal{I}} + \mathbf{A}_t^{\mathcal{I}}(\mathbf{x}_t^{\mathcal{I}} - \mathbf{x}_t^{p\mathcal{I}}) + \mathbf{B}_t^{\mathcal{I}}(\mathbf{u}_t^{\mathcal{I}} - \mathbf{u}_t^{p\mathcal{I}}) + \mathbf{G}_t^{\mathcal{I}}\mathbf{w}_t^{\mathcal{I}} + O(\delta), \\ \mathbf{z}_t^{\mathcal{I}} =& \mathbf{z}_t^{p\mathcal{I}} + \mathbf{H}_t^{\mathcal{I}}(\mathbf{x}_t^{\mathcal{I}} - \mathbf{x}_t^{p\mathcal{I}}) + \mathbf{M}_t^{\mathcal{I}}\mathbf{v}_t^{\mathcal{I}} + O(\delta), \\ J_{\boldsymbol{\pi}} =& J^p + \tilde{J}_1 + O(\delta), \end{aligned}$$

$$\tilde{J}_1 := \sum_{t=0}^{K_{\mathcal{I}}-1} [\mathbf{C}_t^{\mathbf{x}_{\mathcal{I}}}(\mathbf{x}_t^{\mathcal{I}} - \mathbf{x}_t^{p_{\mathcal{I}}}) + \mathbf{C}_t^{\mathbf{u}_{\mathcal{I}}}(\mathbf{u}_t^{\mathcal{I}} - \mathbf{u}_t^{p_{\mathcal{I}}})] + \mathbf{C}_{K_{\mathcal{I}}}^{\mathbf{x}_{\mathcal{I}}}(\mathbf{x}_{K_{\mathcal{I}}}^{\mathcal{I}} - \mathbf{x}_{K_{\mathcal{I}}}^{p_{\mathcal{I}}}),$$

$$J^p := \sum_{t=0}^{K_{\mathcal{I}}-1} c_t(\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}) + c_{K_{\mathcal{I}}}(\mathbf{x}_{K_{\mathcal{I}}}^{p_{\mathcal{I}}}).$$

The Jacobians are:

$$\mathbf{A}_t^{\mathcal{I}} := \nabla_{\mathbf{x}} \mathbf{f}^{\mathcal{I}}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}}, \mathbf{B}_t^{\mathcal{I}} := \nabla_{\mathbf{u}} \mathbf{f}^{\mathcal{I}}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}}, \mathbf{G}_t^{\mathcal{I}} := \epsilon \boldsymbol{\sigma}^{\mathbf{f}_{\mathcal{I}}}(t),$$

$$\mathbf{H}_t^{\mathcal{I}} := \nabla_{\mathbf{u}} \mathbf{h}(\mathbf{x}^{\mathcal{I}})|_{\mathbf{x}_t^{p_{\mathcal{I}}}}, \mathbf{M}_t^{\mathcal{I}} := \epsilon \boldsymbol{\sigma}^{\mathbf{h}_{\mathcal{I}}}(t),$$

$$\mathbf{C}_t^{\mathbf{x}_{\mathcal{I}}} := \nabla_{\mathbf{x}} c_t^{\boldsymbol{\pi}^{\mathcal{I}}}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}}, \mathbf{C}_{K_{\mathcal{I}}}^{\mathbf{x}_{\mathcal{I}}} := \nabla_{\mathbf{x}} c_{K_{\mathcal{I}}}^{\boldsymbol{\pi}^{\mathcal{I}}}(\mathbf{x})|_{\mathbf{x}_t^{p_{\mathcal{I}}}}, \mathbf{C}_t^{\mathbf{u}_{\mathcal{I}}} := \nabla_{\mathbf{u}} c_t^{\boldsymbol{\pi}^{\mathcal{I}}}(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}}.$$

$\mathbf{u}_t^{\mathcal{I}}$ is obtained by deigning a optimal LQR feedback policy for the mean of the joint estimate to track the joint nominal trajectory $\mathbf{x}_t^{p_{\mathcal{I}}}$ as:

$$\mathbf{u}_t^{\mathcal{I}} = \mathbf{u}_t^{p_{\mathcal{I}}} - \mathbf{L}_t^{\mathcal{I}}(\hat{\mathbf{x}}_t^{\mathcal{I}} - \mathbf{x}_t^{p_{\mathcal{I}}}), \tag{7.4}$$

$$\mathbf{L}_t^{\mathcal{I}} = (\mathbf{W}_t^{u_{\mathcal{I}}} + (\mathbf{B}_t^{\mathcal{I}})^T \mathbf{S}_t^{\mathcal{I}} \mathbf{B}_t^{\mathcal{I}})^{-1} (\mathbf{B}_t^{\mathcal{I}})^T \mathbf{S}_t^{\mathcal{I}} \mathbf{A}_t^{\mathcal{I}}. \tag{7.5}$$

$\mathbf{S}_t^{\mathcal{I}}$ is obtained using the backward dynamic Riccati equation with $\mathbf{S}_{K_{\mathcal{I}}}^{\mathcal{I}} = \mathbf{W}_{K_{\mathcal{I}}}^{x_{\mathcal{I}}}$:

$$\mathbf{S}_{t-1}^{\mathcal{I}} = (\mathbf{A}_t^{\mathcal{I}})^T \mathbf{S}_t^{\mathcal{I}} \mathbf{A}_t^{\mathcal{I}} - (\mathbf{A}_t^{\mathcal{I}})^T \mathbf{S}_t^{\mathcal{I}} \mathbf{B}_t^{\mathcal{I}} \mathbf{L}_t^{\mathcal{I}} + \mathbf{W}_t^{x_{\mathcal{I}}}, \tag{7.6}$$

where $\mathbf{W}_t^{x_{\mathcal{I}}} \succeq 0$ and $\mathbf{W}_t^{u_{\mathcal{I}}} \succeq 0$ are two block-diagonal positive semi-definite weight matrices (with blocks of $\mathbf{W}_t^{x_i} \succeq 0$ of dimension $n_x^i \times n_x^i$ and $\mathbf{W}_t^{u_i} \succeq 0$ of dimension $n_u^i \times n_u^i$, respectively). Moreover, the mean of the estimate is obtained using the KF equations, whose error evolution defined as $\mathbf{e}_t^{\mathcal{I}} := \hat{\mathbf{x}}_t^{\mathcal{I}} - \mathbf{x}_t^{p_{\mathcal{I}}}$ is:

$$\mathbf{e}_{t+1}^{\mathcal{I}} = \mathbf{T}_t^{\mathbf{e}_{\mathcal{I}}}(\mathbf{e}_t^{\mathcal{I}}) + \mathbf{T}_t^{\mathbf{u}_{\mathcal{I}}}(\mathbf{u}_t^{\mathcal{I}} - \mathbf{u}_t^{p_{\mathcal{I}}}) + \mathbf{T}_t^{\mathbf{z}_{\mathcal{I}}}(\mathbf{z}_{t+1}^{\mathcal{I}} - \mathbf{z}_{t+1}^{p_{\mathcal{I}}}). \tag{7.7}$$

*Structure of feedback:* As shown above, $\mathbf{L}^{\mathcal{I}}_t$ that is designed using the single-agent decoupling principle depends on the entire mean of the state estimation. Next, we will analyze the structure of feedback and prove the multi-agent decoupling principle for the partially-observed case.

*Remark:* Once again, we have only shown the first-order linearizations above, it should be noted that the second-order linearizations also follows similarly with proper indexing of the single-agent variables. Therefore, we avoid repeating them. Nevertheless, for the rest of the proof only first-order variables suffice.

## 7.2 Decoupling of Feedback and Estimator Designs

**Proof 26 (proof of Result 4)** *This result has two parts. The first part is the decoupling of feedback gain designs, and the second part is the decoupling of the estimator designs. Both of the results are proven similarly to the proof of Result 2. Because of the separation of controller and estimator designs, the controller equations are the same as in* (7.4) *and* (7.6)*, except that the controller is designed to compensate for the estimator error rather than the state error. Therefore, the rest of the proof for controller design is the same as for Result 2. The second part is the decoupling of estimator implementations which once again follows from the independence of the dynamics and the fact that the estimation equation follows a similar algebraic equations. In particular, the mean update is given by* (7.7) *where the Kalman gain is obtained using the joint covariance update given by the following forward dynamic Riccati equation with* $\mathbf{P}^{\mathcal{I}}_0 = \epsilon^2 \mathbf{\Sigma}_{\mathbf{x}^{\mathcal{I}}_0}$ *and:*

$$\begin{aligned}
\bar{\mathbf{P}}^{\mathcal{I}}_t &:= \mathbf{A}^{\mathcal{I}}_{t-1}\mathbf{P}^{\mathcal{I}}_{t-1}(\mathbf{A}^{\mathcal{I}}_{t-1})^T + \mathbf{G}^{\mathcal{I}}_{t-1}\mathbf{\Sigma}_{\mathbf{w}}(\mathbf{G}^{\mathcal{I}}_{t-1})^T \\
\mathbf{K}^{\mathcal{I}}_t &:= \bar{\mathbf{P}}^{\mathcal{I}}_t(\mathbf{H}^{\mathcal{I}}_t)^T(\mathbf{H}^{\mathcal{I}}_t\bar{\mathbf{P}}^{\mathcal{I}}_t(\mathbf{H}^{\mathcal{I}}_t)^T+\mathbf{M}^{\mathcal{I}}_t\mathbf{\Sigma}_{\mathbf{v}}(\mathbf{M}^{i}_t)^T)^{-1} \\
\mathbf{P}^{\mathcal{I}}_t &= (\mathbf{I} - \mathbf{K}^{\mathcal{I}}_t\mathbf{H}^{\mathcal{I}}_t)\bar{\mathbf{P}}^{\mathcal{I}}_t, \qquad (7.8)
\end{aligned}$$

*which deterministically depends on the nominal trajectory. Both the mean and covariance equations separate into $m$ non-interacting equations following the same reasoning as stated for the controller feedback gain. Hence, the estimator of each agent can be implemented separately from the other agents without the need to have the knowledge of their current estimation information. Therefore, the feedback policy for each agent (which depends on the state estimate of that agent) can also be fully decoupled into $m$ non-interacting feedback policies, as long as the conditions of Result 3 are met.*

**Remark 5** *Result 4 proves that under the conditions of Result 3 and Theorem 7, and the independence of the dynamics, the feedback gain designs and estimator implementation of the agent $i$ can be optimally calculated separately from the agent $j \neq i$. It states that the joint forward Riccati equation of covariance updates (with index $\mathcal{I}$) breaks up into $m$ separate Riccati equations. As a result, the dimension of the optimal linear feedback gain for agent $i$ reduces to $n_u^i \times n_x^i$, which is the same as an LQR design to track the fully-observed nominal state of agent $i$. Moreover, the Riccati equations for estimation can also be factored out, which leads to a separate marginal belief evolution implementation of the Kalman filters for different agents. This design leads to a decentralized multi-agent planning approach of MT-LQG, which is near-second-order optimal as $\epsilon \downarrow 0$.*

**Remark 6** *Note that once again, the shared cost such as inter-agent collision is taken into account in the nominal trajectory design stage with sufficient safety margins such that within the $\delta$ tubes of the agents, the shared cost vanishes to zero. Therefore, the feedback design for each agent becomes the LQG tracking problem within a tube without considering the shared cost. This is addressed in more details in Chapter 9.*

## 7.3 MT-LQG: Multi-agent Trajectory-optimized LQG

The design approach resulting from the combination of Results 3 and 4 for a multi-agent system with imperfect state information consists of two steps. The first step is to solve the joint nominal trajectory design problem. The second step is to design $m$ LQG trackers one for each of the agents, separately.

**Problem 13 (MT-LQG Nominal Trajectory Design Problem)** *Given an initial joint mean* $\bar{\mathbf{x}}_0^{\mathcal{I}} =: \mathbf{x}_0^{p_{\mathcal{I}}}$*, solve:*

$$\min_{\mathbf{u}^{p_{\mathcal{I}}}_{0:K_{\mathcal{I}}-1}} \mathbb{E}[\sum_{t=0}^{K_{\mathcal{I}}-1} c_t(\mathbf{x}_t^{p_{\mathcal{I}}}, \mathbf{u}_t^{p_{\mathcal{I}}}) + c_{K_{\mathcal{I}}}(\mathbf{x}_{K_{\mathcal{I}}}^{p_{\mathcal{I}}})]$$

$$s.t.\ \mathbf{x}_{t+1}^{p_i} = \mathbf{f}^i(\mathbf{x}_t^{p_i}, \mathbf{u}_t^{p_i}),\ \ i \in \mathcal{I} \tag{7.9a}$$

$$\mathbf{z}_{t+1}^{p_i} = \mathbf{h}^i(\mathbf{x}_{t+1}^{p_i}),\ \ i \in \mathcal{I}. \tag{7.9b}$$

*Control policy:* After solving Problem (13), the control policy is an LQG policy designed for agent $i$ applied on the estimation error $(\hat{\mathbf{x}}_t^i - \mathbf{x}_t^{p_i})$ as:

$$\mathbf{u}_t^i = \mathbf{u}_t^{p_i} - \mathbf{L}_t^i(\hat{\mathbf{x}}_t^i - \mathbf{x}_t^{p_i}), \tag{7.10}$$

$$\mathbf{L}_t^i = (\mathbf{W}_t^{u_i} + (\mathbf{B}_t^i)^T \mathbf{S}_t^i \mathbf{B}_t^i)^{-1} (\mathbf{B}_t^i)^T \mathbf{S}_t^i \mathbf{A}_t^i, \tag{7.11}$$

where the Jacobians are

$$\mathbf{A}_t^i := \nabla_{\mathbf{x}} \mathbf{f}^i(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_i}, \mathbf{u}_t^{p_i}}, \mathbf{B}_t^i := \nabla_{\mathbf{u}} \mathbf{f}^i(\mathbf{x}, \mathbf{u})|_{\mathbf{x}_t^{p_i}, \mathbf{u}_t^{p_i}}, \mathbf{G}_t^i := \epsilon \boldsymbol{\sigma}^{\mathbf{f}_i}(t),$$

$$\mathbf{H}_t^i := \nabla_{\mathbf{u}} \mathbf{h}(\mathbf{x}^i)|_{\mathbf{x}_t^{p_i}}, \mathbf{M}_t^i := \epsilon \boldsymbol{\sigma}^{\mathbf{h}_i}(t),$$

and $\mathbf{S}_t^i$ is obtained using a single-agent backward dynamic Riccati equation with $\mathbf{S}_{K_i}^i = \mathbf{W}_{K_i}^{x_i}$:

$$\mathbf{S}_{t-1}^i = (\mathbf{A}_t^i)^T \mathbf{S}_t^i \mathbf{A}_t^i - (\mathbf{A}_t^i)^T \mathbf{S}_t^i \mathbf{B}_t^i \mathbf{L}_t^i + \mathbf{W}_t^{x_i}. \tag{7.12}$$

Moreover, the mean of the estimate is obtained using the KF equations, whose error evolution defined as $\tilde{\mathbf{e}}_t^i := \hat{\mathbf{x}}_t^i - \mathbf{x}_t^{p_i}$ is:

$$\tilde{\mathbf{e}}_{t+1}^i = \mathbf{T}_t^{\mathbf{e}_i}(\tilde{\mathbf{e}}_t^i) + \mathbf{T}_t^{\mathbf{u}_i}(\mathbf{u}_t^i - \mathbf{u}_t^{p_i}) + \mathbf{T}_t^{\mathbf{z}_i}(\mathbf{z}_{t+1}^i - \mathbf{z}_{t+1}^{p_i}). \tag{7.13}$$

# II. BELIEF SPACE PLANNING

# 8. BELIEF-SPACE PLANNING FOR SINGLE-AGENT SYSTEM

In this chapter, we use the theory of the previous sections, particularly, the T-LQG approach, to solve robotic trajectory planning problems. We consider single-agent planning under process and measurement uncertainties. As mentioned before, this requires the solution of a stochastic control problem in the space of feedback policies. Also formulated as a POMDP problem, this problem is referred to as the belief space planning problem in the literature [14], as well. In this Dissertation, we reserve the “belief” keyword to refer to the conditional distribution of the system (or its approximation) when the distribution is Gaussian. For more general situations, we will refer to the conditional distribution as the information state.

In this chapter, we define a special cost function that is suited for belief space planning and utilize the T-LQG algorithm and the Decoupling Principle of the past chapters to tackle the belief space planning problem. Using the T-LQG method, by restricting the policy class to the linear feedback polices, we reduce the general $(n^2 + n)$-dimensional belief space planning problem to an $n$-dimensional problem. As opposed to the previous literature that searches in the space of open-loop optimal control policies, we obtain this reduction in the space of closed-loop policies by obtaining a Linear Quadratic Gaussian (LQG) design with the best nominal performance. Then, by taking the entire underlying trajectory of the LQG controller as the decision variable, we pose a coupled design of the trajectory and estimator (while keeping the design of the controller separate) as a NonLinear Program (NLP) that can be solved by a general NLP solver. Our algorithm's validity is based on the theory proven in the previous chapters. We provide an analysis on the existing major belief space planning methods and show that our algorithm maintains a low computational burden while searching in the policy space. Finally, we extend our solution to contain general state and control constraints. Our simulation results support our design.

## 8.1 Introduction

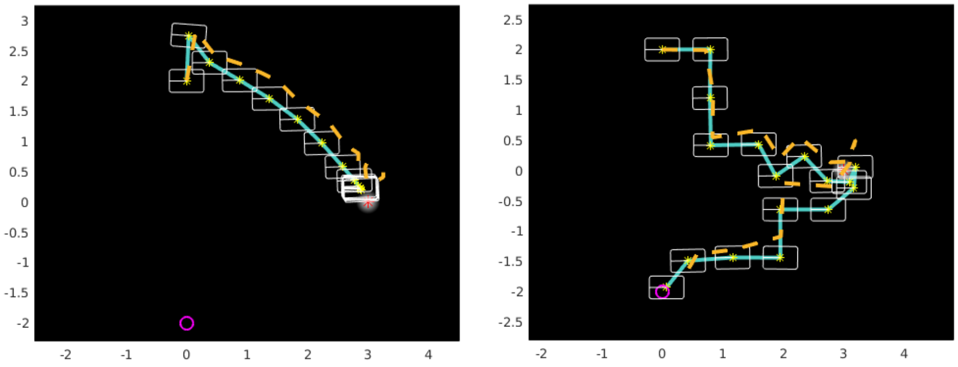
The Linear Quadratic Gaussian (LQG) methodology provides the optimal estimator and controller for linear systems with Gaussian noises [159]. However, an LQG planner requires a nominal trajectory to begin with. Therefore, the problem consists of three elements including the nominal trajectory, the estimator, and the control law. One approach separately designs the trajectory from the LQG policy (estimator plus controller) by providing finite number of different a priori (RRT-based) trajectories and comparing the LQG performance over each one [59]. Another approach performs an alternating iterative process of designing the policy and the trajectory with the other fixed [35, 181], reaching to a high-dimensional belief controllers. An approach to a coupled design of trajectory and the policy in nonlinear systems utilizing the Extended Kalman Filter (EKF) is based on the heuristic assumption of Most-Likely Observations (MLO) during planning [57, 58]. Another class of POMDP solvers [11] utilize a Monte-Carlo representation of beliefs [61, 62]. The state-of-the-art POMDP solvers are posed on decision trees, reducing the search space to a finite set of reachable belief nodes given an initial belief. However, this 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.

We overcome this hurdle by utilizing the decoupling principle using which we near-optimally decouple the design of the nominal trajectory and feedback policy. As mentioned in Chapter 6, there exists an exactly linear system ($l$-system) which provides a linear Gaussian surrogate representation of the original nonlinear non-

Gaussian system and is always within some $O(\epsilon^{2-\gamma})$ of the original system for $0 < \gamma \ll 1$ for small noise. We refer to the conditional distribution of the $l$-system as the belief of the system, which is always Gaussian. Then, we utilize the T-LQG approach and define a special cost function that aims for the best estimation performance and utilize the properties of the $l$-system to design a best nominal trajectory for the original system. In particular, we utilize the estimation covariance of the $l$-system provided by the Kalman filter as an approximation to the original system's estimation performance and use the fact that for a linear Gaussian system, the covariance evolution is deterministic once the underlying trajectory of the system is fixed. The trace of this covariance evolution becomes part of the nominal trajectory design in the T-LQG approach.

Therefore, we provide a coupled design of trajectory and estimator aiming for the best estimation performance using the underlying trajectory of the LQG controller as the optimization variable, while keeping the design of controller separate from the design of the trajectory and the estimator. This simplifies the belief space planning to an optimization problem that can be solved by a general NonLinear Programming (NLP) solver aimed at the design of the nominal trajectory with the best nominal estimation performance meanwhile incorporating the cost of the control effort. One can intuitively interpret this as that, we use the decoupling principle for nonlinear systems and the control theory separation principle for linear systems in addition to the structure of the LQG method to pose an optimization problem on sequence of control actions parameterizing LQG polices to reach a quantifiably near-optimal policy, rather than optimizing over the general policy space using the DP equation. This method reduces the dimension of the underlying state in the belief space planning optimization problem from $n+n^2$ (Gaussian belief dimension) to $n$ (state dimension), reducing the computational burden significantly. 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 Gaussian belief space planning method in the space of feedback policies. It is worth mentioning that performing a feedback design in the $(n+n^2)$-dimensional belief space, results in the computational burden of $O(Kn^6)$ as in [35].

As mentioned before, over a given nominal trajectory, the nominal performance of the estimator (using the $l$-system) is deterministically given by the dynamic Riccati equations independent from the actual observations and the controller form. Therefore, the trajectory planning 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 trajec-



(a) RHC-based results after $K = 16$ steps

(b) T-LQG results after one plan and execution

Figure 8.1: Comparison of T-LQG and an MLO plus RHC-based method [58]. In each figure, the dashed line shows the ground truth trajectory, and the solid line shows the state estimate trajectory. A purple circle denotes the target, and the white region shows the landmark for a range and bearing observation model. a) In the RHC-based method, re-planning is triggered at every step. However, these methods for *stochastic* systems fail to reach the goal after $K$ steps, and require heuristic adjustments to work; b) however, in T-LQG, for this example, planning only happens once, and the resulting feedback policy is executed for the entire horizon, reaching the goal state after $K$ steps.

tory. The key observation on the belief space planning problem from the decoupling principle is the following: fixing the feedback policy as a linear policy and designing an LQG policy for a linearized system around a given nominal trajectory provides a solution of a near-optimal estimation and control performance along that specific trajectory.

For a fixed linearization around a 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, a near-optimal estimator and controller to follow that trajectory. This method provides a theoretically coherent planning approach while providing a low computational burden. Other methods such as the MLO method of [57, 58] while also solving trajectory optimization problems on covariance performance, provide no guarantees for their design and most importantly provide control policies that are either high-dimensional (such as Belief-LQR) or computationally expensive (such as Receding Horizon Control (RHC)). Although [57] also provides simple LQG policy as one of the controllers, the paper also suggests utilizing Belief-LQR similar to [35].

The decoupling principle proves that the decoupled design of policy and the trajectory is only possible when there is an assumed existence of the control law in the loop from the beginning to keep the state around the nominal trajectory. Otherwise, the state deviation from the nominal trajectory keeps growing and the validity region of the nominal (linearization) trajectory (of control and subsequent state and observations) collapse, reducing the approach to a heuristic design that requires replanning at every time step as in [58, 182] (see Fig. 8.1) or requiring high-dimensional belief planners as in [57]. In contrast, in our approach, the low-dimensional controller keeps the state around the nominal trajectory, and therefore, the nominal estimation performance remains valid, thereby obviating the need for constant replanning. Also, using a high-dimensional controller such as a belief-LQR over an $(n+n^2)$-dimensional space as in [57] or [35] and [181] entails disregarding the separation principle by coupling the controller design with the design of the estimator. Using the decoupling and separation principles also enables us to pose the problem as a standard NLP in $n$-dimensional space, rather than using a dynamic programming mechanism in a local linearization region (which involves calculations in an $(n+n^2)$-dimensional space to solve the coupled equations of the belief estimation and the controller design, as in [35, 181]). Moreover, since the DP is only solved locally it lacks the optimality properties of the global DP which is performed using the original nonlinear equations of the system over the entire domain of the problem, reducing the mentioned approaches to second-order optimization problems in a local region of the problem.

Finally, when the accumulated linearization error (or other errors) increases above a tolerable threshold during the execution, replanning can be triggered. This is also a 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 use of of-the-shelf state-of-the-art optimization software and tools.

Unlike point-based POMDP solvers [96, 95, 26], in T-LQG the time-horizon is a linear factor in the computational complexity, rather than a factor in the exponent, viz. the curse of history. This means that T-LQG is capable of solving belief space problems on a considerably larger scale. Indeed, current point-based solvers cannot scale to the continuous state, action and observation space problems that are considered here.

## 8.2 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. The problem definition is the same as in Chapter 6, and we avoid repeating it.

*Belief:* The conditional distribution of $\mathbf{x}_t$ given the data history up to time $t$, is called the information state. While for a nonlinear system with additive Gaussian perturbations, the informations state is non-Gaussian, we have shown in the previous chapters that under small noise assumption, a carefully constructed linear Gaussian system can be used a surrogate system for control and estimation of the original non-Gaussian system, in a near-optimal fashion. For this linear Gaussian system, the conditional distribution is also Gaussian. We refer to this Gaussian distribution as the belief, and will denote it by $\mathbf{b}_t := ((\hat{\mathbf{x}}_t)^T, \text{vec}(\mathbf{P}_t)^T)$, a vector comprised of the mean and covariance of the conditional distribution of the linear Gaussian surrogate system, $l$-system. The update equation for the belief follows a Kalman filter. Then, we will define the entire problem in terms of the belief, and refer to it as the belief space planning.

*Assumptions:* We assume that the underlying system is a mechanical system. Hence, the actuators have saturation constraints and this causes the control effort at each time step to be bounded. We also assume that the covariance of the estimation is finite, therefore, the expected state deviation also becomes bounded. Last, we assume that problem is also finite horizon.

## 8.3 Belief Space Planning Method: T-LQG

We provide details of our design for the planning problem.

*Definition of the cost:* We consider a quadratic cost in terms of the deviation of the state rather than the state as well as the control effort:

$$\mathbb{E}[J] := \mathbb{E}[\sum_{t=0}^{K-1} \mathbf{c}_t(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{c}_K(\mathbf{x}_K)], \tag{8.1}$$

where

$$\mathbf{c}_t(\mathbf{x}_t, \mathbf{u}_t) := \tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t + \mathbf{u}_t^T \mathbf{W}_t^u \mathbf{u}_t, \tag{8.2a}$$

$$\mathbf{c}_K(\mathbf{x}_K) := \tilde{\mathbf{x}}_K^T \mathbf{W}_K^x \tilde{\mathbf{x}}_K, \tag{8.2b}$$

where $\mathbf{W}_t^x, \mathbf{W}_t^u \succeq 0$ are two positive-definite weight matrices, and $\mathbf{W}_t^x$ is symmetric, thereby, it has a square root. Moreover, we choose the weight matrices such that $|\mathbf{W}_t^u| \ll |\mathbf{W}_t^x|$, i.e., the magnitude of the weight matrices for the control effort is chosen smaller than that of the weight matrix for the state deviation. The reason behind this is that the second term in (8.2a) is the control effort itself and its magnitude is in the same order of the state cost. That is, because the controller that we use is LQG, then $\tilde{\mathbf{u}}_t = -\mathbf{L}_t \tilde{\mathbf{x}}_t$ and therefore, $O(|\tilde{\mathbf{u}}_t|) = O(|\tilde{\mathbf{x}}_t|)$ which was also shown in Chapter 6. On the other hand, $O(|\mathbf{u}_t|) = O(|\mathbf{x}_t|)$ and $O(|\tilde{\mathbf{x}}_t|) \ll O(|\mathbf{x}_t|) = O(|\mathbf{u}_t|)$. Therefore, choosing the magnitude of the weight matrices in the same order would cause the first term to be completely dominated by the second term. As a result of our choice of the weight matrices, $O(|\mathbf{u}_{t-1}^T \mathbf{W}_t^u \mathbf{u}_{t-1}|) = O(|\tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t|)$. Therefore, $O(|\tilde{\mathbf{u}}_{t-1}^T \mathbf{W}_t^u \tilde{\mathbf{u}}_{t-1}|) \ll O(|\tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t|)$. Note that one might even want to choose weights such that $|\mathbf{u}_{t-1}^T \mathbf{W}_t^u \mathbf{u}_{t-1}| < |\tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t|$ to emphasize the cost of deviation rather than effort. That can be the design choice of the engineer as long as $|\mathbf{W}_t^u| \ll |\mathbf{W}_t^x|$ for the above cost function to be meaningful. One might also ask the reason behind the choosing the cost of deviation of the state rather than the state. This is mainly due to the fact that the deviation of the state is related to the estimation covariance and we address this next.

First, note that the first term of the cost function is quadratic in $\tilde{\mathbf{x}}_t$. Therefore, the second-order expansion of this term around the nominal trajectory is itself. This means that the nominal and the first order terms of this function are zero. Hence, the expansion of the cost function to the second order is as follows:

$$J = J^p + \tilde{J}_1 + \tilde{J}_2, \tag{8.3}$$

where we have:

- $J^p := \sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p$ denotes the nominal cost;
- $\tilde{J}_1 := \sum_{t=0}^{K-1} 2\mathbf{W}_t^u \tilde{\mathbf{u}}_t$ is the first order cost error;
- $\tilde{J}_2 := \sum_{t=0}^{K-1} (\tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t + \tilde{\mathbf{u}}_t^T \mathbf{W}_t^u \tilde{\mathbf{u}}_t) + \tilde{\mathbf{x}}_K^T \mathbf{W}_K^x \tilde{\mathbf{x}}_K$ is the second order cost error, where we have used the fact that $c_t \in \mathbb{C}^2$;
- $J_2 := J^p + \tilde{J}_1 + \tilde{J}_2$ is the second order approximation of the cost function.

Note that since the cost function is quadratic, there is no $o(\cdot)$ terms in (8.3) and the expansion is exact. Therefore, the cost function can be written as

$$J = \sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p + \sum_{t=0}^{K-1} 2\mathbf{W}_t^u \tilde{\mathbf{u}}_t + \sum_{t=0}^{K-1} (\tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t + \tilde{\mathbf{u}}_t^T \mathbf{W}_t^u \tilde{\mathbf{u}}_t) + \tilde{\mathbf{x}}_K^T \mathbf{W}_K^x \tilde{\mathbf{x}}_K. \tag{8.4}$$

We next show that after taking expectation in the above formula, the only terms that are dominating are the following terms:

$$\sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p + \sum_{t=0}^{K} \mathbb{E}[\tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t].$$

First, note that in Chapter 6, we proved that for the T-LQG policy using (6.76), the cost function can be written as follows (with probability $P(\Omega(\epsilon^{2-\gamma}))$):

$$\begin{aligned}
J =& J^p + (\sum_{t=0}^{K} \mathbf{C}^{\mathbf{L}}_t \mathbf{U}^{\mathbf{x}_0}_{t-1})\tilde{\mathbf{x}}_0 + \sum_{t=0}^{K}(\sum_{s=t}^{K-1} \mathbf{C}^{\mathbf{L}}_s \mathbf{V}^{\mathbf{v}}_{t,s-1} - \mathbf{C}^{\mathbf{u}}_s \mathbf{L}_{t,s}\mathbf{M}_t)\mathbf{v}_t + \sum_{t=0}^{K}(\sum_{s=t}^{K-1} \mathbf{C}^{\mathbf{L}}_s \mathbf{W}^{\mathbf{w}}_{t,s-1})\mathbf{w}_t \\
&+ \sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T \mathbf{H}^{(\mathbf{h},f,\mathbf{h})^{kij}}_{t-s-1}\tilde{\mathbf{x}}_j + \tilde{\mathbf{x}}_i^T \mathbf{H}^{(\mathbf{h},f,\mathbf{v})^{kij}}_{t-s-1}\mathbf{v}_j + \mathbf{v}_i^T \mathbf{H}^{(\mathbf{v},f,\mathbf{v})^{kij}}_{t-s-1}\mathbf{v}_j)\mathbf{C}^{\mathbf{L}}_t \mathbf{Q}_s \mathbf{e}^{n_x}_k \\
&- \sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{r=0}^{t-s-1}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T \mathbf{H}^{h^j}_r \tilde{\mathbf{x}}_r)\mathbf{C}^{\mathbf{L}}_t \mathbf{Q}_s \mathbf{B}_{t-s-1}\mathbf{L}_{r,t-s-1}\mathbf{e}^{n_z}_j \\
&- \sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T \mathbf{H}^{h^j}_s \tilde{\mathbf{x}}_s)\mathbf{C}^{\mathbf{u}}_t \mathbf{L}_{s,t}\mathbf{e}^{n_z}_j + \sum_{t=0}^{K}\tilde{\mathbf{x}}_t^T \mathbf{C}^{\mathbf{LL}}_t \tilde{\mathbf{x}}_t + O(\epsilon^{2-\gamma}),
\end{aligned} \tag{8.5}$$

where $\mathbf{C}^{\mathbf{u}}_t = \mathbf{W}^u_t$, $\mathbf{C}^{\mathbf{L}}_t = -\sum_{s=t}^{K-1} 2\mathbf{W}^u_s \mathbf{L}_{t,s}\mathbf{H}_t, 0 \leq t \leq K-1$, $\mathbf{C}^{\mathbf{L}}_K = 0$, $\mathbf{C}^{\mathbf{LL}}_t = \mathbf{W}^x_t + \mathbf{L}_t^T \mathbf{W}^u_t \mathbf{L}_t, 0 \leq t \leq K-1$, and $\mathbf{C}^{\mathbf{LL}}_K = \mathbf{W}^x_K$. Now, based on the assumptions, the cost function is bounded. Therefore, after taking the expectation and using the calculations of the proof of Theorem 10, we have:

$$\begin{aligned}
\mathbb{E}[J] =& J^p + 0 + \\
&\mathbb{E}[\sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{k=1}^{n_x}\sum_{i=0}^{t-s-1}\sum_{j=0}^{t-s-1}(\tilde{\mathbf{x}}_i^T \mathbf{H}^{(\mathbf{h},f,\mathbf{h})^{kij}}_{t-s-1}\tilde{\mathbf{x}}_j + \tilde{\mathbf{x}}_i^T \mathbf{H}^{(\mathbf{h},f,\mathbf{v})^{kij}}_{t-s-1}\mathbf{v}_j + \mathbf{v}_i^T \mathbf{H}^{(\mathbf{v},f,\mathbf{v})^{kij}}_{t-s-1}\mathbf{v}_j)\mathbf{C}^{\mathbf{L}}_t \mathbf{Q}_s \mathbf{e}^{n_x}_k \\
&- \sum_{t=0}^{K}\sum_{s=0}^{t-1}\sum_{r=0}^{t-s-1}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_r^T \mathbf{H}^{h^j}_r \tilde{\mathbf{x}}_r)\mathbf{C}^{\mathbf{L}}_t \mathbf{Q}_s \mathbf{B}_{t-s-1}\mathbf{L}_{r,t-s-1}\mathbf{e}^{n_z}_j \\
&- \sum_{t=0}^{K-1}\sum_{s=0}^{t}\sum_{j=1}^{n_z}(\tilde{\mathbf{x}}_s^T \mathbf{H}^{h^j}_s \tilde{\mathbf{x}}_s)\mathbf{C}^{\mathbf{u}}_t \mathbf{L}_{s,t}\mathbf{e}^{n_z}_j \\
&+ \sum_{t=0}^{K-1}\tilde{\mathbf{x}}_t^T \mathbf{L}_t^T \mathbf{W}^u_t \mathbf{L}_t \tilde{\mathbf{x}}_t + \sum_{t=0}^{K}\tilde{\mathbf{x}}_t^T \mathbf{W}^x_t \tilde{\mathbf{x}}_t] + O(\epsilon^{2-\gamma}),
\end{aligned} \tag{8.6}$$

Now, note that except for $J^p$, the rest of the terms are all quadratic in terms of $\tilde{\mathbf{x}}$. Moreover, except for the last term, all the other terms are weighted in $\mathbf{W}^u_t$. Now, using the assumption that $|\mathbf{W}_t^u| \ll |\mathbf{W}_t^x|$, we approximate the above expression as:

$$\begin{aligned}\mathbb{E}[J] =& J^p + \sum_{t=0}^{K} \mathbb{E}[\tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t] + O(\epsilon^{2-\gamma}), && (8.7a)\\ =& \sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p + \sum_{t=0}^{K} \mathbb{E}[\tilde{\mathbf{x}}_t^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t] + O(\epsilon^{2-\gamma}), && (8.7b)\end{aligned}$$

where we still have kept the equality since the terms that we have ignored are in the order of $O(\epsilon^{2-\gamma})$. They only change the pre-constant; however, due to the fact that $|\mathbf{W}_t^u| \ll |\mathbf{W}_t^x|$, the only dominant term in the pre-constant of the error is the one associated with $\mathbf{W}_t^x$.

*Approximating the cost function:* Next, we use the fact that with probability $P(\Omega(\epsilon^{2-\gamma}))$, we have (based on (6.48b)):

$$\mathbf{x}_{t+1} = \mathbf{x}_{t+1}^l + O(\epsilon^{2-\gamma}).$$

Therefore with the same probability, we have

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_{t+1}^l + O(\epsilon^{2-\gamma}).$$

Replacing the above expression in (8.7b), and once again using the fact that with probability $1 - P(\Omega(\epsilon^{2-\gamma}))$, the cost is bounded, we have

$$\begin{aligned}\mathbb{E}[J] &= \sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p + \sum_{t=0}^{K} \mathbb{E}[(\tilde{\mathbf{x}}_t^l)^T \mathbf{W}_t^x \tilde{\mathbf{x}}_t^l] + O(\epsilon^{2-\gamma}) && (8.8a)\\ &= \sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p + \sum_{t=0}^{K} \mathbb{E}[(\tilde{\mathbf{x}}_t^l)^T \mathbf{W}_t^T \mathbf{W}_t \tilde{\mathbf{x}}_t^l] + O(\epsilon^{2-\gamma}) && (8.8b)\\ &= \sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p + \sum_{t=0}^{K} \mathbb{E}[(\mathbf{W}_t \tilde{\mathbf{x}}_t^l)^T \mathbf{W}_t \tilde{\mathbf{x}}_t^l] + O(\epsilon^{2-\gamma}) && (8.8c)\end{aligned}$$

$$= \sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p + \sum_{t=0}^{K} \mathbb{E}[\mathrm{tr}[(\mathbf{W}_t \tilde{\mathbf{x}}_t^l)(\mathbf{W}_t \tilde{\mathbf{x}}_t^l)^T]] + O(\epsilon^{2-\gamma}) \tag{8.8d}$$

$$= \sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p + \sum_{t=0}^{K} \mathbb{E}[\mathrm{tr}[\mathbf{W}_t \tilde{\mathbf{x}}_t^l (\tilde{\mathbf{x}}_t^l)^T \mathbf{W}_t^T]] + O(\epsilon^{2-\gamma}) \tag{8.8e}$$

$$= \sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p + \sum_{t=0}^{K} \mathrm{tr}[\mathbf{W}_t \mathbb{E}[\tilde{\mathbf{x}}_t^l (\tilde{\mathbf{x}}_t^l)^T] \mathbf{W}_t^T] + O(\epsilon^{2-\gamma}) \tag{8.8f}$$

$$= \sum_{t=0}^{K-1} (\mathbf{u}_t^p)^T \mathbf{W}_t^u \mathbf{u}_t^p + \sum_{t=0}^{K} \mathrm{tr}[\mathbf{W}_t \mathbf{P}_t^l \mathbf{W}_t^T] + O(\epsilon^{2-\gamma}), \tag{8.8g}$$

where $\mathbf{P}_t^l$ was defined in (6.28) as is repeated below:

$$\hat{\tilde{\mathbf{x}}}_{t+1}^l = \mathbf{A}_t \hat{\tilde{\mathbf{x}}}_t^l + \mathbf{B}_t \tilde{\mathbf{u}}_t^l + \mathbf{K}_{t+1}(\tilde{\mathbf{z}}_{t+1}^l - \mathbf{H}_{t+1}(\mathbf{A}_t \hat{\tilde{\mathbf{x}}}_t^l + \mathbf{B}_t \tilde{\mathbf{u}}_t^l)), \tag{8.9a}$$

$$\bar{\mathbf{P}}_{t+1} = \mathbf{A}_t \mathbf{P}_t^l \mathbf{A}_t^T + \mathbf{G}_t \mathbf{\Sigma}_{\mathbf{w}} \mathbf{G}_t^T, \tag{8.9b}$$

$$\mathbf{\Sigma}_{t+1}^{\mathbf{v}} = \mathbf{H}_{t+1} \bar{\mathbf{P}}_{t+1} (\mathbf{H}_{t+1})^T + \mathbf{M}_{t+1} \mathbf{\Sigma}_{\mathbf{v}} (\mathbf{M}_{t+1})^T, \tag{8.9c}$$

$$\mathbf{K}_{t+1} = \bar{\mathbf{P}}_{t+1} \mathbf{H}_{t+1}^T (\mathbf{\Sigma}_{t+1}^{\mathbf{v}})^{-1}, \tag{8.9d}$$

$$\mathbf{P}_{t+1}^l = (\mathbf{I} - \mathbf{K}_{t+1} \mathbf{H}_{t+1}) \bar{\mathbf{P}}_{t+1}. \tag{8.9e}$$

where $\mathbf{P}_0^l := \epsilon^2 \mathbf{\Sigma}_{\mathbf{x}_0}$ and $\hat{\tilde{\mathbf{x}}}_0^l := \mathbf{0}$. Also $\mathbf{W}_t^x = \mathbf{W}_t^T \mathbf{W}_t$ is the (non-unique) Cholesky decomposition of $\mathbf{W}_t^x$, where the diagonal entries of the real upper triangular matrix $\mathbf{W}_t$ can be zero [183]. This factorization exists because of the assumption that $\mathbf{W}_t^x$ is symmetric and positive semidefinite. 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.

Note that the evolution of $\mathbf{P}_t^l$ is deterministically dependent on the underlying nominal trajectory and is independent of the observations. Therefore, unlike the MLO method, there is no assumption on the observations in here.

**Problem 14 Belief Space Planning Problem Using T-LQG** *Given an initial belief $b_0 \in \mathbb{B}$, a goal region represented as an $\ell_2$-norm ball, $B_{r_g}(\mathbf{x}_g)$, 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:*

$$\min_{\mathbf{u}^p_{0:K-1}} \sum_{t=1}^{K} [\mathrm{tr}[\mathbf{W}_t \mathbf{P}^l_t \mathbf{W}^T_t] + (\mathbf{u}^p_{t-1})^T \mathbf{W}^u_t \mathbf{u}^p_{t-1}]$$

$$s.t. \;\; \bar{\mathbf{P}}_t = \mathbf{A}_{t-1} \mathbf{P}^l_{t-1} \mathbf{A}^T_{t-1} + \mathbf{G}_{t-1} \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{G}^T_{t-1} \tag{8.10a}$$

$$\boldsymbol{\Sigma}^{\mathbf{v}}_t = \mathbf{H}_t \bar{\mathbf{P}}_t \mathbf{H}^T_t + \mathbf{M}_t \boldsymbol{\Sigma}_{\mathbf{v}} \mathbf{M}^T_t \tag{8.10b}$$

$$\mathbf{P}^l_t = (\mathbf{I} - \bar{\mathbf{P}}_t \mathbf{H}^T_t (\boldsymbol{\Sigma}^v_t)^{-1} \mathbf{H}_t) \bar{\mathbf{P}}_t \tag{8.10c}$$

$$\mathbf{P}^l_0 = \boldsymbol{\Sigma}_{\mathbf{x}_0} \tag{8.10d}$$

$$\mathbf{x}^p_0 = \bar{\mathbf{x}}_0 \tag{8.10e}$$

$$\mathbf{x}^p_{t+1} = \mathbf{f}(\mathbf{x}^p_t, \mathbf{u}^p_t) \;\; 0 \leq t \leq K-1 \tag{8.10f}$$

$$\|\mathbf{x}^p_K - \mathbf{x}_g\|_2 < r_g \tag{8.10g}$$

$$\|\mathbf{u}^p_t\|_2 \leq r_u, \;\; 1 \leq t \leq K. \tag{8.10h}$$

*Equations* (10.20a)-(10.20c) *are regarded as one constraint at each time step, and are used to calculate the first term of the objective at that time step, equations* (10.20d) *and* (9.5b) *represent the initial conditions, equation* (10.20e) *defines the state propagation (and relates the optimization variables to the state trajectory), equation* (10.20f) *constrains the terminal state to* $B_{r_g}(\mathbf{x}_g)$*, equation* (10.20g) *accounts for the saturation constraints for* $r_u > 0$*. Moreover, the first term of the objective tends to minimize the estimation uncertainty, whereas the second term penalizes the control effort. This problem is an optimization in the space of control actions with all other variable, such as the covariances, a function of those controls. Note that* (10.20f) *is not necessary as a constraint and can be incorporated in the terminal cost, e.g., using* $\mathbf{W}^x_K$*.*

*Feedback control:* We denote the resulting optimized trajectory of problem (14) with $\{\mathbf{x}_t^o\}_{t=0}^K$, $\{\mathbf{u}_t^o\}_{t=0}^{K-1}$. The rest of the algorithm is the same as Chapter 6 and the LQG policy is defined to track the optimized trajectory as prescribed by the T-LQG algorithm. Therefore, $\mathbf{u}_t = -\mathbf{L}_t(\hat{\mathbf{x}}_t - \mathbf{x}_t^o) + \mathbf{u}_t^o$, where the feedback gain $\mathbf{L}_t$ is obtained using the backward Riccati recursions. The evolution of $\hat{\mathbf{x}}_t$ is obtained from the KF equations using the actual observations during the execution. The details of these equations are in Chapter 6.

### *8.3.1 Discussion*

*Differences between this algorithm and Chapter 6's algorithm:* Note that the algorithm presented in here is in fact a result of the algorithm presented in Chapter 6, and has the same theoretical guarantees as the T-LQG presented in there. The only difference is that, the T-LQG presented in Chapter 6 has a slightly simpler nominal trajectory design problem which only considers $J^p$. However, for the cost function that is defined in this chapter, if we only consider $J^p$, there will be no cost associated with the state along the trajectory. That cost would also be blind to the estimation performance and would not include any properties of the observation model or that of the noise. However, the design of this chapter in fact is slightly more accurate for the particularly defined cost function in that, it achieves a better pre-constant by optimizing the major component of the second order terms of the cost as well. Nevertheless, it still has the same order of optimality. However, for the problem considered in this chapter, optimizing the pre-constant also is meaningful due to the fact that the cost function does not include the cost of state itself, rather it includes the cost of deviation or the estimation performance.

*Remark:* Note that in an RHC implementation as in [58], $\mathbf{u}_t$ would only consist of $\mathbf{u}_t^o$, and, to get the corrections from the output, the planning problem is solved again at each time step from the current belief, Thereby multiplying the whole effort of the algorithm (optimization problem plus convergence to an optimized trajectory) by a factor of $K$.

*Replanning during execution:* In a stochastic system, even with a closed-loop control strategy, after a finite number of execution steps, the state estimate may deviate from the planned trajectory. This happens due to the accumulation of errors resulting from the unmodeled dynamics or forces, noise, and nonlinearities. This we have proven that happens with at most probability of $1 - P(\Omega(\epsilon^{2-\gamma}))$, the exact details of which are provided in Chapter 6. In such a situation, the planned policy becomes irrelevant and a new policy is needed to drive the agent toward the predefined goals. One can detect the deviation by several methods. For instance, testing the whiteness of the innovation in KF, checking the magnitude of innovation during the execution, and checking the magnitude of the deviation of the state estimate from the planned state are some of the methods to detect the deviation. Also calculating the offline probabilities of $1 - P(\Omega(\epsilon^{2-\gamma}))$ for each time step (e.g., by changing the value of horizon $K$ and evaluating the probabilities) can also help to predict when such a deviation is highly likely. Another method is to utilize Kullback-Leibler (KL) divergence concept, which we explain 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

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

with $q_1(\mathbf{x})$ and $q_2(\mathbf{x})$ denoting the densities of $Q_1$ and $Q_2$. Note that our approximate planned belief is $\mathcal{N}(\mathbf{x}_t^p, \mathbf{P}_t^l)$, whereas during the execution the conditional distribution is non-Gaussian. One can use a more accurate estimator during the execution and obtain the $D_{KL}$ between the two mentioned distributions in order to detect deviation. A simpler method is just to utilize a Kalman filter and approximate the conditional distribution with KF. In that situation, the $D_{KL}$ also reduces to the distance between the estimate and the planned trajectories. This is because, let $p_t^{\mathbf{b}} := \mathcal{N}(\hat{\mathbf{x}}_t, \mathbf{P}_t^l)$, and $p_t^{\mathbf{b}^p} := \mathcal{N}(\mathbf{x}_t^p, \mathbf{P}_t^l)$ be the Gaussian approximation of the distribution during the execution and the nominal Gaussian belief, respectively. Then, using the KL divergence formula for multivariate Gaussian distributions [184], the distance between $p_t^{\mathbf{b}}$ and $p_t^{\mathbf{b}^p}$ is:

$$\begin{aligned} d(p_t^{\mathbf{b}}, p_t^{\mathbf{b}^p}) =& \frac{1}{4}[\log \frac{|\mathbf{P}_t^l|}{|\mathbf{P}_t^l|} - n_x + \mathrm{tr}((\mathbf{P}_t^l)^{-1}\mathbf{P}_t^l) + (\mathbf{x}_t^o - \hat{\mathbf{x}}_t)^T(\mathbf{P}_t^l)^{-1}(\mathbf{x}_t^o - \hat{\mathbf{x}}_t)] \\ &+ \frac{1}{4}[\log \frac{|\mathbf{P}_t^l|}{|\mathbf{P}_t^l|} - n_x + \mathrm{tr}((\mathbf{P}_t^l)^{-1}\mathbf{P}_t^l) + (\hat{\mathbf{x}}_t - \mathbf{x}_t^o)^T(\mathbf{P}_t^l)^{-1}(\hat{\mathbf{x}}_t - \mathbf{x}_t^o)] \\ =& \frac{1}{2}[n_x(n_x-1) + (\hat{\mathbf{x}}_t - \mathbf{x}_t^o)^T(\mathbf{P}_t^l)^{-1}(\hat{\mathbf{x}}_t - \mathbf{x}_t^o)], \end{aligned} \tag{8.11}$$

where $|\mathbf{P}|$ denotes the determinant of the matrix $\mathbf{P}$. therefore, the only relevant information is $\hat{\mathbf{x}}_t - \mathbf{x}_t^o$. Once $\|\hat{\mathbf{x}}_t - \mathbf{x}_t^o\| > d_{th}$ a deviation is detected, the planning module is initialized with the current belief, and all planning procedures are performed again. Later in the next chapter we will discuss about the frequency of such replanning triggers. For now, it is seen that the replanning in fact occurs much less frequent than that of an RHC policy. In fact for small noise, no replanning is required.

## 8.4 Non-Convex State Constraints

Barrier functions are used for non-convex state constraints.

*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 [185]. Particularly, for an obstacle $i$, its 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. We further add several 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 of this fact. Thus, given the ellipsoid parameters $\mathcal{C} := (\mathbf{c}^1, \mathbf{c}^2, \cdots, \mathbf{c}^{n_b}) \in \mathbb{R}^{n_x \times n_b}$ and $\mathcal{E} := (\mathbf{E}^1, \cdots, \mathbf{E}^{n_b}) \in \mathbb{R}^{n_x^2 \times n_b}$, the Obstacle Barrier Function (OBF) can constructed as:

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

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 functions whose values tend to infinity along the axes of the ellipsoid at points formed by a 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. We define the cost of avoiding obstacles as:

$$\text{cost}_{obst}(\mathbf{x}_{t1}, \mathbf{x}_{t2}) := \int_{\mathbf{x}_{t1}}^{\mathbf{x}_{t2}} \Phi^{(\mathcal{E},\mathcal{C})}(\mathbf{x}')d\mathbf{x}', \tag{8.12}$$

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 (8.12) is approximated by a finite Riemann sum consisting of fewer points between $\mathbf{x}_{t1}$ and $\mathbf{x}_{t2}$. Hence, defining $\epsilon_{m'} = 1/m', m' \in \mathbb{Z}^+$ the modified obstacle cost is as follows:

$$\text{cost}_{obst}(\mathbf{x}_{t1}, \mathbf{x}_{t2}) := \sum_{\theta=0:\epsilon_{m'}:1} \Phi^{(\mathcal{P},\mathcal{C})}(\theta\mathbf{x}_{t1} + (1-\theta)\mathbf{x}_{t2})$$

Using this equation, we add the running obstacle cost of $\text{cost}_{obst}(\mathbf{x}_{t-1}, \mathbf{x}_t)$ 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 [186].

### *8.4.1 Homotopy Classes*

*Homotopy classes:* Homotopy classes of trajectories are defined as sets of trajectories that can be transformed into each other by a continuous function without colliding with obstacles [187, 188]. As shown in Fig. 8.2 the two solid trajectories are in one homotopy class, while the dashed trajectory is in a different class.

*Non-continuous policy:* When the domain of the problem is non-convex, e.g., it has banned areas as shown in Fig. 8.2, the optimal policy might not be continuous. Consider a situation where the execution trajectory arrives near a symmetrical obstacle along its line of symmetry. At that point, the random noise can push the sample path to either side of the line of symmetry. Thereafter, the optimal policy can be determined based on the last (estimated) location of the robot. That is, the policy loses its continuity because of change in the homotopy class. In that situation, in order to obtain the optimal policy, the optimal trajectories in multiple homotopy classes

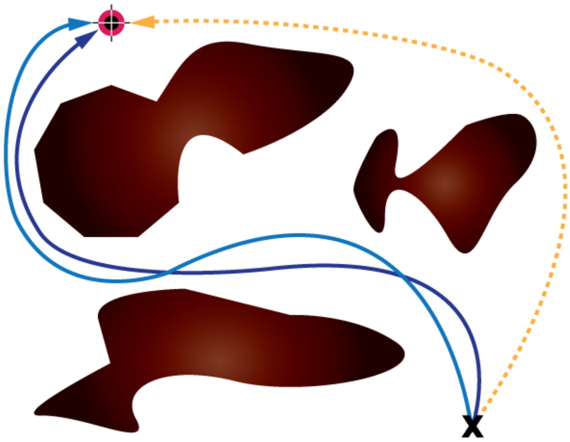


Figure 8.2: Homotopy classes. The solid trajectories are in a different homotopy class from the dashed trajectory.

need to be planned and then during the execution, the feedback policy based on the homotopy class of the robot's path can be determined. Within a single homotopy class, the policy is still continuous. Note our T-LQG analysis requires the continuity of the policy in order to provide the near-optimality guarantees. Therefore, within one homotopy class, the T-LQG will still provide the near-optimal solution. In order to provide the ner-optimal solution in the entire domain of the problem, one can obtain optimized nominal trajectories in the relevant (or if needed all non-looped) homotopy classes and then compare the solutions and choose the policy to track the trajectory. The hybrid solution of the planning in homotopy classes with the T-LQG approach will provide a near-optimal policy for the entire domain of the problem. In the next chapters we discuss more the homotopy classes and how to find them.

## 8.5 Comparison of Methods

In this section, we provide a comparison between T-LQG and other state-of-the-art belief space planning approaches from a methodological and computational complexity perspective. We make occasional references to the following methods: a)

LQG-MP [59], b) iLQG-based method [35], c) SELQR [181] d) the method utilizing MLO [57], e) the non-Gaussian Receding Horizon Control (RHC)-based method [61], f) the non-Gaussian observation covariance reduction method [62], g) the covariance-free open-loop optimization problem coupled with RHC implementation [58], and h) the point-based POMDP solvers [11, 96, 95, 26]. Table 11.1 summarizes the key differences between the methods. Regarding the Table, we note that:

- 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 [59].
- For the method of [57], $n_{tr}$ 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; for the complexity row of this method, the second provided computational complexity is valid if the B-LQR is also used, otherwise, the first provided 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 the optimization method.
- *DP* is Dynamic Programming
- *Second order* is the general rate of Newton-like methods.
- Method of [58] defines an optimization problem with dimension of $O(n_x + n_u)$ whereas T-LQG's problem dimension is $O(n_u)$. Moreover, [58] utilizes an approach similar to [57] with MLO assumption and EKF design; however, [58] utilizes RHC as the final implementation.
- The computational complexity only reflects the calculations of the core problems for belief space planning in each method. For obstacle-avoidance, each

Table 8.1: 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 [59] | None | RRT trajectories (No) | — | $O(n_r K n^3)$ | — |
| iLQG-based [35] | DP | Fixed at each iteration (No) | Stochastic observations | $O(Kn^6)$ | Second order (line-search tuning) |
| SELQR [181] | DP | Fixed at each iteration (No) | MLO | $O(Kn^6)$ | Second order |
| MLO [57] | NLP | Predicted mean update (Yes) | MLO | $O(n_{tr}(Kn^3+kn^2))$ or $O(n_{tr}(Kn^3+kn^2)+Kn^6)$ | SQP rate |
| Non-Gaussian RHC-Based [61] | Convex | Linear propagation of initial estimate (Yes) | MLO | $O(NK(Kn^3+Nn^2))$ | $\Omega((N+Kn)\log(\frac{1}{\epsilon}))$ |
| Non-Gaussian Obs. Cov. Reduction [62] | Convex | Linear propagation of initial estimate (Yes) | Predicted ensemble of observation particles | $O(Kn^3+Nn^2)$ | $\Omega(Kn\log(\frac{1}{\epsilon}))$ |
| Cov.-Free RHC[58] | NLP | Predicted mean update (Yes) | MLO | $O(Kn^3)$ | Second order |
| T-LQG | NLP | Nonlinear propagation of initial estimate (Yes) | — | $O(Kn^3)$ | Second order |

method has a different approach, which is out of the scope of this discussion and can be further detailed in a pure motion-planning perspective. The information in table 11.1 and the calculations regarding the computational complexity are estimated to the best of our knowledge.

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 11.1 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, no assumptions on observations in our method are made. Importantly, the computational complexity of T-LQG is the lowest among all of the above, while still providing a near-optimal feedback policy, a claim that none of the other methods can make.

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 11.1 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.

### *8.5.1 An Overall Summary of the Methods*

**a) LQG-MP [189]** 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 [35]** 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 [181]** 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 [57]** 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.

**e) Non-Gaussian RHC-Based [61]** 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 [62]** 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 [15]** 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 [11, 96, 95, 26]** The POMDP problem was introduced in 1971 in [11], with an algorithm to obtain the exact optimal solution using the alpha-vectors. The algorithm then evolved into an anytime algorithm in 2003 in [96], which introduced the point-based POMDP solvers. This method has been the foundation for the majority of research in the POMDP field [95]. There have been many successes in solving POMDP benchmark problems with low CPU-times. Even the latest advances in the field, such as [26], suffer from multiple limitations. For instance, the scalability with time-horizon, in particular exponential dependency, seems to be a fundamental limitation that might be difficult to 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, may not be a feasible solution for practical problems.

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* optimal 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 a solution obtain by the decoupling principle, 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 deviation is more than a tolerable threshold, replanning is done, which may be impractical in long-horizon point-based POMDP solvers. FIRM [15] 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 offline.

### *8.5.2 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 fourth 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 $O(Kn^6)$ 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 $O(n_{tr}(Kn^3 + kn^2))$, 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 [190] to obtain a *globally optimal* solution with $\epsilon$ confidence and $N$ number of samples, the algorithm requires $O(NK(Kn^3 + Nn^2))$ computations and the convergence needs $\Omega((N + Kn)log(1/\epsilon))$ calls to the oracle. In method (f), the convex problem requires $O(Kn^3 + Nn^2)$ computations and $\Omega(Knlog(1/\epsilon))$ calls to the oracle [191]. Our current method requires $O(Kn^3)$ 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.

### *8.5.3 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), ar 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), 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.

**Algorithm 1:** T-LQG

```
Input: Initial belief $\mathbf{b}_0$, Goal region $B_{r_g}(\mathbf{x}_g)$, Planning horizon $K$, Obstacle
       parameters $(\mathcal{E}, \mathcal{C})$
$t \leftarrow 0$;
while $\mathcal{P}(\mathbf{b}_t, r_g, \mathbf{x}_g) \leq p_g$ do
    if $\|\hat{\mathbf{x}}_t - \mathbf{x}^o_t\| > d_{th}$ or $t == 0$ or $t == K$ then
        Optimal Trajectory: $\{\mathbf{u}^o_{0:K-1}, \mathbf{x}^o_{0:K}\} \leftarrow$ planner$(b_0, \mathcal{E}, \mathcal{C}, K, \mathbf{x_g})$;
        $t \leftarrow 0$;
    end
    else
        Policy Function: $\hat{\mathbf{x}}_t \leftarrow \mathbb{E}[b_t]$, $\mathbf{u}_t \leftarrow -\mathbf{L}_t(\hat{\mathbf{x}}_t - \mathbf{x}^o_t) + \mathbf{u}^o_t$;
        Execution: $\mathbf{x}_{t+1} \leftarrow \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{G}_t\mathbf{w}_t$;
        Perception: $\mathbf{z}_{t+1} \leftarrow \mathbf{h}(\mathbf{x}_{t+1}) + \mathbf{M}_{t+1}\mathbf{v}_{t+1}$;
        Estimation: $\mathbf{b}_{t+1} \leftarrow \boldsymbol{\tau}_t(\mathbf{b}_t, \mathbf{u}_t, \mathbf{z}_{t+1})$;
        $t \leftarrow t + 1$;
    end
end
```

## 8.6 Simulation Results

In this section, we provide 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 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 six simulations for a KUKA youBot base model, with six different observation models including models adapted from the literature. In the second scenario, we perform a comparison between the performance of T-LQG and an RHC-based method [58]. Then, we present 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. 8.3, and the overall T-LQG algorithm is reflected in Algorithm 4. As is seen in Fig. 8.3 and Alg. 4, the planning problem starts with the supply of an initial belief and ends whenever the probability of reaching the goal region is greater than a predefined threshold $p_g > 0$. The planner function (which is the optimization Problem (14)) is fed the initial belief $\mathbf{b}_0$, the obstacle parameters $(\mathcal{E}, \mathcal{C})$, planning horizon $K$, a goal region

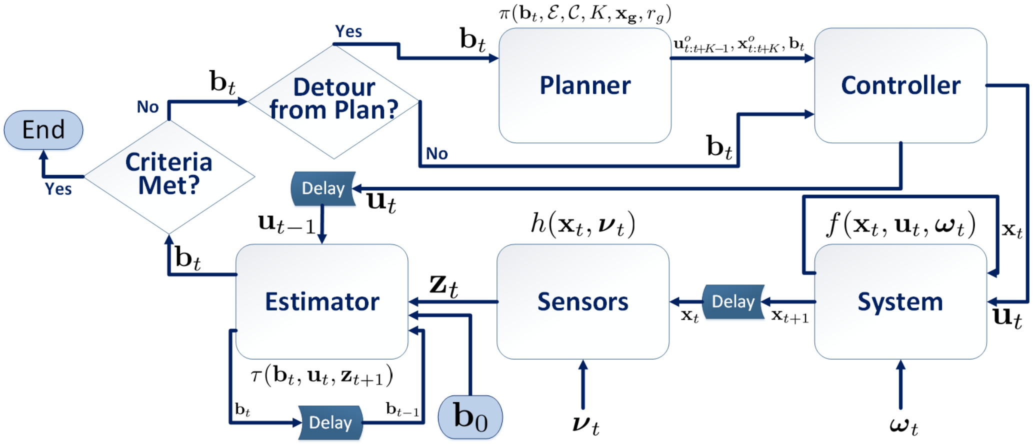
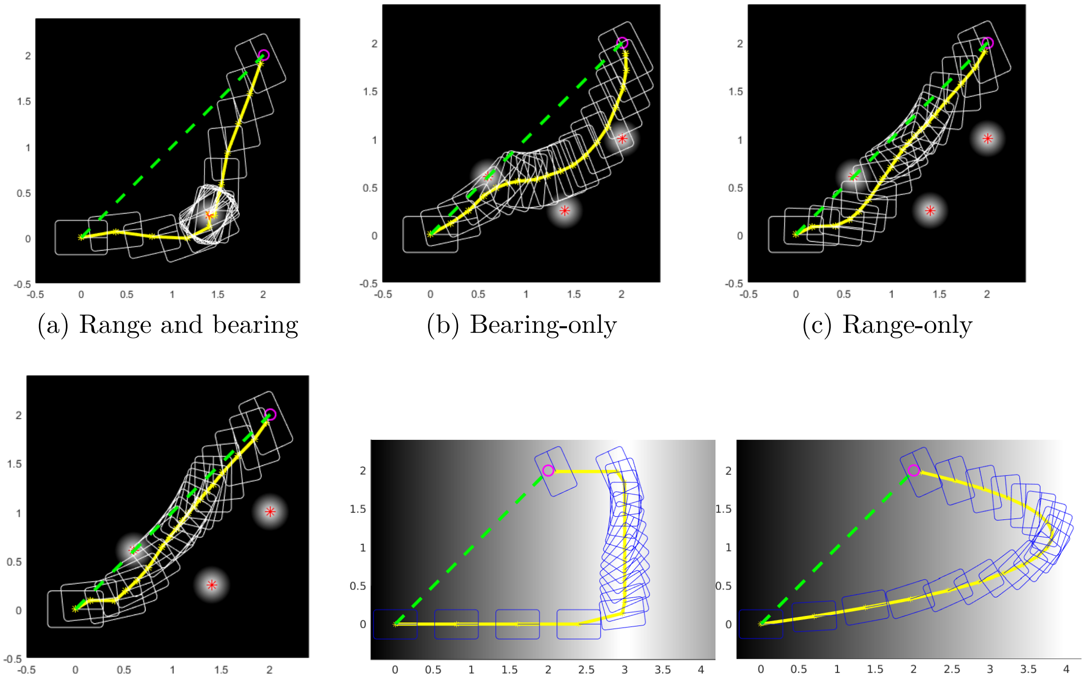


Figure 8.3: The overall feedback control loop.



(a) Range and bearing

(b) Bearing-only

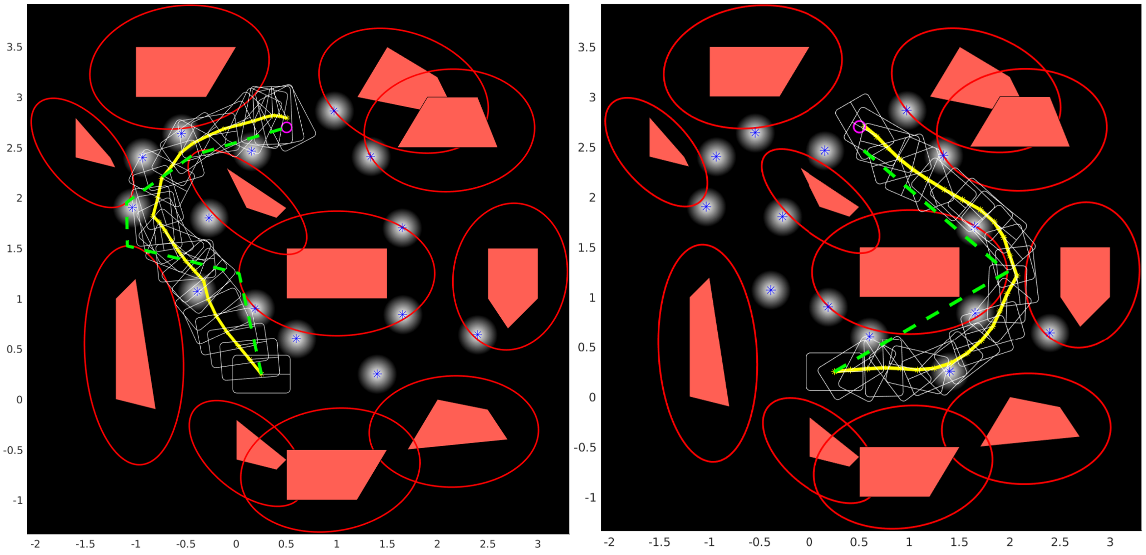
(c) Range-only

(d) Range-squared

(e) Light-dark [57]

(f) Light-dark [61]

Figure 8.4: Simulation results for an obstacle-free situation with different observation models. The information is color-coded. A lighter shade denotes 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 $r_g = 0.1$.



(a) (b)

Figure 8.5: 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, $\hat{\mathbf{x}}_0 = (0.25, 0.25, 0)$, $\mathbf{x}_g = (0.5, 2.7, 2)$, and $r_g = 0.1$. The optimized trajectory in case (b) has a lower overall cost.

$B_{r_g}(\mathbf{x}_g)$, 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 made and a new belief is obtained. If the distance between the updated belief and the nominal belief $\|\hat{\mathbf{x}}_t - \mathbf{x}^o_t\| > d_{th}$ is greater than a threshold, or the policy execution is finished but the criteria is not met, the planning algorithm restarts.

*Obstacle-free environment:* We use the kinematics equations of the KUKA youBot base as described in [192]. Particularly, the state vector can be denoted by a 3D vector, $\mathbf{x} = [\mathbf{x}_{\mathrm{x}}, \mathbf{x}_{\mathrm{y}}, \mathbf{x}_{\theta}]^T$, 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

Table 8.2: Comparison T-LQG with method of [58].

| | Final Distance from Goal (after 16 steps) | Total Time (MATLAB) |
|---|---|---|
| RHC-based [58] | 4.09 (m) | 352 (seconds) |
| T-LQG | 0.45 (m) | 20 (seconds) |

shown that the discrete motion model can be written as $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{G}_t\mathbf{w}_t = \mathbf{x}_t + \mathbf{B}\mathbf{u}_t dt + \mathbf{G}\mathbf{w}_t\sqrt{dt}$, where $\mathbf{B}$ and $\mathbf{G}$ are appropriate constant matrices, and $dt$ is the time-discretization period. The results depicted in Fig. 8.4 are for different observation models; including range and bearing; bearing-only, and range-only observations from landmarks in cases (a)-(c). In case (d), the observation function is changed to the square of the range function. Finally, in cases (e), and (f), the light-dark models of the chapters [57] and [61] 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 (e) and a hyperbolic function with a minimum at $+\infty$ in (f). Finally as noted 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.

*Comparison:* Depicted in Fig. 8.1, are the results of T-LQG and our implementation of an RHC-based method which uses the MLO assumption [58], for a youbot in a landmark-based observation model with range and bearing information. Table 10.3 compares of the costs after $K = 16$ steps of execution. In our method, the optimization problem is solved only once and the resulting feedback policy is executed without the need to re-plan. In contrast, in the method of [58], replanning is triggered at every time step in order to close the feedback loop. However, this means the optimization problem is solved 16 times, and yet the agent does not reach the goal after 16 steps. It is worth mentioning that although both methods have the same order of complexity for the optimization problem, the fact that in [58] the optimization is solved for convergence 16 times more (and possibly much more in order to reach the goal), in T-LQG it is only solved once (for this example). Thus, the overall execution time of T-LQG is $O(K)$ times lower, with more reliable plans.

*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 [193], we model the robot with a configuration of a set of points that represent the balls' centers that cover the body of the robot. As it is seen in Fig. 8.5. we have initialized 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 is 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 optimized trajectories, the better of the two (trajectory in Fig. 8.5b) is chosen as the plan for execution.

## 8.7 Conclusion

In this chapter, we have 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 trajectory for a nonlinear system with a nonlinear observation model around which to linearize, 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 and the decoupling principle that provides us with an LQR controller for a linearized system along that nominal trajectory. We have proved that the accumulated error that results 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 have also extended the method to non-convex environments by adding a cost function to avoid collision with the obstacles. Finally, we have performed simulations for a common robotic system with several observation functions in obstacle-free environments, and complex narrow passages with obstacles.

In conclusion, while T-LQG and the MLO method of [57] address a similar optimization problem, their theoretical approaches are vastly different:

- 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 derive 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 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 are made;
- 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 reduces the complexity to $O(Kn^3)$.

# 9. BELIEF-SPACE PLANNING FOR MULTI-AGENT SYSTEMS

In this chapter, we extend the belief space planning method of the previous chapter to a multi-agent situation. Particularly, we tune the MT-LQG approach to solve robotic trajectory planning problems using the concepts developed in the previous chapter. Likewise Chapter 7, we consider multi-agent planning under process and measurement uncertainties. Formulated as a stochastic control problem, the solution of a general Decentralized Partially Observed Markov Decision Process (Dec-POMDP) is a collection of feedback policies, one for each agent, maximizing a joint value function. In this chapter, we design $m$ LQG policies for $m$ number of agents maximizing the joint performance of the team. Casting the problem as a NonLinear Program (NLP), we propose a framework that reduces the optimization dimension from $(mn)^2 + mn$ to $mn$ with $n$ referring to the dimension of each individual agent's state space. As a result, the proposed method reduces the formidable generic Dec-POMDP to a computationally tractable multi-agent planning under uncertainty. Our results in 2D and 3D environments demonstrate the performance of the algorithm and its ability to predict and avoid inter-agent collisions.

## 9.1 Introduction

Finding the optimal solution for the single-agent version of this problem is *PSPACE*-complete [194]. The problem is exacerbated when controlling multiple agents, $m \geq 2$. The general Decentralized POMDP (Dec-POMDP) problem is proven to be in the *NEXP* class of problems [46].

In Dec-POMDPs, each agent obtains local observations and performs an action in response; however, the state transition probability and the incurred reward depends on the joint actions of agents. This chapter is concerned with the multi-agent navigation problem with three main objectives: *(i)* each agent needs to reach its individual goal point in the environment, while minimizing its localization uncertainty, *(ii)* the team as whole needs to avoid inter-agent collisions, *(iii)* the solution is desired to be decentralized (to reduce the communication burden), which follows from the MT-LQG algorithm's properties and the multi-agent extension of the Decoupling Principle proven in Chapter 7.

A simple extension of continuous POMDP solvers in a centralized fashion for multi-agent situations can behave poorly due to the fundamental limitations of POMDPs in dealing with high-dimensional spaces and long horizons. However, there has been significant success in finding algorithms that approximate Dec-POMDPs. Recent methods such as [56] utilize macro-actions, plan centrally for those macro-actions, and implement a local plan for each agent in a decentralized fashion. These methods are more successful than the original POMDP (or Dec-POMDP) solvers in a continuous state, as well as in action and observation spaces.

Due to the challenges in searching for a closed-loop policy, many methods aim for computing an approximate open-loop solution for the problem of planning under uncertainty. In these methods, planning under the Gaussianity assumption of the conditional distributions can significantly reduce the dimension of the problems. Particularly, whenever the policy is designed to track a reference trajectory for a system with Gaussian additive perturbations, this assumption is locally valid in many applications. We have rigorously proven this fact in the part I of this dissertation. Only in situations where there is ambiguity, such as in a kidnapped robot case do, there exist true multi-modal non-Gaussian approaches that can perform particularly better. Note that in such a situation, the perturbation may not be Gaussian either. Similarly, the small noise assumption is not valid, either.

LQG-based approaches fall into this category. As mentioned before, LQG-MP [59] compares the performance of the LQG controller on a set of trajectories constructed using RRT. However, with increasing dimensions of the system (such as in multi-agent systems), the performance of sampling-based methods can be far from optimal due to the high dispersion of sampled points. Methods such as [35] and [36] define the single-agent belief space planing problem in an $(n^2+n)$-dimensional space whose extension into a multi-agent scenario results in an $((mn)^2+mn)$-dimensional space. We have discussed the other belief space planning methods in Chapter 6.

As discussed previously, in a departure from the literature, the T-LQG [195, 196] designs an $n$-dimensional problem which provides a solution that involves a search over *closed*-loop feedback policies, and achieves the requirement for low-computation in the policy space. As a result, T-LQG provides provably better results and guarantees on the solution. The MT-LQG approach which was extended in Chapter 7 provides a theoretical design to tackle the multi-agent problem when the agents are coupled through the cost function but decoupled though the dynamics and observations. It is in fact the low-dimension of the core optimization in MT-LQG that enables an extension to higher dimensional problems.

This chapter's method differs from the existing related methods in a few aspects: *(i)* compared to open-loop methods (e.g., [57, 58]), it performs the search in the closed-loop policy space, *(ii)* there is no need for high-dimensional belief feedback as in [57, 35, 36], *(iii)* there is no need to re-plan at every step as opposed to many RHC-based Gaussian and non-Gaussian methods [61, 182], *(iv)* there is no need for an MLO assumption, *(v)* compared to point-based POMDP solvers, the underlying domain of the problem remains continuous, *(vi)* the MT-LQG method is based on the proven decoupling principle and provides near-optimality guarantees, *(vii)* the computational burden is a polynomial or low order while the approach is decentralized in execution, hence, the communication burden is also low. We provide an analysis of the method, and exhibit the performance and scalability of the method in several challenging scenarios, including multi-agent navigation in 2D and 3D environments with dynamic obstacles. We test the method with different observation models including range and bearing measurement with possibly state-dependent noise intensities. Finally, the proposed method can extend methods like [197] to multi-agent problems with long horizons.

## 9.2 MT-LQG

We follow the same notation as Chapter 7 and show the individual agents with index $i$. We also show the joint variables and spaces with index $\mathcal{I}$. The general problem and the system equations are as defined in that chapter. We use the belief space planning concepts developed in Chapter 8 and extend it to multi-agent situation in this chapter. We assume the environment's map is fully known and the only source of uncertainty is because of the uncertainty in each agents' actions and perception.

*Belief:* Similar to the earlier chapter, the conditional distribution of $\mathbf{x}_t^i$ given the data history up to time $t$, is called the information state. We refer to the information state of the linear Gaussian surrogate system as the "belief", and denote it by $\mathbf{b}_t^i := ((\hat{\mathbf{x}}_t^i)^T, \mathrm{vec}(\mathbf{P}_t^i)^T)$, a vector comprised of the mean and covariance of the conditional distribution of the linear Gaussian surrogate system for agent $i$. The update equation for the belief follows a Kalman filter.

*Linear surrogate system:* Similar to the single-agent situation, in this chapter, we use the exactly linear Gaussian system (the $l_i$-system) for agent $i$ as a surrogate to obtain the control law and estimator via the KF. The covariance of the $l_i$-system is defined as $\mathbf{P}_t^{l_i} := \mathbb{E}[(\mathbf{x}_t^{l_i} - \mathbf{x}_t^{p_i})(\mathbf{x}_t^{l_i} - \mathbf{x}_t^{p_i})^T]$, with the initial condition $\mathbf{P}_0^{l_i} = \mathbf{\Sigma}_{\mathbf{x}_0^i}$, and its evolution given by the forward recursions of the Riccati equation as follows:

$$\begin{aligned}\bar{\mathbf{P}}_t^{l_i} &= \mathbf{A}_{t-1}^i \mathbf{P}_{t-1}^{l_i} (\mathbf{A}_{t-1}^i)^T + \mathbf{G}_{t-1}^i \mathbf{\Sigma}_{\mathbf{w}} (\mathbf{G}_{t-1}^i)^T \\ \mathbf{K}_t^{l_i} &= \bar{\mathbf{P}}_t^{l_i} (\mathbf{H}_t^i)^T (\mathbf{H}_t^i \bar{\mathbf{P}}_t^{l_i} (\mathbf{H}_t^i)^T + \mathbf{M}_t^i \mathbf{\Sigma}_{\mathbf{v}} (\mathbf{M}_t^i)^T)^{-1} \\ \mathbf{P}_t^{l_i} &= (\mathbf{I} - \mathbf{K}_t^{l_i} \mathbf{H}_t^i) \bar{\mathbf{P}}_t^{l_i}. \end{aligned} \tag{9.1}$$

*Planning strategy:* We plan centrally for a number of robots that want to start their execution; then, the robots execute the plans in a decentralized manner based on the MT-LQG approach of Part I, Chapter 7. However, during the execution, whenever a robot needs replanning (because of a large deviation from its planned trajectory), it preemptively requests replanning from the central planner. In the period of time between the request and the reassessment, the robot either continues to execute its original path or, if it predicts unsafe movements, stops. The replanning strategy is discussed later in the chapter. Next, we discuss the details of the cost function.

*Estimation cost:* Similar to the single-agent belief space planning problem of Chapter 8, we just use the approximation of the cost function obtained as in (8.8g) for agent $i$. We choose $\mathbf{W}_t^{\mathbf{x}^i} \succeq 0$ and symmetric such that $\mathbf{W}_t^i := (\mathbf{W}_t^{\mathbf{x}^i})^{1/2}$, and rewrite the estimation cost as $c_t^{\text{est}}$:

$$c_t^{\text{est}}(\mathbf{x}_t^{p_i}) := \sum_{i=1}^{m} \operatorname{tr}(\mathbf{W}_t^i \mathbf{P}_t^{l_i} (\mathbf{W}_t^i)^T). \tag{9.2}$$

*Effort cost:* Similarly, the cost of effort for agent $i$ is also replaced by $\mathbf{u}_t^{p_i} \mathbf{W}_t^{\mathbf{u}^i} \mathbf{u}_t^{p_i}$ where $\mathbf{W}_t^{\mathbf{u}^i} \succeq 0$. Hence, the total cost of effort, $c_t^{\text{eff}}$, is defined as:

$$c_t^{\text{eff}}(\mathbf{u}_t^{p_i}) := \sum_{i=1}^{m} (\mathbf{u}_t^{p_i})^T \mathbf{W}_t^{\mathbf{u}^i} \mathbf{u}_t^{p_i}. \tag{9.3}$$

Also note that similar to the previous section, $|\mathbf{W}_t^{\mathbf{u}^i}| \ll |\mathbf{W}_t^{\mathbf{x}^i}|$ so that the theoretical approach of the previous section extends here, as well.

*Obstacle Penalty Function (OPF):* We extend our previous method of incorporating obstacle-avoidance cost into the cost function, presented in Chapter 8, to a case where obstacles are moving. Later in the chapter, we utilize this method for re-planning individual robots when other agents are moving (modeled as dynamic obstacles). Inspired by [193], first we cover each robot with the minimum number of spheres capable of encasing the robot's entire shape. Most importantly, the spheres are all identical in size with respect to the individual robot (note, the sphere size for each robot varies depending on the size of the robot). Once the spheres are applied, the radius for one of the spheres is taken (per robot). Then each obstacle is inflated by that value, which increases the size of the obstacle-avoidance zone. Next, we find the Minimum Volume Enclosing Ellipsoid (MVEE) [185] for each newly-inflated obstacle. Lastly, we then deflate the robot's dimensions and reduce it to a set of points—namely, the centers of the spheres encompassing robot $i$ at time $t$, which is then represented by a finite ($n_i$) number of 3D location points (or 2D if need be). These points are defined by matrix $\mathbf{L}_t^i(\mathbf{x}_t^{p_i}) := [\boldsymbol{\ell}_t^{1_i}(\mathbf{x}_t^{p_i}), \cdots, \boldsymbol{\ell}_t^{n_i}(\mathbf{x}_t^{p_i})] \in \mathbb{R}^{3\times n_i}$ where $\{\boldsymbol{\ell}_t^{k_i}(\mathbf{x}_t^{p_i})\}_{k_i=1_i}^{n_i}$ is calculated based on the orientation of the robot or from the robot's state $\mathbf{x}_t^{p_i}$.

Therefore, $\boldsymbol{\ell}_t^{k_i} : \mathbb{X}^i \to \mathbb{R}^3$ is a function of $\mathbf{x}_t^{p_i}$; however, for simplicity of the notation we use $\boldsymbol{\ell}_t^{k_i}$ and $\mathbf{L}_t^i$ in the rest of the chapter. Ellipsoid $j$'s estimated parameters at time $t$ are a center $\hat{\mathbf{o}}_t^j$ and a positive-definite matrix $\hat{\mathbf{E}}_t^j$. Defined next is

- $\hat{\mathcal{E}}_t := [\hat{\mathbf{E}}_t^1, \cdots, \hat{\mathbf{E}}_t^{n_o}] \in \mathbb{R}^{3\times 3\times n_b}$; and
- $\hat{\mathcal{O}}_t := [\hat{\mathbf{o}}_t^1, \cdots, \hat{\mathbf{o}}_t^{n_o}] \in \mathbb{R}^{3\times n_b}$

where $n_o$ is the number of obstacle ellipsoids. Moreover, we define $\mathcal{J}$ to be the set of obstacle indices $j$, which are in the neighborhood of the robot; i.e., $\sum_{k_i=1_i}^{n_i}(\boldsymbol{\ell}_t^{k_i} - \hat{\mathbf{o}}_t^j)^T\mathbf{E}^i(\boldsymbol{\ell}_t^{k_i} - \hat{\mathbf{o}}_t^j) < r_{th}$, $r_{th} > 1$. The OPF for agent $i$, $\Phi^{(\hat{\mathcal{E}}_t,\hat{\mathcal{O}}_t)} : \mathbb{X}^i \rightarrow \mathbb{R}$ is the sum of Gaussian-like functions defined as follows:

$$\Phi^{(\hat{\mathcal{E}}_t,\hat{\mathcal{O}}_t)}(\mathbf{x}_t^{p_i}) {:=} M \sum_{j \in \mathcal{J}} \sum_{k_i=1_i}^{n_i} \exp(-[(\boldsymbol{\ell}_t^{k_i} - \hat{\mathbf{o}}_t^j)^T\mathbf{E}^i(\boldsymbol{\ell}_t^{k_i} - \hat{\mathbf{o}}_t^j)]^q),$$

where $M > 0$ and $q \geq 1$. Even if the dimension of state $n_x > 3$, still $\boldsymbol{\ell}_t^{k_i}$ is at most 3D. If the obstacles are static, then the subscript $t$ of $(\hat{\mathcal{E}}_t, \hat{\mathcal{O}}_t)$ can be ignored. Furthermore, if those static obstacles are known a priori, then there is no need to estimate them either.

*Obstacle-avoidance cost:* Assuming a linear interpolation (i.e., fitting a curve using linear polynomials), we calculate the obstacle cost along the whole trajectory. If we have obstacle $j$'s parameters at time steps $t_1$ and $t_2 > t_1$, then the parameters between these two times are interpolated or estimated as well. For instance, if obstacle $j$ has translational and rotational movements, then at time $t > t_1$, $\hat{\mathbf{o}}_t^j = \hat{\mathbf{o}}_{t_1}^j + \hat{\mathbf{v}}^j(t - t_1)$ (assuming a constant estimated velocity vector $\hat{\mathbf{v}}^j$), then $\hat{\mathbf{E}}_t^j = R_{\hat{\alpha}}^j \hat{\mathbf{E}}_{t_1}^j$ where $R_{\hat{\alpha}}^j$ is the estimated rotation matrix by $\hat{\alpha}$ degrees.

However, if the obstacle changes its shape or new obstacles appear, the MVEE algorithm is used to find the new parameters. As mentioned before, the agents themselves are treated as moving obstacles.

On the other hand, for non-agent obstacles, we assume there is a separate estimator that tracks and estimates those obstacles' parameters and that our planner only uses the results obtained by that tracker to find the optimized trajectory. Otherwise, we either utilize linear interpolation or, based on the information on the state vector, the parameters are estimated. Hence, we define the cost of obstacle-avoidance for robot $i$ between $t_1$ and $t_2$, $c^{\text{obst}}_{t_1:t_2}$, as:

$$c^{\text{obst}}_{t_1:t_2}(\mathbf{x}^{p_i}_{t_1}, \mathbf{x}^{p_i}_{t_2}) := \sum_{i=1}^{m} \sum_{\theta=0:\epsilon:1} \Phi^{(\hat{\mathcal{E}}_\tau, \hat{\mathcal{O}}_\tau)}(\theta \mathbf{x}^{p_i}_{t_1} + (1-\theta)\mathbf{x}^{p_i}_{t_2}), \tag{9.4}$$

where $\tau = \theta t_1 + (1-\theta)t_2$, $\text{ceil}(1/\epsilon)$ is the number of interpolation steps and $\text{ceil}(\cdot)$ is the ceiling function.

*Collision Penalty Function (CPF):* In order to penalize (i.e., avoid) the collision between agents, we utilize a similar approach to obtain the obstacle-avoidance. As mentioned in *OPF*, $\mathbf{L}^i_t$ denotes the set of points originating from robot $i$ at time $t$'s spherical centers. Define $\mathcal{J}' := \{i+1 \leq j \leq m : \|(\mathbf{L}^i_t - \mathbf{L}^j_t)\|_F < r'_{th}, r'_{th} > \min r_i, r_j\}$, where $r_i$ and $r_j$ represent the radius value of robots $i$ and $j$ respectively, and $\|\cdot\|_F$ defines the Frobenius (Euclidean) norm of a matrix. Moreover, let $K_{i\wedge j} := \min\{K_i, K_j\}$; so, when computing the collision cost between agents $i$ and $j$, we only need to concerned when both are moving. When one agent has stopped, it can be considered a static obstacle. Then, we define the CPF for collision between agents $(i, j), j \in \mathcal{J}'$ at time $0 \leq t \leq K_{i\wedge j}$, as a function $\Psi^{(i,j)} : \mathbb{X}^i \times \mathbb{X}^j \to \mathbb{R}$; such that:

$$\Psi^{(i,j)}(\mathbf{x}^{p_i}_t, \mathbf{x}^{p_j}_t) := M' \exp(-\|(\mathbf{L}^i_t - \mathbf{L}^j_t)\|^q_F).$$

In this formula, we assume (for simplicity) that the size of $n_i$ is the same for all robots ($\forall i \in \mathcal{I}$)—i.e., all robots are represented by the same number of spheres/points. If $n_i$'s are not the same for all robots, then when computing $\Psi^i$, there are two possible solutions:

- *i)* at each time step, the difference between all components of $\boldsymbol{\ell}^{k_i}_t$ and $\boldsymbol{\ell}^{k_j}_t$ are compared; or
- *ii)* the robot with the lower number of spheres is given the same number of

spheres as the other robot and the formula above is utilized.

Note also in the above formula, $M' > 0$ should be chosen with large enough values in order to make the trajectories distinct so the sizes of the spheres are taken into account.

*Inter-agent collision-avoidance cost:* Once again assuming a linear interpolation of the trajectories, we can define the cost of inter-agent collision-avoidance, $c^{\text{coll}}_{i\wedge\mathcal{J}'}$, for agent $i$ as follows:

$$c^{\text{coll}}_{i\wedge\mathcal{J}'}(\mathbf{x}^{p_i}_{0:K_{i\wedge\mathcal{J}'}}, \mathbf{x}^{p_{\mathcal{J}'}}_{0:K_{i\wedge\mathcal{J}'}}) := \sum_{j\in\mathcal{J}'}\sum_{t=1}^{K_{i\wedge j}}\sum_{\theta=0:\epsilon:1}\Psi^{(i,j)}(\theta\mathbf{x}^{p_i}_{t-1}+(1-\theta)\mathbf{x}^{p_i}_{t}, \theta\mathbf{x}^{p_j}_{t-1}+(1-\theta)\mathbf{x}^{p_j}_{t}),$$

where:

- $\mathbf{x}^{p_i}_{0:K_{i\wedge j}} := \{\mathbf{x}^{p_i}_t\}_{t=0}^{K_{i\wedge j}}$
- $\mathbf{x}^{p_i}_{0:K_{i\wedge\mathcal{J}'}} := \{\mathbf{x}^{p_i}_{0:K_{i\wedge j}}\}_{j\in\mathcal{J}'}$
- $\mathbf{x}^{p_{\mathcal{J}'}}_t$ is defined in a similar way to $\mathbf{x}^{\mathcal{I}}_t$

The total number of collision checks between robots is $m(m-1)/2$. This cost indicates the cooperative cost in our problem.

*Optimization problem:* Problem (15) defines the optimization problem whose solution provides the underlying reference trajectory of the LQG policy for the team that has the best nominal performance among all other LQG policies. In other words, the LQG policy defined around that trajectory performs the best in terms of the estimation and tracking performance. Moreover, along with the tracking controller it is also near-optimal as proven by the decoupling principle.

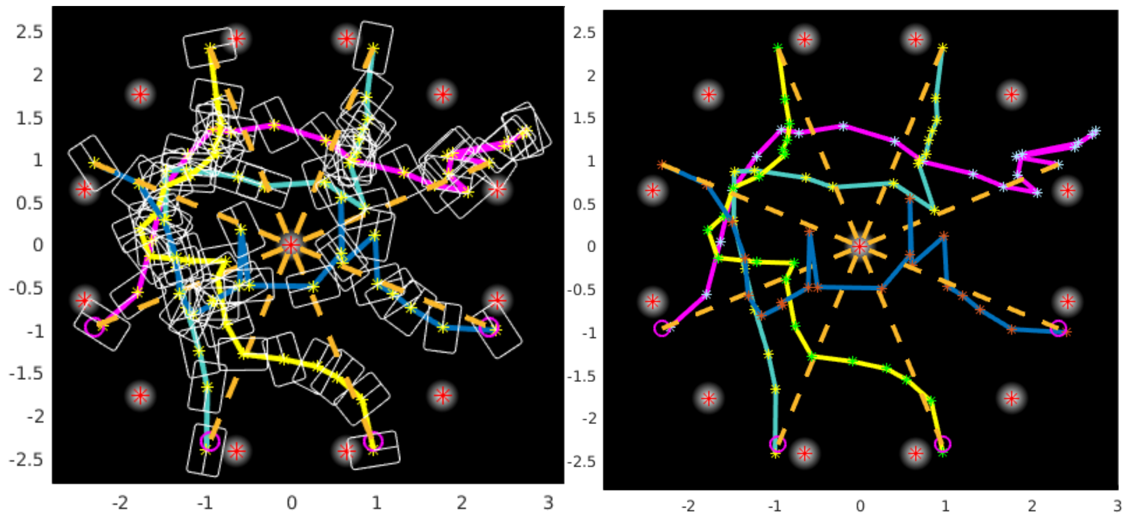
**Problem 15 Multi-Agent Planning Problem** *Given an initial joint belief $b^{\mathcal{I}}_0 \in \mathbb{B}^{\mathcal{I}}$; goal regions for each agent $B_{r^i_g}(\mathbf{x}^i_g)$ (defining an $\ell_2$-norm ball of radius $r^i_g$ around a goal state $\mathbf{x}^i_g \in \mathbb{X}^{\mathcal{I}}$); and planning horizons of $K_i > 0$ for each robot; the planning* *problem is defined as follows:*

$$
\begin{aligned}
\min_{\{\mathbf{u}^{p_i}_{0:K_i-1}, i\in\mathcal{I}\}} \sum_{i=1}^{m}\Big[\sum_{t=1}^{K_i}[c^{\text{est}}_t(\mathbf{x}^{p_i}_t) + c^{\text{eff}}_t(\mathbf{u}^{p_i}_t) + c^{\text{obst}}_{t-1:t}(\mathbf{x}^{p_i}_{t-1}, \mathbf{x}^{p_i}_t)] + c^{\text{coll}}_{i\wedge\mathcal{J}'}(\mathbf{x}^{p_i}_{0:K_{i\wedge\mathcal{J}'}}, \mathbf{x}^{p_{\mathcal{J}'}}_{0:K_{i\wedge\mathcal{J}'}})\Big]&\\
s.t.\ \bar{\mathbf{P}}^{l_i}_t =& \mathbf{A}^i_{t-1}\mathbf{P}^{l_i}_{t-1}(\mathbf{A}^i_{t-1})^T + \mathbf{G}^i_{t-1}\boldsymbol{\Sigma}_{\mathbf{w}}(\mathbf{G}^i_{t-1})^T\\
\mathbf{K}^{l_i}_t =& \bar{\mathbf{P}}^{l_i}_t(\mathbf{H}^i_t)^T(\mathbf{H}^i_t\bar{\mathbf{P}}^{l_i}_t(\mathbf{H}^i_t)^T + \mathbf{M}^i_t\boldsymbol{\Sigma}_{\mathbf{v}}(\mathbf{M}^i_t)^T)^{-1}\\
\mathbf{P}^{l_i}_t =& (\mathbf{I} - \mathbf{K}^{l_i}_t\mathbf{H}^i_t)\bar{\mathbf{P}}^{l_i}_t,\ i\in\mathcal{I} && (9.5a)\\
\mathbf{P}^{l_i}_0 =& \boldsymbol{\Sigma}_{\mathbf{x}^i_0},\ i\in\mathcal{I} && (9.5b)\\
\mathbf{x}^{p_i}_0 =& \bar{\mathbf{x}}^i_t,\ i\in\mathcal{I} && (9.5c)\\
\mathbf{x}^{p_i}_{t+1} =& \mathbf{f}(\mathbf{x}^{p_i}_t, \mathbf{u}^{p_i}_t),\ 0\leq t\leq K_i-1,\ i\in\mathcal{I} && (9.5d)\\
\|\mathbf{x}^{p_i}_K - \mathbf{x}^i_g\|_2 <& r^i_g,\ i\in\mathcal{I} && (9.5e)\\
\|\mathbf{u}^{p_i}_t\|_2 \leq& r^i_u,\ 1\leq t\leq K_i,\ i\in\mathcal{I}, && (9.5f)
\end{aligned}
$$

*Optimized trajectory of agents:* The resulting trajectory of problem (15) is the trajectory that is used as the underlying *linearization trajectory* of the system equations for each agent, and the *reference trajectory* of the LQG policy, or the *nominal trajectory* on which the nominal performance of the KF (the nominal trajectory of belief) is built upon. We denote this trajectory for agent $i$ with a superscript $o_i$.

*The LQG policy:* The LQR controller for agent $i$ is obtained by minimizing the original quadratic cost to follow the optimized trajectory. Therefore, the system equations for agent $i$ are linearized around the $o_i$ similar to equations (10.2) to obtain $\mathbf{A}^{o_i}_t, \mathbf{B}^{o_i}_t$, and $\mathbf{H}^{o_i}_t$, where the $o_i$ index shows that the Jacobians are obtained around that trajectory. Then, the following quadratic cost is minimized:

$$\sum_{t=1}^{K_i}[(\hat{\mathbf{x}}^i_t - \mathbf{x}^{o_i}_t)^T\mathbf{W}^{\mathbf{x}^i}_t(\hat{\mathbf{x}}^i_t - \mathbf{x}^{o_i}_t) + (\tilde{\mathbf{u}}^{o_i}_{t-1})^T\mathbf{W}^{\mathbf{u}^i}_t\tilde{\mathbf{u}}^{o_i}_{t-1}],$$



(a) No collision occurs

(b) Only the trajectories

Figure 9.1: Four youBots moving diagonally in a circle. The initial paths (the yellow, dashed lines) are straight lines and highly conflicting. The optimized paths (solid lines) are optimized, collision-free, and utilize the information with respect to the limited resources of effort, horizon, and collision-avoidance obstacles. The observations consists of range and bearing from landmarks, shown in lighter areas. In a) both the trajectories and robot snapshots are depicted, whereas in b) only trajectories are depicted.

where $\tilde{\mathbf{u}}_t^{o_i} := \mathbf{u}_t^i - \mathbf{u}_t^{o_i}$; which provides the feedback policy as $\tilde{\mathbf{u}}_t^{o_i} = -\mathbf{F}_t^{o_i}(\hat{\mathbf{x}}_t^i - \mathbf{x}_t^{o_i})$ with the linear feedback gain $\mathbf{F}_t^{o_i}$ given as:

$$\mathbf{F}_t^{o_i} = (\mathbf{W}_t^{\mathbf{u}^i} + (\mathbf{B}_t^{o_i})^T \mathbf{S}_t^i \mathbf{B}_t^{o_i})^{-1} (\mathbf{B}_t^{o_i})^T \mathbf{S}_t^i \mathbf{A}_t^{o_i}.$$

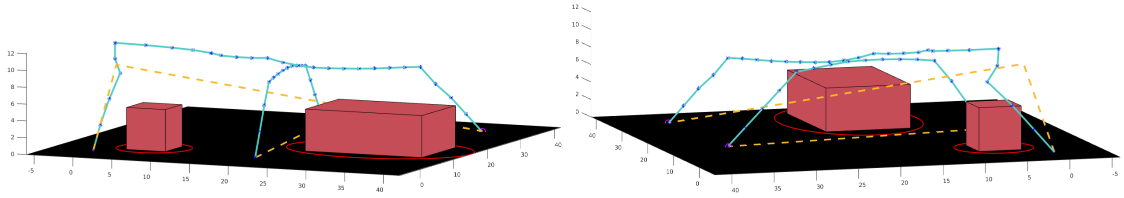
The terminal condition $\mathbf{S}_{K_i}^i = \mathbf{W}_t^{\mathbf{x}^i}$, the matrix $\mathbf{S}_t^i$ is obtained through the backward iterations of the dynamic Riccati equation:

$$\mathbf{S}_{t-1}^i = (\mathbf{A}_t^{o_i})^T \mathbf{S}_t^i \mathbf{A}_t^{o_i} - (\mathbf{A}_t^{o_i})^T \mathbf{S}_t^i \mathbf{B}_t^{o_i} (\mathbf{W}_t^{\mathbf{u}^i} + (\mathbf{B}_t^{o_i})^T \mathbf{S}_t^i \mathbf{B}_t^{o_i})^{-1} (\mathbf{B}_t^{o_i})^T \mathbf{S}_t^i \mathbf{A}_t^{o_i} + \mathbf{W}_t^{\mathbf{x}^i}.$$

Lastly, the evolution of the mean estimate is provided using a Kalman filter and

**Algorithm 2:** MT-LQG

```
Input: Initial joint belief $\mathbf{b}^{\mathcal{I}}_0$, Goal regions $\mathbf{B}^{\mathcal{I}}_{r^{\mathcal{I}}_g}(\mathbf{x}^{\mathcal{I}}_g)$, Lookahead horizons $K_{\mathcal{I}}$,
       Estimates of dynamic obstacle parameters $\{(\hat{\mathcal{E}}_t, \hat{\mathcal{O}}_t)\}_{t=1}^{\max K_{\mathcal{I}}}$
$t \leftarrow 0$;
while $\mathcal{P}(\mathbf{b}^{\mathcal{I}}_t, r^{\mathcal{I}}_g, \mathbf{x}^{\mathcal{I}}_g) \leq p_g$ do
    if $\|\hat{\mathbf{x}}^i_t - \mathbf{x}^{p_i}_t\| > d_{th}$ or $t == 0$ or $t == K$ then
        if $t == 0$ then
            Plan for all agents:
            $\{\mathbf{u}^{o_{\mathcal{I}}}_{0:(K_{\mathcal{I}}-1)}, \mathbf{x}^{o_{\mathcal{I}}}_{0:K_{\mathcal{I}}}\} \leftarrow$ planner($\mathbf{b}^{\mathcal{I}}_0, \hat{\mathcal{E}}_{0:(\max K_{\mathcal{I}})}, \hat{\mathcal{O}}_{0:(\max K_{\mathcal{I}})}, K_{\mathcal{I}}, \mathbf{x}^{\mathcal{I}}_g, r^{\mathcal{I}}_g$);
        end
        Re-plan for agent $i$:
        $\{\mathbf{u}^{o_i}_{0:K_i-1}, \mathbf{x}^{o_i}_{0:K_i}\} \leftarrow$ planner($\mathbf{b}^i_0, \hat{\mathcal{E}}_{0:K_i}, \hat{\mathcal{O}}_{0:K_i}, K_i, \mathbf{x}^i_g, r^i_g$);
    end
    else
        for $i = 1 : m$ do
            Policy Function: $\hat{\mathbf{x}}^i_t \leftarrow \bar{\mathbf{x}}^i_t$, $\mathbf{u}^i_t \leftarrow -\mathbf{F}^i_t(\hat{\mathbf{x}}^i_t - \mathbf{x}^{o_i}_t) + \mathbf{u}^{o_i}_t$;
            Execution: $\mathbf{x}^i_{t+1} \leftarrow \mathbf{f}(\mathbf{x}^i_t, \mathbf{u}^i_t) + \mathbf{G}^i_t\mathbf{w}^i_t$;
            Perception: $\mathbf{z}^i_{t+1} \leftarrow \mathbf{h}(\mathbf{x}^i_{t+1}) + \mathbf{M}^i_t\mathbf{v}^i_{t+1})$;
            Estimation: $\mathbf{b}^i_{t+1} \leftarrow \boldsymbol{\tau}(\mathbf{b}^i_t, \mathbf{u}^i_t, \mathbf{z}^i_{t+1})$;
            $t \leftarrow t + 1$;
        end
    end
end
```



(a) Higher altitudes mean less noise

(b) Another view point

Figure 9.2: Two agents in a 3D environment, with GPS observations. Higher altitudes offer less building clutter and represent less observation noise. The dashed and solid lines provide the initial and optimized trajectories of each agent. The time-stamped trajectories (marked with markers) indicate that the optimized trajectories are collision-free.

$\hat{\mathbf{x}}_0^i = \mathbb{E}(\mathbf{x}_0^i)$ by:

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

where:

- $\mathbf{f}_t^{o_i} := \mathbf{f}(\mathbf{x}_t^{o_i}, \mathbf{u}_t^{o_i}) - \mathbf{A}_t^{o_i}\mathbf{x}_t^{o_i} - \mathbf{B}_t^{o_i}\mathbf{u}_t^{o_i}$; and
- $\mathbf{h}_t^{o_i} := \mathbf{h}(\mathbf{x}_t^{o_i}) - \mathbf{H}_t^{o_i}\mathbf{x}_t^{o_i}$

Furthermore, the evolution of the covariance during execution is provided by the equation (9.1) using the Jacobians linearized around the optimized trajectory.

*Replanning strategy:* As mentioned before, planning is performed centrally for a set of robots. Each robot begins the execution of the plan in a decentralized fashion. In essence, agent $i$ keeps track of its nominal plan. Whenever, its estimate deviates from its plan greater than a threshold of $d_{th} > 0$, a replanning request occurs. At this point, the central planner assumes all moving agents are dynamic obstacles with their predicted covariances. The planner can then estimate the agents' MVEEs using that. A single-agent version of the problem (5) is solved with $m = 1$, while ignoring the $c^{\text{coll}}$ terms.

*The decentralized aspect:* In the earlier chapter, we showed that the stochastic cost function of joint problem is dominated by the nominal part of the cost function by using the decoupling principle. In our design, first the nominal joint trajectory design problem (15) is solved taking into account all the nonlinearities, the cost of obstacles, the collision cost and the estimation performance. The nominal trajectory is constructed such that it accounts for enough safety margins with regards to the obstacles and the inter-agent collisions. That is we tune the parameters of the obstacle and collision penalty functions such that the safety margin of the collision is larger than the tube size ($\delta$). Then, with high probability, the agents' trajectories will remain within their tubes. As a result, the feedback law, does not need to incorporate the shared inter-agent collision-avoidance or even the obstacle cost. That is the nominal cost of collision-avoidance or obstacle-avoidance dominates their corresponding stochastic costs and they vanish quickly within the tubes of the agents. Hence the control law for each agent is only a tracking LQR controller for an agent whose trajectory lies within its planned tube with high probability. This is also the reason behind choosing the weight matrices of the effort and state deviations block-diagonal from the beginning. Since if other wise was chosen, then a block-diagonal approximation is made by the above method. Therefore, after designing the joint nominal trajectory of the system, the resulting policy and estimation is a decentralized policy in the execution phase with low communication burden.

*Overall time complexity:* Algorithm 4 summarizes the details of the algorithm. Assuming that the dimension of state, action and observation vectors are $O(n)$, for $m$ number of agents with $O(K)$ lookahead horizon, the overall complexity of the MT-LQG algorithm is $O(mKn(n^2 + mn))$. This is much lower than the general

Dec-POMDPs in which the number of joint policies is $O((|\mathbb{U}|^{\frac{|\mathbb{Z}|^K-1}{|\mathbb{Z}|-1}})^m)$ [198], where $|\mathbb{U}| = \max_{i\in\mathcal{I}} \mathbb{U}^i$ and $|\mathbb{Z}| = \max_{i\in\mathcal{I}} \mathbb{Z}^i$ are finite (while in our method all underlying spaces are continuous). Note the exponential factor of both horizon and number robots in Dec-POMDP, whereas in our method, horizon is linear factor and number of agents is a squared factor. This shows the scalability of our method for multi-agent belief space planning.

## 9.3 Simulation Results

In this section, we provide our simulation results for 2D and 3D scenarios. We use MATLAB 2016a on a 2.90 GHz CORE i7 machine with dual core technology and 8 GB of RAM. In order to solve the optimization problem, we use MATLAB fmincon solver. In the first scenario, we consider four agents with highly conflicting paths in an obstacle-free environment, with landmark-based information sources. In the second scenario, two agents plan to reach their destinations in a highly cluttered environment. In the third situation, two agents with linear models begin their trajectory from two ground locations and, after flying in a 3D environment, reach their destinations on the ground. In the fourth situation, two agents fly in an obstacle-free space with 3D range-based observation model to obtain measurements from three antennas. In the last simulation, which is in another 3D environment, two agents have plans similar to the previous scenario, except that in this case, other than static 3D obstacles, there is also a dynamic 3D obstacle the agents must avoid.

*Robots with highly conflicting paths:* Figure 9.1 shows a scenario in which four robots with KUKA youBot base models navigate diagonally along a 2D circle. Each robot's starting position is located on a circle with equal consecutive distances. The objective is to move diagonally towards the other side of the circle in an environment without obstacles. Each robot obtained *range and bearing* measurements from a number of landmarks that fall in the visibility range of that robot. Therefore, the motion model for agent $i \in \{1, 2, 3\}$ can be written as $\mathbf{x}^i_{t+1} = f(\mathbf{x}^i_t, \mathbf{u}^i_t, \mathbf{w}^i_t) = \mathbf{x}^i_t + \mathbf{B}\mathbf{u}^i_t dt + \mathbf{G}\mathbf{w}^i_t\sqrt{dt}$, such that $\mathbf{x}^i_t$ represents the $x_x$, $x_y$ coordinates and $x_\theta$ (heading orientation) of the robot base, and $\mathbf{u}^i_t$ represents the velocities of the four wheels. Moreover, $\mathbf{B}$ and $\mathbf{G}$ are appropriate constant matrices as indicated in [199], and $dt$ is the time-sampling interval. As reflected in Fig. 9.1, the robots safely navigate to their destinations while utilizing the information from the landmarks, taking into account the limitations on their lookahead horizon, control saturation constraints, and the relatively small space provided for maneuvering.

*Two robots in a 3D environment:* Depicted in Fig. 9.2 are two agents with single-integrator dynamics ($\mathbf{A} = \mathbf{B} = \mathbf{G} = \mathbf{I}$) and equipped with GPS for observing their state (representing the three spacial coordinates, $\mathbf{x}_x, \mathbf{x}_y$, and $\mathbf{x}_z$). Although the system and observation equations are linear in this case, the covariance of the observation noise has spatial dependence. Therefore, the noise process is a spatio-temporal stochastic process approximated by a Gaussian noise process with space-dependent covariance. The covariances are constant in time and the Gaussian approximations are valid. The specific form of covariance is $\text{diag}[\sigma_1, \sigma_1, \sigma_2(\mathbf{x}_z - 20)^{-1}]$, for $\sigma_1, \sigma_2 > 0$. This is similar to a three dimensional version of the light-dark environment [61]. Moreover, we have added a constraint on the elevation which is relaxed in the beginning and the last steps of the flight. As a result, the observation covariance also remains positive-definite.

*Two youBots in a cluttered environment:* In this scenario, two youBots move in an environment with many obstacles. As shown in Fig. 9.3, the initial paths of the robots have collisions. Moreover, the initial trajectory for the robot shown on the right hand side collides with obstacles. However, the optimized trajectories are collision-free, and, as shown in the figure, they navigate closer to the information sources utilizing the range and bearing information.

*3D environment with range information:* Figure 9.4 represents a case where two robots in a similar situation to Fig. 9.2, and with similar dynamics, utilize range information from three antennas with spherical broadcast. Therefore, the robots observe their range from the antennas and navigate to get closer to the information resources while approaching their destinations. The initial trajectories as depicted by the dashed lines are highly infeasible, but the optimizer is able to convert them to fully optimized feasible paths.

*3D dynamic environment with a flying obstacle:* Figure 9.5 represents a situation where two robots with similar observation models as Fig. 9.4 fly over a 3D environment with static obstacles. However, the environment is augmented with a flying obstacle that moves with a constant speed (representing a bird, another agent or similar object) that the agents must also avoid. Therefore, the robots observe their range from the antennas and navigate to get closer to the information resources while approaching their destinations. The initial trajectories as depicted by the dashed lines are highly infeasible and the optimizer is again able to convert them to fully optimized feasible paths.

### *9.3.1 Heterogeneous Robots*

In this section, we simulate the algorithm for a heterogeneous robot model situation. For the simulations of this section, we use MATLAB 2016b's `fmincon` solver to solve the optimization problem.

*Motion models:* We use the (non-holonomic) car-like robot's model [179] for two robots:

$$\dot{x} = v\cos(\theta),\ \dot{y} = v\sin(\theta),\ \dot{\theta} = \frac{v}{L}\tan(\phi), \tag{9.6}$$

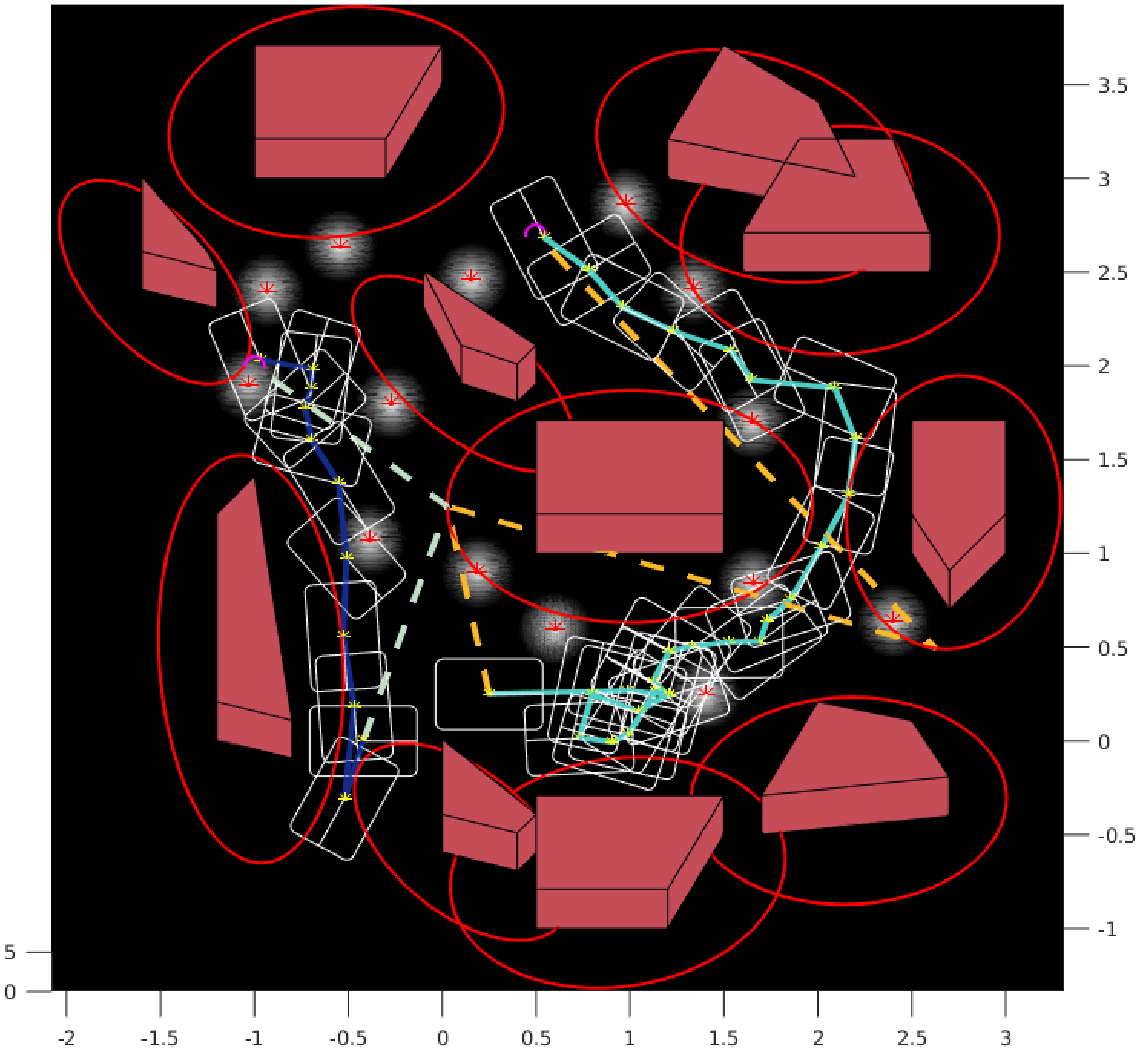


Figure 9.3: Two youBots in a cluttered environment. The initial trajectories (dashed lines) show collisions between robots (and with obstacles for the robot shown on the right side). The optimized solid trajectories are fully safe, and optimized with respect to the information sources. The purple circles on the upper side of the plot show the targets.

where $\mathbf{x} = (x, y, \theta)$, $\mathbf{u} = (v, \phi)$, $|\phi| < \pi/2$, $|v| \leq 0.6$, $|\mathbf{u}_t - \mathbf{u}_{t-1}| \leq (0.01, \pi/45)$, $\mathbf{x}_0^1 = (3.5, -1, 0)$, $\mathbf{x}_0^2 = (1, -1.1, \pi/3)$, $\mathbf{x}_g^1 = (3, 1, 5\pi/6)$, $\mathbf{x}_g^2 = (1.3, 1, \pi/2)$, $r_g^1 = r_g^2 = 0.05$,

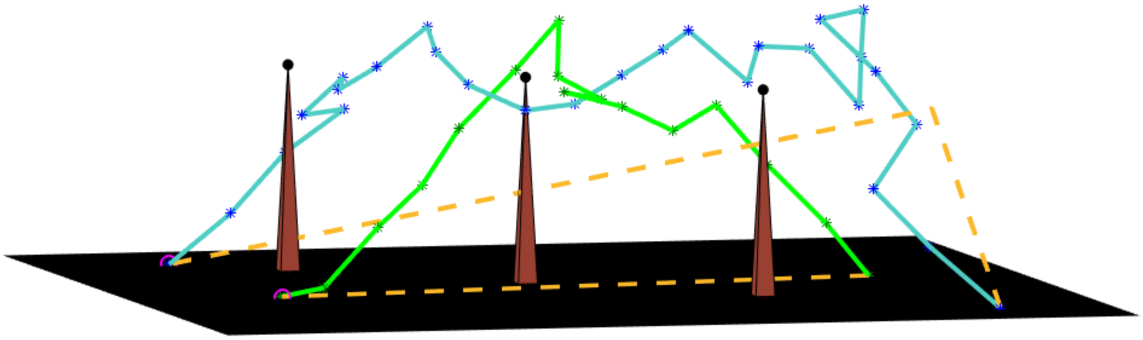


Figure 9.4: Two flying robots with information sources available as antennas, providing the range information. The initial and optimized paths are shown with dashed and solid paths, respectively. In the optimized paths, no collision occurs, and the robots fly close to the antennas. The robots are considered to be spherical, but for the clarity of the picture, only their centers are depicted. The targets are shown with purple circles, and the markers on the paths indicate the trajectory points.

and $K_1 = K_2 = 20$. We also use the (holonomic) youBot base model [199]:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{B}\mathbf{u}_t \tag{9.7}$$

where $\mathbf{x} = (x, y, \theta)^T$, $\mathbf{u} = [v_1, v_2, v_3, v_4]^T$ includes the velocity of four wheels, $\mathbf{B}$ depends on the robot sizes (which we have scaled down in our simulations), $|v_i| \leq 0.8$, $\mathbf{x}_0^3 = (1.7, 0.3, \pi/4)$, $\mathbf{x}_g^3 = (4.1, 0.7, \pi/4)$, $r_g^3 = 0.05$, and $K_3 = 25$. We perturb these motions models by additive zero-mean Gaussian noise with $\mathbf{\Sigma_w} = \epsilon^2 \max_t\{\|\mathbf{u_t}\|_2\}\mathbf{I}$ (i.e., 7% of the control). We assume $\mathbf{\Sigma}_{\mathbf{x}_0} = \epsilon^2\mathbf{I}$.

*Observation model:* We use range and bearing from the features (shown with light areas in the figures) perturbed by additive Gaussian noise with $\mathbf{\Sigma_v} = (\epsilon/5)^2\mathbf{I}$. While in the theory and planning we utilized KF, in execution phase for estimation purpose, we use EKF due to its better practical performance.

Figures 9.6 and 9.7 show the simulation results for initial planning and a typical execution trajectory for $\epsilon = 5\%$. In these figures, the two car-like robots navigate through the main obstacle to follow their individual destinations in the corridors. The third youBot in the middle is navigating through a narrow passage. While this robot expects a potential conflict with the other robot due to the narrowness of the passage, it waits enough until the passage is safe to pass it and reach the destination.

Next, for the same setup, we perform a Monte Carlo analysis on the frequency of replanning vs. $\epsilon$. For each $\epsilon$ between 1% and 10% with an step size of 1%, we conduct the entire simulation 10 times and obtain the average and standard deviation of the replanning frequency. We set the replanning criteria to be $\|\hat{\mathbf{x}}_t^i - \mathbf{x}_t^{p_i}\| > 0.3$, or $\|\mathbf{z}_t^i - \mathbf{z}_t^{p_i}\| > 1$, where the second criteria is the observation innovation. Figure 9.8 show that the noise level can be categorized into three levels. For small noise level, replanning is either not needed, or is required every (order of) 10 steps. For moderate levels of noise, replanning is required every (order of) 5 steps. Last, in the high noise levels, replanning is often required and the optimal method tends to an MPC strategy.

Note that in MATLAB, we stopped the optimizations (of replanning problems) on 3000 iterations, while for a typical problem of this size, higher iterations can potentially yield better results and less number of replannings. Moreover, we noted that parameters such as the stopping criteria, density of observation features, density of obstacles, (non-)holonomicity of dynamics, etc., all affect the frequency of replanning, which can be analyzed in a future work.

*Replanning and tube size:* As mentioned before, replanning becomes necessary when a large deviation occurs. This is happens when the execution trajectory of a robots exits its $\delta$-tube. In Chapter 6 we calculated the probability of such exit over the horizon of $K$. We noted that if the tube size is chosen to be $\delta = \sqrt{-r\log(\epsilon)}\epsilon$ for $r \geq 2$, then, the exist happen with a probability whose change rate is $o(\epsilon^r)$. The pre-constants of this probability are what determines the difference between the particular feedback policies. In fact, we also noted that with an optimized feedback, such as T-LQG (or MT-LQG), the constants $\bar{\beta}$ and $\bar{\gamma}$ change and therefore the replanning rate also changes. Therefore, although the rate of the exit probability with respect to $\epsilon$ is the same for different feedback policies, it is the feedback that can significantly increase the tolerance of the system to higher levels of noise. Particularly, our simulations showed that our design approach can be used for moderate levels of noise with infrequent replanning, as well. Whereas for small noise levels with high probability no replanning is required. On the other hand, for a high level of noise, even constant replanning may lead to undesirable results, although it is a heuristic resort. Lastly, note that the dimension of the problem and the time-horizon are the other factors that can change the exit probability and therefore the replanning frequency.

## 9.4 Conclusion

Focusing on multi-robot navigation and collision avoidance applications, in this chapter, we have presented a method to reduce decentralized POMDPs to a tractable planning problem. The solution is in the form of a collection of decentralized Linear Quadratic Gaussian (LQG) controllers for agents maximizing the joint performance of the team. The proposed solution has several desirable features: *(i)* It performs optimization over closed-loop policies, *(ii)* eliminates the need for high-dimensional belief feedback, *(iii)* does not require re-planning at every step, *(iv)* is not limited to most likely observations, and *(v)* can handle continuous domain problems. We have analyzed the method, and discussed its performance and scalability. The method's performance is demonstrated in several challenging scenarios, including multi-agent navigation in 2D and 3D environments with dynamic obstacles, in the presence of state-dependent motion and sensing uncertainties.

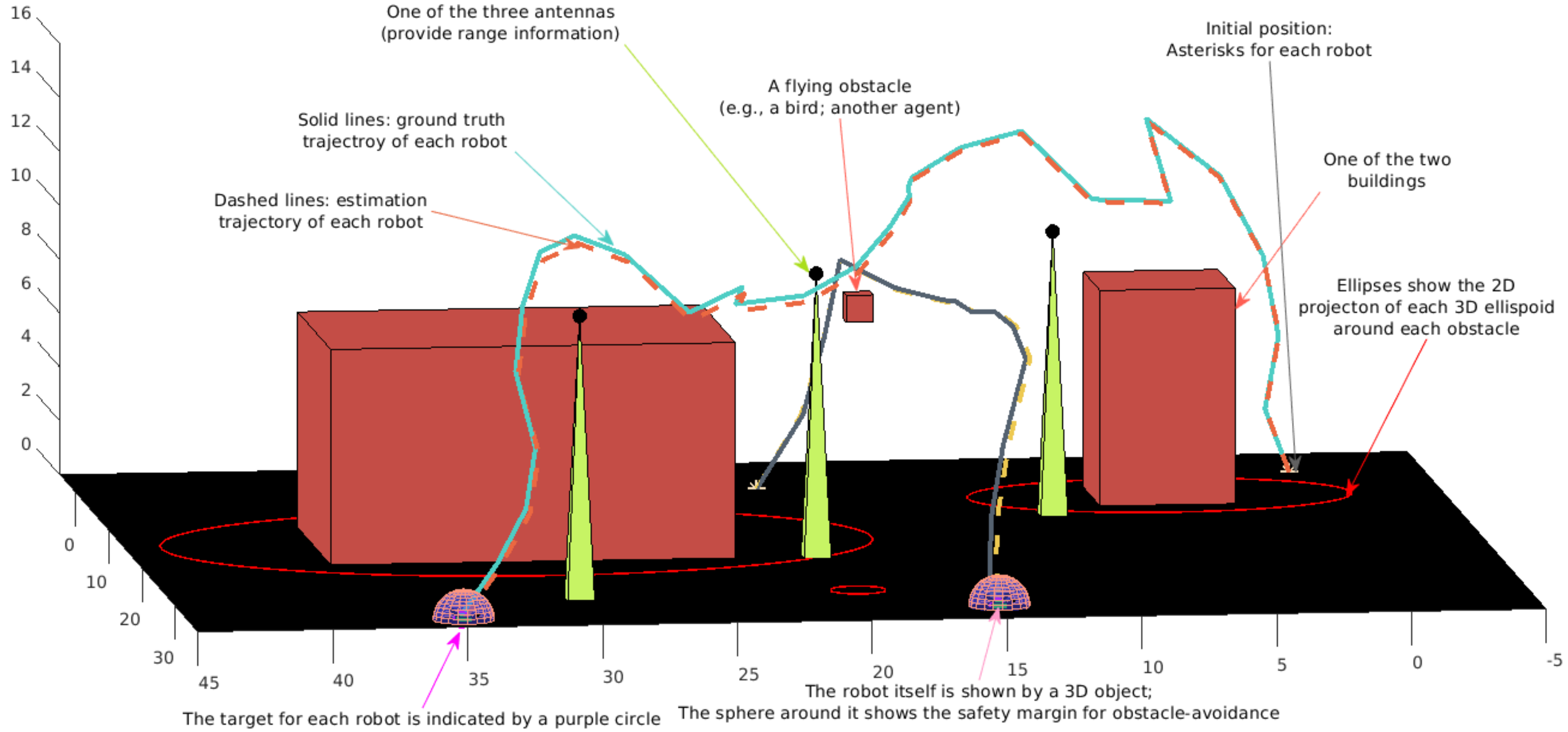


Figure 9.5: Two agents (spherical) in a 3D dynamic environment, with range sensors with respect to three ground antennas. Two tall buildings are the large static obstacles. The snapshot of a flying object is depicted. This object can represent a previously flying agent, a bird, or any similar object. No collisions occur between any objects. For simplicity only 2D elliptical projections of the 3D safety ellipsoids around the obstacles are depicted. The initial and optimized paths are shown with the dashed and solid lines respectively. The targets of the drones are shown with purple circles.

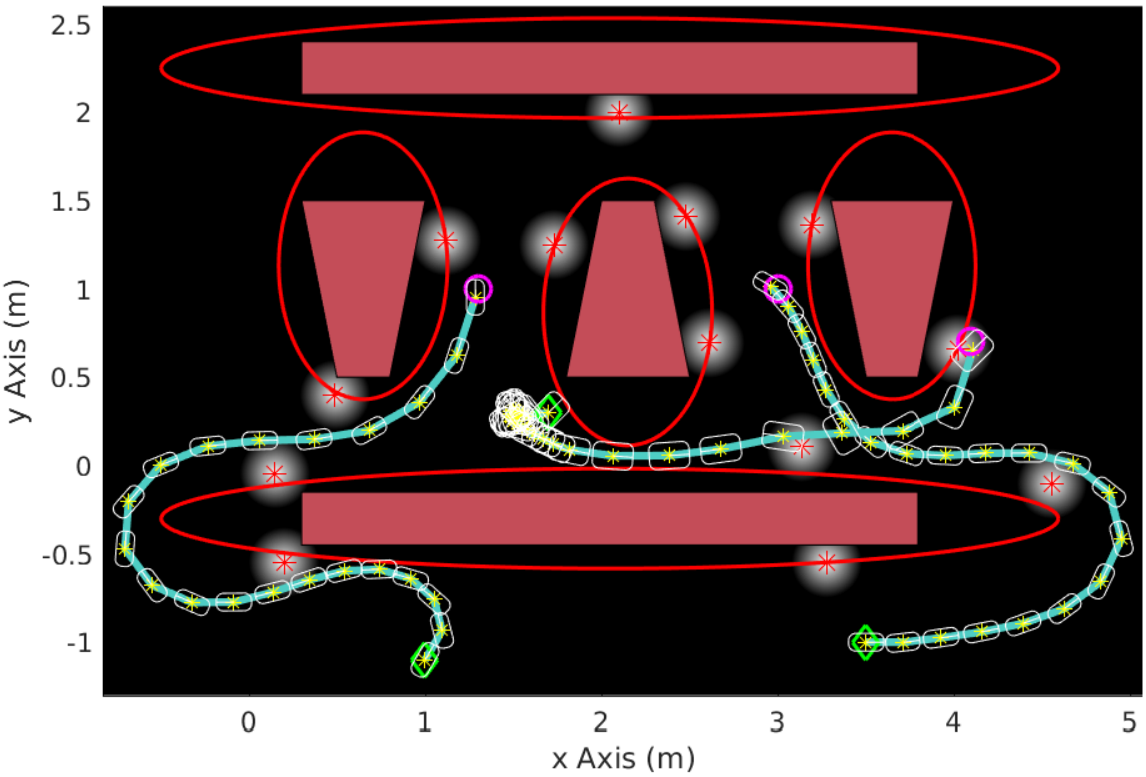


Figure 9.6: Simulation results for two car-like agents (left and right) and one youBot base (middle) in a cluttered environment with narrow passages and some features to obtain range and bearing information. The targets are depicted with purple circles. The ellipses show the regions designed to be avoided by the trajectories of the centers of the robots. Optimal planned trajectories for initial step are shown with solid lines.

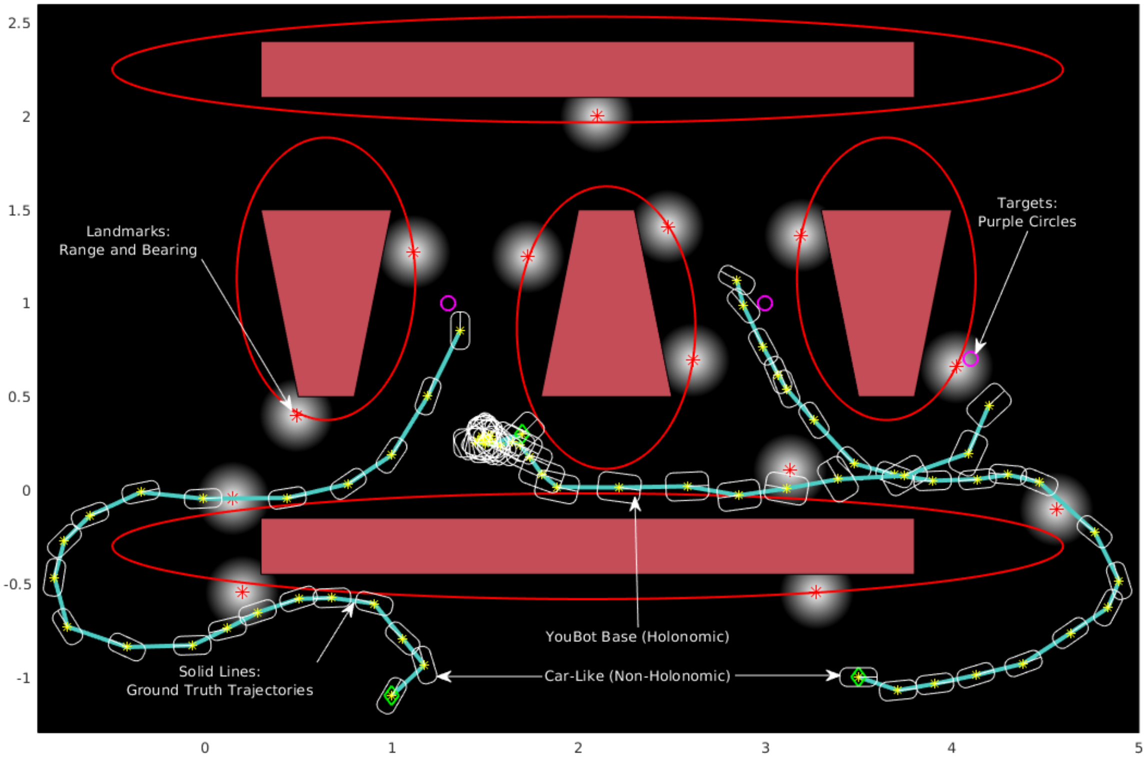


Figure 9.7: Simulation results for two car-like agents (left and right) and one youBot base (middle) in a cluttered environment with narrow passages and some features to obtain range and bearing information. The targets are depicted with purple circles. The ellipses show the regions designed to be avoided by the trajectories of the centers of the robots. Typical execution trajectories for $\epsilon = 5\%$ are shown with solid lines.

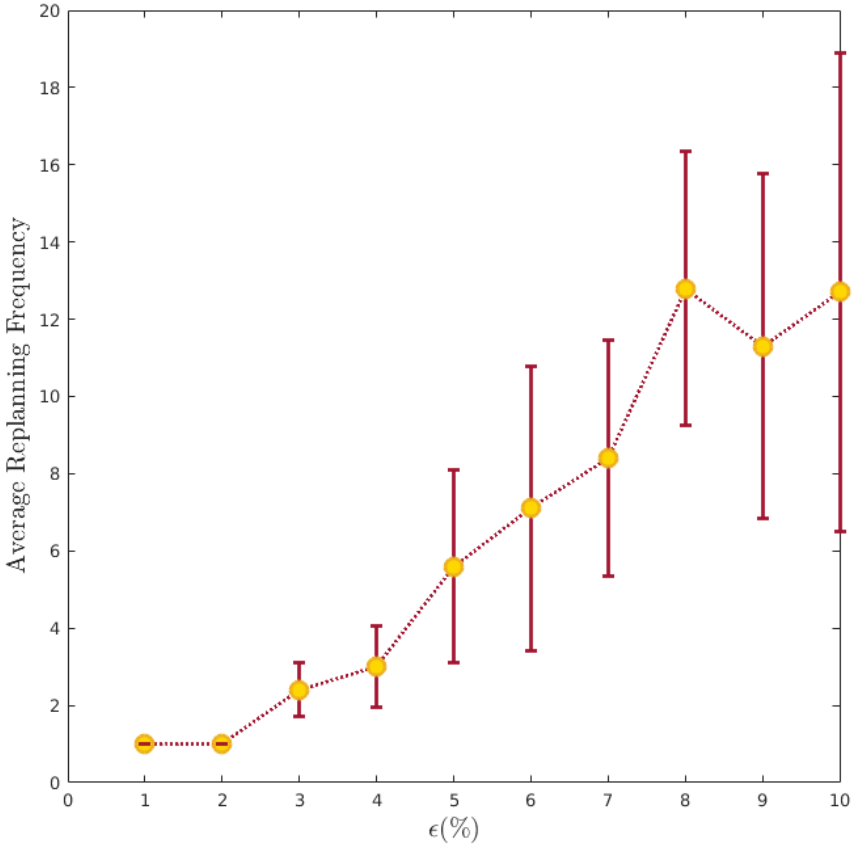


Figure 9.8: For the simulation scenario with two car-like agents and one youBot base, a Monte Carlo analysis is performed to depict the average number of replannings and their standard deviations for different noise levels.

# III. OBSERVABILITY GRAMIAN

# 10. ON THE USE OF OBSERVABILITY GRAMIAN IN ROBOTIC PATH PLANNING AND CONTROL

In this chapter, we use the theory of the previous sections, particularly, the T-LQG theory to analyze a well-known heuristic in robotic path planning and control that employs the observability Gramian of the system for planning under observation uncertainty. We consider planning under process and measurement uncertainties. We show that optimizing measures of the observability Gramian as a surrogate for the estimation performance may provide irrelevant or misleading trajectories for planning under observation uncertainty.

## 10.1 Introduction

The Observability Gramian (OG) is used to determine the observability of a deterministic linear time-varying system [200, 201, 202]. For such systems, the properties of the OG have been well-studied [200, 203, 204]. When sensors provide noisy stochastic measurements, the state is only partially observed. The general problem of planning under process and observation uncertainties has been formulated as such a stochastic control problem with noisy observations. However, the computational hurdle for finding a solution to HJB equations has necessitated the study of a variety of methods to approximate the solution [62, 61, 108, 59]. One approach has been to maximize the estimation performance by planning for trajectories that can exploit the properties of observation, process and a priori models. We examine the appropriateness or lack thereof of methods based on the OG, and show that they can provide misleading trajectories.

Borrowed from deterministic control theory, the OG has been exploited in order to provide *more observable* trajectories, particularly in trajectory planing problems

[205, 206, 207, 208, 209, 210, 211]. In the special case of a diagonal observation covariance with the same uncertainty level in each direction [200], the Standard Fisher Information Matrix (SFIM) does reduce to the OG. Indeed the usage of the OG in filtering problems has been justified through its connections to the SFIM and its relations to the parameter estimation problem [212, 206]. In fact, tailored to the parameter estimation problem, the SFIM only addresses the amount of information in the measurements alone [200], and neglects both the prior information and process uncertainty. Closely-related approaches are the methods that base their planning on the observation model or the likelihood function [62, 213], and the analysis of this chapter can be helpful in providing a better understanding of those problems.

In contrast, the *Posterior* FIM (PFIM), whose inverse coincides with the Posterior Cramér-Rao Lower Bound for the estimation uncertainty in a general stochastic problem [214], can capture the history of evolution of uncertainty in the problem. In particular, for a linear system, it has been shown that the Riccati equations for the covariance evolution of the state estimation resulting from the Kalman Filter (KF) coincide with evolution of the PFIM in the form of the inverse covariance or the information filter [214, 215, 216]. Indeed, it is only this measure that can capture the entire information required to calculate the optimal policy along with the nominal trajectory of a stochastic system. It is therefore no surprise that these equations provide the evolution of the information state (the posterior or conditional distribution of the state given the entire history of actions and observations) as the sufficient statistic for decision-making through the Bayesian filtering equations.

In this chapter, through a series of analytic and numerical examples, we show that the observability Gramian does not generally provide an appropriate solution for the problem of planning under uncertain observations. We provide examples for two commonly used nonlinear observation models including the range and squared-range observation models that provide noisy information regarding the state of the system with respect to a set of information sources or landmarks. The examples show that the OG is insensitive to the uncertainty parameters of the problem, with none of the three main covariances, i.e., process, observation or initial, appearing quantitatively. Similarly, we show that the SFIM also suffers the same problems as the OG.

The numerical examples illustrate the performance of simple planning problems when a measure of the OG (or SFIM in special case) is utilized as the optimization objective. In these examples, the trace of the error covariance, which represents the sum of mean squared errors along the trajectory, is used as the measure of performance of trajectory. In each example, the OG-based trajectory's performance is evaluated against both an initial trivial path and the optimized path with respect to the trace of the covariance. The results indicate that for all three models there are situations where the OG-based trajectory can perform significantly poorly with respect to these two trajectories, including even the initial trivial path. In some situations the trajectories produced are qualitatively similar, while their estimation performances are very different.

On the other hand, due to some very special circumstances OG-based planning may sometimes be close to the optimal outcome, and we provide such an example too. The above examples shows that OG-based planning is not reliable. One of the main reasons for usage of the OG-based method has been its relatively simpler computation, in comparison to the Riccati equation. However, we show that while there is a constant-factor computational difference in terms of the matrix calculations, a careful formulation of the original problem can lead to the same “order" of computation as the OG-based problem.

We introduce the preliminary notations and definitions of the Gramian and some

OG-based measures in the next section. Then, we proceed to the analytic examples in Section 10.3. In Section 10.4, we provide several formulations of planning problems and describe the numerical simulation results.

## 10.2 Preliminaries

We begin with some preliminary definitions.

*Process and observation models:* Let $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^{n_x}$, $\mathbf{u} \in \mathbb{U} \subset \mathbb{R}^{n_u}$ and $\mathbf{z} \in \mathbb{Z} \subset \mathbb{R}^{n_z}$ denote the state, control and observation vectors, respectively. We use boldface variables to denote the vectors in lower case and matrices in upper case, respectively. Let $\mathbf{f} : \mathbb{X} \times \mathbb{U} \times \mathbb{R}^{n_u} \to \mathbb{X}$ and $\mathbf{h} : \mathbb{X} \to \mathbb{Z}$ denote the general process and observation models:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t), \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{w}}), \tag{10.1a}$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{v}_t), \qquad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{v}}), \tag{10.1b}$$

where $\{\mathbf{w}_t\}$ and $\{\mathbf{v}_t\}$ are zero mean independent, identically distributed (i.i.d.) mutually independent random sequences, with $\mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$ denoting a normal distribution with mean $\mathbf{m}$ and covariance $\mathbf{\Sigma}$.

*Parameterized Trajectories*: Starting with an initial estimate, $\mathbf{x}^p_0 := \hat{\mathbf{x}}_0$, and using a set of unknown control inputs $\{\mathbf{u}^p_t\}_{t=0}^{K-1}$, we parametrize the possible feasible nominal trajectories of the system:

$$\mathbf{x}^p_{t+1} := \mathbf{f}(\mathbf{x}^p_t, \mathbf{u}^p_t, \mathbf{0}), \quad 0 \leq t \leq K-1,$$

$$\mathbf{z}^p_t := \mathbf{h}(\mathbf{x}^p_t, \mathbf{0}), \quad 1 \leq t \leq K.$$

*Linearization of the system equations:* We linearize the nonlinear motion and observation models of equation (10.1) about the parametrized trajectory:

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{G}_t\mathbf{w}_t, \tag{10.2a}$$

$$\tilde{\mathbf{z}}_t = \mathbf{H}_t\tilde{\mathbf{x}}_t + \mathbf{M}_t\mathbf{v}_t, \tag{10.2b}$$

where $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}^p_t$, $\tilde{\mathbf{u}}_t := \mathbf{u}_t - \mathbf{u}^p_t$, and $\tilde{\mathbf{z}}_t := \mathbf{z}_t - \mathbf{z}^p_t$ denote the state, control and observation errors, respectively, and

$$\begin{aligned}
&\mathbf{A}_t := \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{w})|_{\mathbf{x}^p_t,\mathbf{u}^p_t,\mathbf{0}}, \mathbf{B}_t := \nabla_{\mathbf{u}}\mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{w})|_{\mathbf{x}^p_t,\mathbf{u}^p_t,\mathbf{0}},\\
&\mathbf{G}_t := \nabla_{\mathbf{w}}\mathbf{f}(\mathbf{x},\mathbf{u},\mathbf{w})|_{\mathbf{x}^p_t,\mathbf{u}^p_t,\mathbf{0}}, \mathbf{H}_t(\mathbf{x}^p_t) := \nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x},\mathbf{v})|_{\mathbf{x}^p_t,\mathbf{0}},\\
&\mathbf{M}_t(\mathbf{x}^p_t) := \nabla_{\mathbf{v}}\mathbf{h}(\mathbf{x},\mathbf{v})|_{\mathbf{x}^p_t,\mathbf{0}}.
\end{aligned}$$

Note that $\{\mathbf{x}^p_t\}^K_{t=0}$, $\{\mathbf{z}^p_t\}^K_{t=0}$, and the Jacobian matrices change upon change of the underlying control inputs $\{\mathbf{u}^p_t\}^{K-1}_{t=0}$.

### *10.2.1 Observability Gramian*

*Observability Gramian:* Let $\tilde{\mathbf{A}}_t := \Pi^t_{\tau=0}\mathbf{A}_\tau$ denote the transition matrix of the linearized system of (10.2) starting from time 0. Then, the $(K{+}1)$-step observability Gramian corresponding to the nominal trajectory is defined as:

$$\mathbf{Q}^p_{K+1} := \sum_{t=0}^{K} \tilde{\mathbf{A}}^T_t\mathbf{H}^T_t\mathbf{H}_t\tilde{\mathbf{A}}_t. \tag{10.3}$$

The noise-less system of exactly linear equations is observable if and only if $\mathrm{rank}(\mathbf{Q}^p_{n_x-1}) = n_x$ [200].

Note that as the control inputs $\mathbf{u}^p_t$ change, $\mathbf{Q}^p_{K+1}$ changes, as well. This has led to a variety of approaches to utilize the OG or some function of the OG as a measure to optimize in the trajectory optimization problems. One motivating factor, as mentioned above, is the low computational burden of computing the OG. Another motivating factor for using the OG is its proven role in determining the *initial state*, $\mathbf{x}_0^p$, i.e., observability property of a deterministic system. However, in the stochastic case, *given* (partial) information around the *initial* state, the goal is to find trajectories where the state becomes more observable along the trajectory (including, in particular, the final state, which may be important to goal-oriented problems, as opposed to the initial state).

*Measures of the Gramian:* In several chapters, e.g., [212, 206], the following scalar measures of the OG have been used with various interpretations related to the uncertainty in the systems:

- Determinant of the inverse OG, $\det((\mathbf{Q}_{K+1}^p)^{-1}) = \det^{-1}(\mathbf{Q}_{K+1}^p)$ (and sometimes logarithm of it);
- Trace of the inverse OG, $\mathrm{tr}((\mathbf{Q}_{K+1}^p)^{-1})$;
- Negative trace of the OG, $-\mathrm{tr}(\mathbf{Q}_{K+1}^p)$;
- Inverse of the OG's minimum eigenvalue, $\lambda_{\min}^{-1}(\mathbf{Q}_{K+1}^p)$;
- Inverse of the OG's maximum eigenvalue, $\lambda_{\max}^{-1}(\mathbf{Q}_{K+1}^p)$;
- The condition number of the OG, $\kappa(\mathbf{Q}_{K+1}^p)$.

### *10.2.2 Standard Fisher Information Matrix*

A metric closely related to the Gramian is the SFIM the inverse of which is a lower bound on the minimum attainable estimation covariance for a parameter estimation problem as given by the Cramér-Rao lower bound [217]. The SFIM, $\mathbf{F}_K$, for the system of equations (10.2) is calculated as [200]:

$$\mathbf{F}_K = \sum_{t=0}^{K} \tilde{\mathbf{A}}_t^T \mathbf{H}_t^T \mathbf{\Sigma}_{\mathbf{v}}^{-1} \mathbf{H}_t \tilde{\mathbf{A}}_K. \tag{10.4}$$

Note that in the special case $\mathbf{\Sigma}_{\mathbf{v}} = \sigma \mathbf{I}_{n_z}$ with $\sigma > 0$, the SFIM reduces to a weighted OG:

$$\mathbf{F}_K = \frac{1}{\sigma} \sum_{t=0}^{K} \tilde{\mathbf{A}}_t^T \mathbf{H}_t^T \mathbf{H}_t \tilde{\mathbf{A}}_K = \frac{1}{\sigma} \mathbf{Q}_{K+1}^p. \tag{10.5}$$

### *10.2.3 Covariance Evolution*

*Information state:* The posterior distribution of $\mathbf{x}_t$ given the history of actions and observations up to time-step $t$, $p_{\mathbf{X}_t|\mathbf{Z}_{0:t};\mathbf{U}_{0:t-1},\mathbf{X}_0}(\mathbf{x}|\mathbf{z}_{0:t};\mathbf{u}_{0:t-1},\mathbf{x}_0)$, is referred to as the information state. It is a sufficient statistic for the stochastic control problem [2, 12]. In the linear Gaussian case, the covariance evolution of the information state is specified by the Kalman filtering equations. The covariance evolution of the KF becomes deterministic once the underlying nominal linearization trajectory of the system equations is fixed:

$$\mathbf{P}_t^- = \mathbf{A}_{t-1} \mathbf{P}_{t-1}^+ \mathbf{A}_{t-1}^T + \mathbf{G}_{t-1} \mathbf{\Sigma}_{\mathbf{w}} \mathbf{G}_{t-1}^T, \tag{10.6a}$$

$$\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{M}_t \mathbf{\Sigma}_{\mathbf{v}} \mathbf{M}_t^T, \tag{10.6b}$$

$$\mathbf{P}_t^+ = (\mathbf{I} - \mathbf{P}_t^- \mathbf{H}_t^T \mathbf{S}_t^{-1} \mathbf{H}_t) \mathbf{P}_t^-, \ \mathbf{P}_0^+ = \mathbf{\Sigma}_{\mathbf{x}_0}. \tag{10.6c}$$

## 10.3 Analytic Evaluation of OG-Based Designs

In this section, we provide two examples based on commonly used range and range-squared observation models in order to compare the amount of information and the different aspects of the models, such as stochasticity captured by the OG, the SFIM, and the PFIM equations.

*System equations:* In the examples of this section, we have $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{u} \in \mathbb{R}^2$, $\mathbf{z} \in \mathbb{R}$, and $K > 1$. Moreover, the process and observation models are:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{u}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma_w}), \tag{10.7a}$$

$$z_t = h(\mathbf{x}_t) + v_t, \qquad v_t \sim \mathcal{N}(0, \Sigma_v), \tag{10.7b}$$

where $\{\mathbf{w}_t\}$ and $\{\nu_t\}$ are zero mean i.i.d. random sequences that are mutually independent of each other, $\mathbf{x}_t = [x_t, y_t]^T$, $\mathbf{\Sigma_w} = \text{diag}(\sigma_{\mathbf{w}_x}, \sigma_{\mathbf{w}_y})$, $\Sigma_v = \sigma_\nu$, and the initial state is distributed as $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{\Sigma}_{\mathbf{x}_0})$, where $\mathbf{\Sigma}_{\mathbf{x}_0} = \text{diag}(\sigma_{x_0}, \sigma_{y_0})$. Later in the simulations, we will consider a non-diagonal initial covariance, as well. Note that except for $\mathbf{H}_t$, the other Jacobians of the above system are common to all examples, and are $\mathbf{A}_t = \mathbf{I}_2, \mathbf{B}_t = \mathbf{I}_2, \mathbf{G}_t = \mathbf{I}_2$, and $\mathbf{M}_t = \mathbf{I}_1$. As a result, $\tilde{\mathbf{A}}_t = \mathbf{I}_2, t \geq 0$.

### *10.3.1 Range-Only Example*

Our first example involves an observation that acquires the range information relative to an information source located at the origin; i.e., $h(\mathbf{x}_t) = r_t =: \sqrt{(x_t)^2 + (y_t)^2}$. The Jacobian of the observation model is $\mathbf{H}_t = (\frac{x_t}{r_t}, \frac{y_t}{r_t})$.

*The OG calculations:* The OG for this system model is

$$\mathbf{Q}^p_{K+1} = \sum_{t=0}^{K} \begin{pmatrix} \frac{x_t^2}{r_t^2} & \frac{x_t y_t}{r_t^2} \\ \frac{x_t y_t}{r_t^2} & \frac{y_t^2}{r_t^2} \end{pmatrix}.$$

Note that the determinant of the OG is

$$\det(\mathbf{Q}^p_{K+1}) = (\sum_{t=0}^{K} \frac{x_t^2}{r_t^2})(\sum_{t=0}^{K} \frac{y_t^2}{r_t^2}) - (\sum_{t=0}^{K} \frac{x_t y_t}{r_t^2})^2 > 0, \tag{10.8}$$

which is positive using the Cauchy-Schwarz inequality, excluding situations where the trajectories of the two coordinates are linearly dependent (which includes a situation in which either coordinate's trajectory is entirely zero, or a situation that the state trajectory is a straight line whose extension can pass the origin). Therefore, except for these degenerate situations this system is observable. The trace of the OG is

$$\mathrm{tr}(\mathbf{Q}^p_{K+1}) = K + 1, \tag{10.9}$$

which is a constant, insensitive to the underlying trajectory.

*Eigenvalues and condition number:* The eigenvalues of the OG are:

$$\frac{K+1}{2} \pm \sqrt{(\frac{K+1}{2})^2 - ((\sum_{t=0}^{K} \frac{x_t^2}{r_t^2})(\sum_{t=0}^{K} \frac{y_t^2}{r_t^2}) - (\sum_{t=0}^{K} \frac{x_t y_t}{r_t^2})^2)}.$$

Once again, just like the other quantities related to the OG, this quantity lacks the ability to capture the uncertainty-related aspects of the problem.

*SFIM calculations:* Since the covariance of the observations is a constant, the SFIM reduces to the form represented in equation (10.5), and $\mathrm{tr}(\mathbf{F}^p) = \sigma_\nu^{-1}\mathrm{tr}(\mathbf{Q}^p_{K+1}) = \sigma_\nu^{-1}(K+1)$, which is a constant, insensitive to the underlying trajectory, just like the trace of the OG. In fact, the SFIM is just a constant multiplier of the OG both in this and all subsequent examples.

*Covariance of the estimation calculations:* The Riccati equations of (10.6) for the evolution of the estimation covariance, in contrast, provide a different perspective than the OG and the SFIM. Starting from the initial covariance $\mathbf{P}_0^+ = \mathbf{\Sigma}_{\mathbf{x}_0}$, the covariance is:

$$\mathbf{P}_1^+ = \begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)^2}{S_1}\frac{x_t^2}{r_t^2} & -\frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)(\sigma_0^y+\sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t y_t}{r_t^2} \\ -\frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)(\sigma_0^y+\sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t y_t}{r_t^2} & (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y+\sigma_{\mathbf{w}}^y)^2}{S_1}\frac{y_t^2}{r_t^2} \end{pmatrix}, \tag{10.10}$$

which shows that the covariance ceases to be a diagonal after just one time step. Finally, the trace of the updated covariance at time-step one is:

$$\mathrm{tr}(\mathbf{P}_1^+) = \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y) + (\sigma_0^x + \sigma_{\mathbf{w}}^x + \sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{x_t^2}{r_t^2} + (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{y_t^2}{r_t^2} + \sigma_\nu}. \qquad (10.11)$$

Unlike in the case of the OG and the SFIM, minimization based on the covariance information is indeed sensitive to the underlying trajectory. In fact, this dependence is revealed after just one step of the Riccati equation's update.

### *10.3.2 Bearing-Only Example*

We consider the same system model as in equation (10.7), except that instead of a range-only observation model, we assume an absolute bearing-only model where $h(\mathbf{x}_t) = \mathrm{atan2}(y_t, x_t)$. Therefore, $\mathbf{H}_t = (-\frac{y_t}{r_t^2}, \frac{x_t}{r_t^2})$.

*The OG calculations:* The OG for this system model is

$$\mathbf{Q}_{K+1}^p = \sum_{t=0}^{K} \begin{pmatrix} \frac{y_t^2}{r_t^4} & \frac{-x_t y_t}{r_t^4} \\ \frac{-x_t y_t}{r_t^4} & \frac{x_t^2}{r_t^4} \end{pmatrix}.$$

Once again, the determinant of the OG is

$$\det(\mathbf{Q}_{K+1}^p) = (\sum_{t=0}^{K} \frac{x_t^2}{r_t^4})(\sum_{t=0}^{K} \frac{y_t^2}{r_t^4}) - (\sum_{t=0}^{K} \frac{x_t y_t}{r_t^4})^2 > 0,$$

which is again positive except in degenerate situations.

Therefore, the trace of the OG is

$$\mathrm{tr}(\mathbf{Q}_{K+1}^p) = \sum_{t=0}^{K} \frac{1}{r_t^2}, \qquad (10.12)$$

maximizing which leads to trajectories that are closer to the origin, where the information source is indeed located. Just as in the previous example, for this system too the SFIM measure also produces similar results.

*Estimation covariance:* Similar to the range-only example, given $\mathbf{P}_0^+ = \mathbf{\Sigma}_{\mathbf{x}_0}$, $\mathbf{P}_1^+$ is

$$\mathbf{P}_1^+ = \begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)^2}{S_1}\frac{y_t^2}{r_t^4} & \frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)(\sigma_0^y+\sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t y_t}{r_t^4} \\ \frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)(\sigma_0^y+\sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t y_t}{r_t^4} & (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y+\sigma_{\mathbf{w}}^y)^2}{S_1}\frac{x_t^2}{r_t^4} \end{pmatrix}. \tag{10.13}$$

Lastly, the trace of the updated covariance at time-step one is

$$\mathrm{tr}(\mathbf{P}_1^+) = \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y) + (\sigma_0^x + \sigma_{\mathbf{w}}^x + \sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{y_t^2}{r_t^4} + (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{x_t^2}{r_t^4} + \sigma_\nu}. \tag{10.14}$$

It is notable that even after just one step, the result of filtering equation differs dramatically from that of the OG or SFIM-based measures. Unlike equation (10.12), this result does not suggest a uniform radial movement towards the origin. Rather, it suggests paths that are dependent and sensitive to the direction of movement with regards to the uncertainty reduction in those directions.

### *10.3.3 Range-Squared-Only Example*

Last, we consider a model that is often used in place of the range-only model and show that the behavior of the OG changes even by a simple squaring of the observation model. We have $h(\mathbf{x}_t) = \frac{1}{2}r^2$, with Jacobian given by $\mathbf{H}_t = (x_t, y_t)$.

*The OG calculations:* The OG is

$$\mathbf{Q}_{K+1}^p = \sum_{t=0}^{K} \begin{pmatrix} x_t^2 & x_t y_t \\ x_t y_t & y_t^2 \end{pmatrix}.$$

Its determinant is

$$\det(\mathbf{Q}^p_{K+1}) = (\sum_{t=0}^{K} x_t^2)(\sum_{t=0}^{K} y_t^2) - (\sum_{t=0}^{K} x_t y_t)^2 > 0, \tag{10.15}$$

which is again taken to positive, assuming non-degenerateness.

The trace of the OG is

$$\mathrm{tr}(\mathbf{Q}^p_{K+1}) = \sum_{t=0}^{K} r_t^2,$$

maximizing which suggests trajectories that are *further* from the origin. We note that a simple squaring of the range produces exactly the opposite result, showing the inappropriateness of an OG-based design and requirement of a careful investigation with the covariance-based design. As in the previous examples the SFIM measure also produces similar results.

*Estimation covariance:* Similar to the previous two examples, given $\mathbf{P}_0^+ = \mathbf{\Sigma}_{\mathbf{x}_0}$, the updated covariance is

$$\mathbf{P}_1^+ = \begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)^2}{S_1} x_t^2 & -\frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)(\sigma_0^y+\sigma_{\mathbf{w}}^y)}{S_1} x_t y_t \\ -\frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)(\sigma_0^y+\sigma_{\mathbf{w}}^y)}{S_1} x_t y_t & (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y+\sigma_{\mathbf{w}}^y)^2}{S_1} y_t^2 \end{pmatrix}. \tag{10.16}$$

Last, the trace of the updated covariance at time-step one is

$$\mathrm{tr}(\mathbf{P}_1^+) = \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y) r_t^2 + (\sigma_0^x + \sigma_{\mathbf{w}}^x + \sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu}{(\sigma_0^x + \sigma_{\mathbf{w}}^x) x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y) y_t^2 + \sigma_\nu}. \tag{10.17}$$

Once again, this result shows that, even after just one time step, the filtering equation provides very different and reasonable solutions than the OG or SFIM measures. Unlike equation (10.16), this result does not suggest a uniform radial movement away from the origin; rather, it suggests paths that are dependent and sensitive to the direction of movement taking into account the uncertainty reductions in those directions.

### *10.3.4 Observations*

Equations (10.11) and (10.17), which represent the trace of the PFIM in each case, provide far more valuable information than the any measure of the OG:

- The trace of the updated PFIM depends on the underlying trajectory. In contrast, the trace of OG can become a constant regardless of the noise covariances, e.g., (10.9);
- PFIM, takes into account the uncertainties in each direction. In contrast, the OG-based design can be insensitive to the directions involved;
- The trace of the updated covariance is dependent on the previous covariance of the state estimation;
- The trace of covariance depends on both the observation and process noise covariances; and
- PFIM's dependence on the process, observation and previous (history of uncertainty and prior) covariances is not uniform in each direction. However, measures of the OG are insensitive to such covariances.

## 10.4 Comparison of Trajectory Planning Approaches

In this section, we consider an optimal control problem that is common in path planning and control problems, particularly in robotic systems. We introduce the general problem and describe a commonly used surrogate open-loop optimal control problem whose cost function is a measure of the OG. Finally, we compare the above approaches with belief space variant of T-LQG [218, 196], which optimizes the underlying trajectory of an LQG system aiming for the best estimation performance. This problem utilizes the trace of the covariance as the optimization objective and is accompanied by a separate feedback design implemented in the execution of the policy. In the previous chapters, we have proven the near-optimality of this framework under a small-noise assumption [196, 161].

**Problem 16 General Stochastic Control Problem** *Given* $\mathbf{x}_0 \sim p(\mathbf{x}_0)$*, solve for the optimal policy:*

$$\begin{aligned} &\min_{\pi} \ \mathbb{E}[\sum_{t=0}^{K-1} c_t^{\pi}(\mathbf{x}_t, \mathbf{u}_t) + c_K^{\pi}(\mathbf{x}_K)] \\ &s.t. \ \ \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \qquad (10.18a) \\ &\qquad \mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{v}_t), \qquad (10.18b) \end{aligned}$$

*where the optimization is over feasible policies,* $\mathbb{\Pi}$*, and:*

- $\pi \in \mathbb{\Pi}$, $\pi := \{\pi_0, \cdots, \pi_t\}$, $\pi_t : \mathbb{Z}^{t+1} \to \mathbb{U}$ *;*
- $\mathbf{u}_t = \pi_t(\mathbf{z}_{0:t})$ *specifies an action given the entire output of the system from the beginning up to time-step* $t$, $\mathbf{z}_{0:t}$*;*
- $c_t^{\pi}(\cdot, \cdot) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ *is the one-step cost function;*
- $c_K^{\pi}(\cdot) : \mathbb{X} \to \mathbb{R}$ *denotes the terminal cost; and* $K > 0$*.*

**Problem 17 OG-Based Trajectory Optimization Problem** *Solve for the optimal trajectory:*

$$\begin{aligned} &\min_{\mathbf{u}^p_{0:K-1}} g(\mathbf{Q}^p_{K+1}) + \sum_{t=1}^{K} (\mathbf{u}^p_{t-1})^T \mathbf{W}^u_t \mathbf{u}^p_{t-1} \\ &s.t. \ \ \mathbf{x}^p_{t+1} = \mathbf{f}(\mathbf{x}^p_t, \mathbf{u}^p_t, 0), \ 0 \le t \le K-1 \qquad (10.19a) \\ &\qquad \mathbf{x}^p_0 = \mathbb{E}_{\mathbf{x}}[p(\mathbf{x}_0)] \qquad (10.19b) \end{aligned}$$

$$\|\mathbf{x}_K^p - \mathbf{x}_g\|_2 < r_g \tag{10.19c}$$

$$\|\mathbf{u}_t^p\|_2 \leq r_u,\ 1{\leq}t{\leq}K, \tag{10.19d}$$

*where the optimization is over feasible controls, $g : \mathbb{R}^{n_x \times n_x} \to \mathbf{R}$ represents a specific operation on the OG, such as trace, determinant, etc., $\mathbf{W}_t^u \succeq 0$, $r_u > 0$, and $r_g > 0$ and $\mathbf{x}_g \in \mathbb{X}$ specify the goal region.*

**Problem 18 T-LQG Planning Problem [196]** *Solve for the optimal linearization trajectory of the LQG policy:*

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

$$s.t.\ \ \mathbf{P}_t^- = \mathbf{A}_{t-1}\mathbf{P}_{t-1}^+\mathbf{A}_{t-1}^T + \mathbf{G}_{t-1}\mathbf{\Sigma}_{\mathbf{w}_{t-1}}\mathbf{G}_{t-1}^T \tag{10.20a}$$

$$\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_t^-\mathbf{H}_t^T + \mathbf{M}_t\mathbf{\Sigma}_{\mathbf{v}_t}\mathbf{M}_t^T \tag{10.20b}$$

$$\mathbf{P}_t^+ = (\mathbf{I} - \mathbf{P}_t^-\mathbf{H}_t^T\mathbf{S}_t^{-1}\mathbf{H}_t)\mathbf{P}_t^-,\ \mathbf{P}_0^+ = \mathbf{\Sigma}_{\mathbf{x}_0} \tag{10.20c}$$

$$\mathbf{x}_0^p = \mathbb{E}_{\mathbf{x}}[p(\mathbf{x}_0)] \tag{10.20d}$$

$$\mathbf{x}_{t+1}^p = \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p, 0),\ 0{\leq}t{\leq}K{-}1 \tag{10.20e}$$

$$\|\mathbf{x}_K^p - \mathbf{x}_g\|_2 < r_g \tag{10.20f}$$

$$\|\mathbf{u}_t^p\|_2 \leq r_u,\ 1{\leq}t{\leq}K, \tag{10.20g}$$

*where the optimization is over feasible controls, and equations* (10.20a)-(10.20c) *represent one iteration of the Riccati equation to calculate the first term of the objective.*

We now describe the performance of the above approaches. We perform several numerical simulations for various initial, process and observation uncertainties for both of the problems (17) and (18) and all three observation models.

First, we provide an example for the range-squared observation model, where we show that the trajectory provided by the OG-based problem of (17) can significantly under-perform in terms of reducing the estimation uncertainty. We show that planning based on the OG can result in undesirable trajectories for these partially observed problems, which stems from the fact that the OG is insensitive to the uncertainty parameters of the problem and provides the same result regardless of the changes in the three covariances.

Next, we provide an example for the other model where qualitatively the output trajectories of the two problems resemble each other, but the covariance evolution results in the slight differences in the state trajectory contributing to a significant difference in the qualities of the trajectories in terms of the filters' performances. Lastly, we provide an example showing that when the intensity of noises tends to zero (particularly, if the sensor noise is very low), the performances of the OG-based and covariance-based trajectories tend to be close to each other. All our simulations are performed in MATLAB 2016b using the `fmincon` solver.

For all the figures that depict the state trajectories:

- $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{u} \in \mathbb{R}^2$, $\mathbf{z} \in \mathbb{R}$, and $K = 7$;
- $\mathbf{W}^u_t = 0\mathbf{I}_2$, $r_u = 0.8$, $r_g = 0.1$ and $\mathbf{x}_g = (-1, 2.25)^T$, which is indicated by a purple circle in the figures;
- The units of the axes are in meters;
- The initial estimate is $\hat{\mathbf{x}} = (-1.5, -0.5)^T$, which is indicated by a green diamond in the figures;
- The information sources are located at the centers of the light areas in the figures;

- The initial trajectory for the solver, indicated with a dashed orange line, consists of three straight segments passing through $(-1.5, -0.5)^T$, $(-1.4, 0.21)^T$, $(-1.1, 1.369)^T$, and $(-1, 2.25)^T$. Hence, the deterministic system is observable for all three models; and

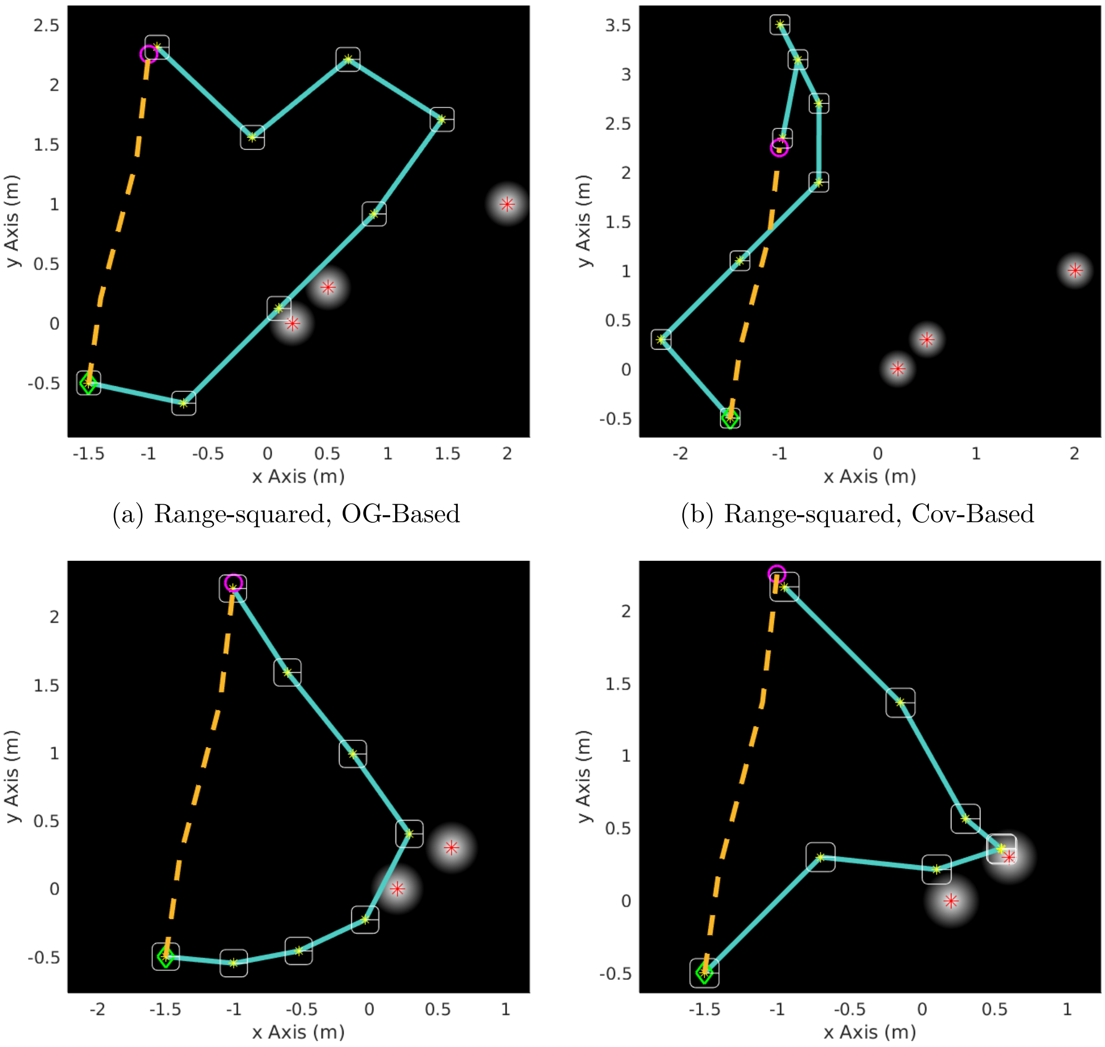
- The optimized trajectory is shown by a solid cyan line.

### *10.4.1 Range-Squared-Only Observations*

Figures 10.1a and 10.1b show the results of the simulations for the range-squared-only observation model using the condition number of the OG and the trace of the covariance along the trajectory as the cost function, respectively. Information sources are at $(0.2, 0)^T$, $(0.5, 0.3)^T$, and $(2, 1)^T$, and

$$\boldsymbol{\Sigma}_{\mathbf{x}_0} = \begin{pmatrix} 0.025 & 0.002 \\ 0.002 & 0.025 \end{pmatrix}, \boldsymbol{\Sigma}_{\mathbf{w}} = \begin{pmatrix} 0.3 & 0.0 \\ 0.0 & 0.1 \end{pmatrix}, \Sigma_v = 0.1.$$

Figure 10.2a shows the evolution of the trace of covariance along the trajectories. While it is expected that the trajectory deigned based on the covariance evolution performs better than the other ones, it is surprising to observe that the OG-based trajectory actually under-performs the initial trajectory as well. Even though we have only shown the results of the simulation for the condition number of OG, the interested reader can find a more detailed set of experiments with other measures of the Gramian in in the next sections, which parallel the results provided here. The quantitative result of Fig. 10.2a, along with the qualitative difference in the trajectories as indicated in Fig. 10.1, indicate that a measure of the OG is not a reliable measure to optimize in a problem with initial, process and observation uncertainties.



(a) Range-squared, OG-Based

(b) Range-squared, Cov-Based

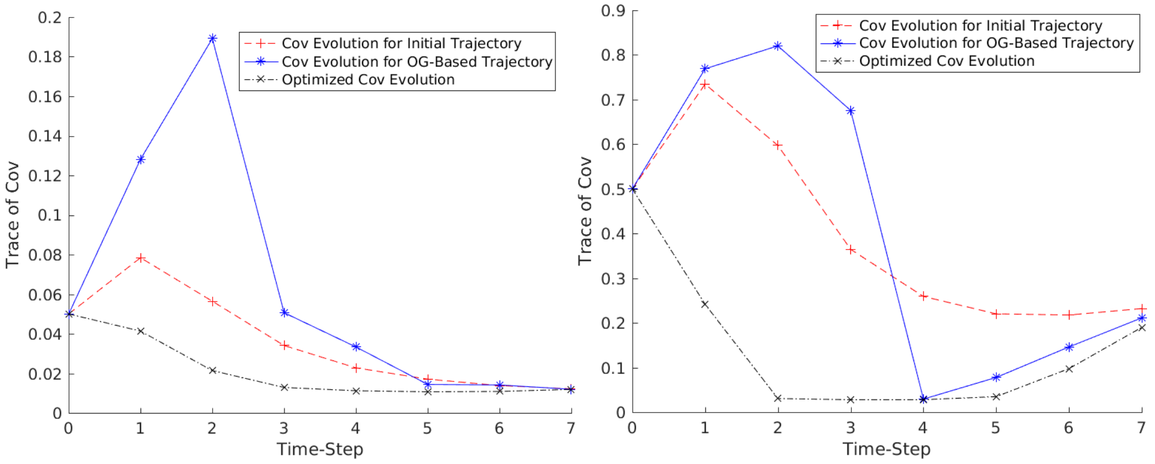
(c) Range, OG-Based

(d) Range, Cov-Based

Figure 10.1: Simulation results for the planning problem (17) based on the condition number of the OG for range-squared and range observation models in (a) and (c), and the planning problem (18) using the trace of the covariance for range-squared and range observation models in (b) and (d), respectively. The information sources are located at the centers of the light areas. The dashed orange line represents the initial trajectory, while the solid cyan line shows the optimized trajectory.

### *10.4.2 Range-Only Observations*

Figures 10.1c and 10.1d show the results of the similar simulations for the range-only observation model with the condition number of the OG and the trace of the



(a) Range-squared

(b) Range

Figure 10.2: Evolution of the trace of the covariance along the trajectory for the initial trajectory, with optimization based on the OG measure, and optimization based on the covariance measure of the trajectories in Fig. 10.1.

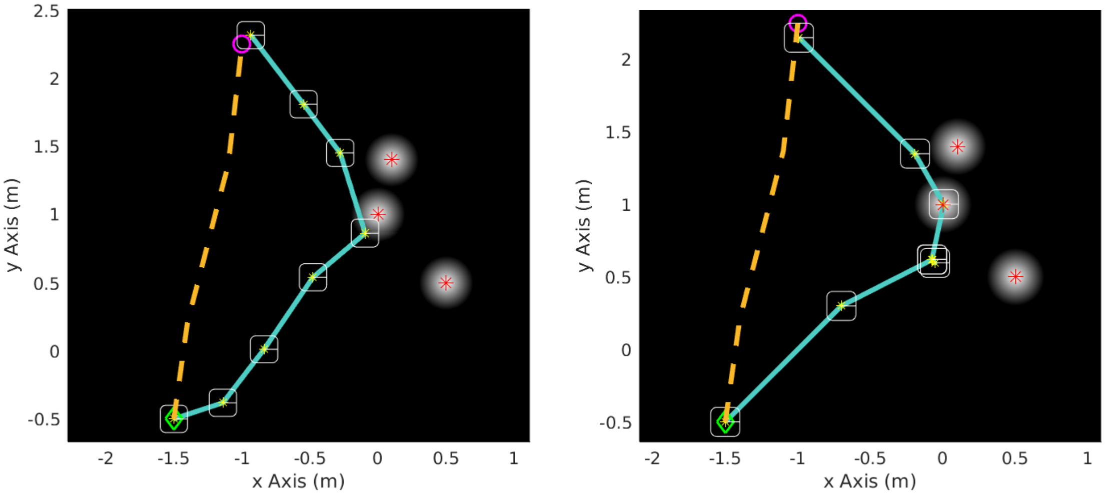
covariance as the cost function, respectively. Information sources are at $(0.2, 0)^T$, and $(0.6, 0.3)^T$, and

$$\boldsymbol{\Sigma}_{\mathbf{x}_0} = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix}, \boldsymbol{\Sigma}_{\mathbf{w}} = \begin{pmatrix} 0.1 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma_v = 0.015.$$

Figure 10.2b shows the covariance evolution for the trajectories of this simulation, which resembles the results of Fig. 10.2a.

### *10.4.3 Bearing-Only Observations*

Figure 10.5 shows the results of simulations for the bearing-only observation model, where in Fig. 10.5a the condition number of the observability Gramian is utilized as the cost function, whereas in Fig. 10.5b, optimization problem (18) is solved using the trace of the covariance along the trajectory as the cost function.



(a) OG-Based Trajectory

(b) Cov-Based Trajectory

Figure 10.3: Range-only observation model: a) The optimized state trajectory of the planning problem (17) using the condition number of the OG as the cost function, b) The optimized state trajectory of the planning problem (18) using the trace of the covariance as the cost function. The information sources are located at the centers of the light areas. The dashed orange line represents the initial trajectory, while the solid cyan line shows the optimized trajectory.

For the simulations of this figure, we have

$$\mathbf{\Sigma}_{\mathbf{x}_0} = \begin{pmatrix} 0.25 & 0.2 \\ 0.2 & 0.25 \end{pmatrix}, \mathbf{\Sigma}_{\mathbf{w}_t} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}, \Sigma_{v_t} = 0.02.$$

Similar to the previous case, for this experiment, two information sources are located at $(0.1, 0.8)^T$, and $(0.1, 1.4)^T$.

Figures 10.2b and 10.6 show the trace of the covariance evolution for the range and bearing models. As indicated in the figures, qualitative resemblance of the trajectories for covariance based optimization and OG-based optimization does not translate directly to the same quality in estimation performance. Indeed, the OG-based trajectories under-perform their covariance-based optimized counterparts significantly.

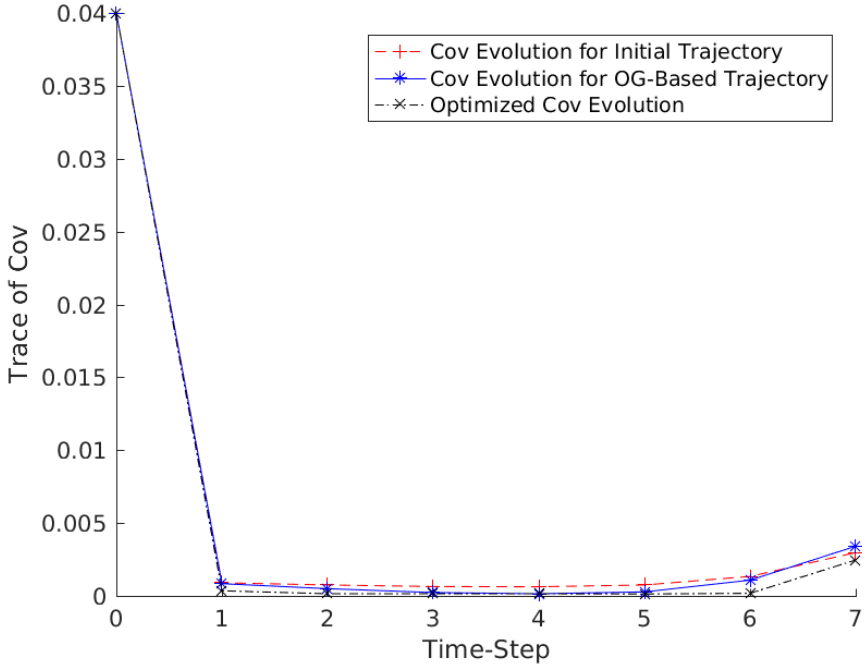


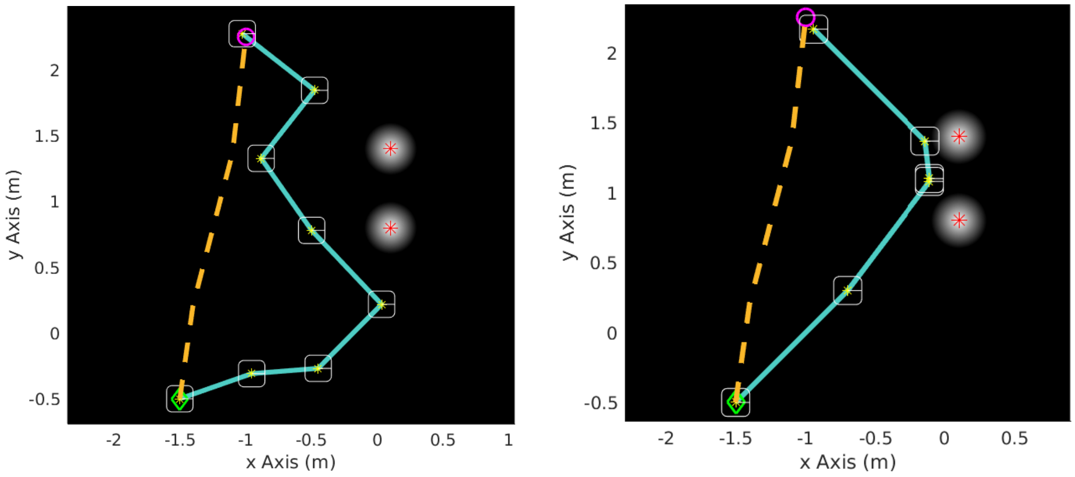
Figure 10.4: Range observation model. Evolution of the trace of the covariance along the trajectory for the initial trajectory, with optimization based on the OG measure, and optimization based on the covariance measure of the trajectories in Fig. 10.3.

### *10.4.4 Another Range-Only Scenario*

Last, Figs. 10.3a and 10.3b show the results of another set of simulations for the range-only observation model using condition number of the OG and the trace of the covariance, respectively. Information sources are located at $(0,1)^T$, $(0.5,0.5)^T$, and $(0.1,1.4)^T$, and

$$\mathbf{\Sigma}_{\mathbf{x}_0} = \begin{pmatrix} 0.02 & 0 \\ 0 & 0.02 \end{pmatrix}, \mathbf{\Sigma}_{\mathbf{w}} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}, \Sigma_v = 0.0001.$$

In this experiment, the reduced noise covariances, particularly the observation covariance, lead to the high quality of measurements from a broad class of trajectories. As a result, the trace of covariance evolution of Fig. 10.3 indicates only a slight



(a) OG-Based Trajectory

(b) Cov-Based Trajectory

Figure 10.5: Bearing-only observation model: a) The optimized state trajectory of the planning problem (17) using the condition number of the OG as the cost function, b) The optimized state trajectory of the planning problem (18) using the trace of the covariance as the cost function. The information sources are located at the centers of the light areas. The dashed orange line represents the initial trajectory, while the solid cyan line shows the optimized trajectory.

difference between the three trajectories.

*Remark:* It should be noted that in all the figures, since the state trajectories are softly constrained to reach to the same goal region at the end of the navigation, the covariance evolutions converge to each other towards the end of the trajectories.

Table 10.1: Simulation results of cost, constraint satisfaction, and time to optimize in the optimization problems of Figs. 10.1, 10.5, and 10.3.

| | Fig. 10.1a | Fig. 10.1b | Fig. 10.1c | Fig. 10.1d | Fig. 10.5a | Fig. 10.5b | Fig. 10.3a | Fig. 10.3b |
|---|---|---|---|---|---|---|---|---|
| Initial Cost | 5.24946 | 0.23473 | 5.74254 | 2.62537 | 4.07897 | 1.88394 | 3.27267 | 0.00798 |
| Final Cost | 1 | 0.12019 | 1 | 0.65005 | 1 | 0.37949 | 1 | 0.00357 |
| Initial Constraint | -0.09972 | -0.09972 | -0.09972 | -0.09972 | -0.09972 | -0.09972 | -0.09972 | -0.09972 |
| Final Constraint | -0.00301 | 1.89e-09 | -0.05709 | 3.37e-11 | -0.06375 | 2.91e-11 | -0.00739 | 1.15e-10 |
| Simulation Time (s) | 3.6230 | 2.6583 | 2.6180 | 3.3281 | 1.1509 | 1.7805 | 2.3298 | 3.3330 |

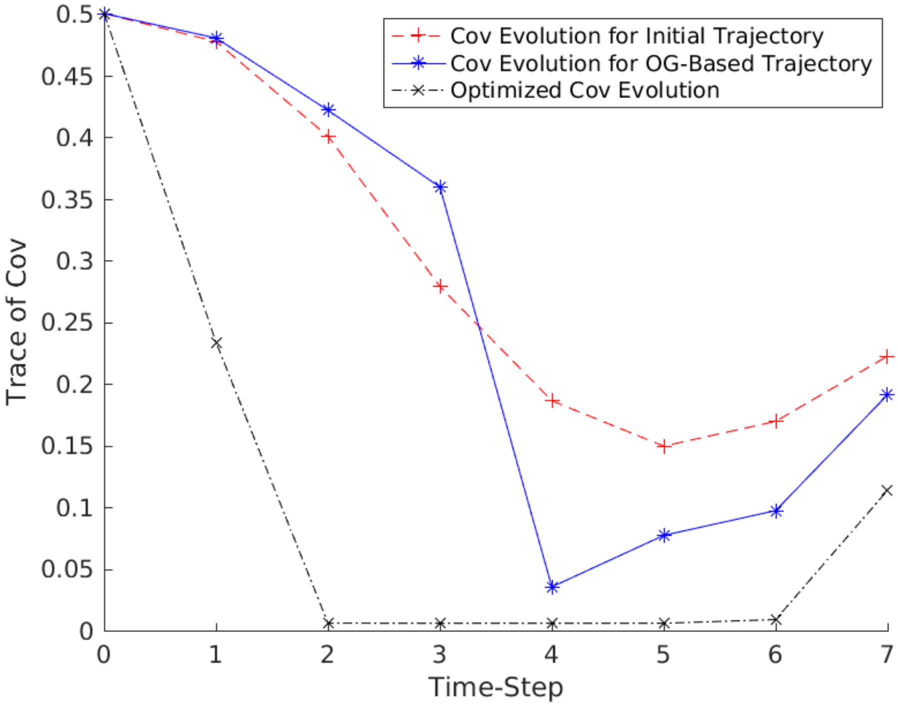


Figure 10.6: Bearing observation model. Evolution of the trace of the covariance along the trajectory for the initial trajectory, with optimization based on the OG measure, and optimization based on the covariance measure of the trajectories in Fig. 10.5.

This is due to the fact that in the Bayesian filtering, the latest observations (which arise from the same region in the state space) carry a higher weight than the prior history. As a result, in comparing the covariance evolutions, the variations in the behavior along the entire trajectory is of concern since a highly certain trajectory can lead to safer navigation, particularly, in a complex environment with obstacles, banned areas or multiple agents.

*Remark:* Table 10.1 summarizes some of the results regarding the optimization problems solved for Figs. 10.1, 10.5, and 10.3, such as the initial and final values (after reaching a local minimum) of the cost and constraint satisfaction, and the simulation time to solve the optimization problem. As shown in this table, the optimization significantly reduces the cost in each scenario.

*Remark:* Finally, note that the simulation times to solve the optimization problem for all cases are of the same order, which stems from the fact that the computation complexity of both the problems (17) and (18) is $O(Kn_x^3)$ [196].

## 10.5 Detailed Analysis Using Various Measures and Parameters

In this section, first, we provide results of our simulations for planning problem of (17) using different measures of the OG as the cost function, and show that different measures of the OG can provide inconsistent results for a single observation model.

Then, we perform some simulations to show the sensitivity of the covariance-based trajectories on the initial uncertainty, process and observation noise uncertainties. In particular, we simulate the problem of (18) which utilizes the covariance evolution and by depicting the resulting trajectories, we show qualitatively that by changing any of the noise parameters, the optimal trajectories can change significantly. This is particularity important, because, as we showed in Section 10.4, any qualitative change of the trajectory can result in a dramatic change in the quantitative performance of the covariance evolution. We also showed in Section 10.4 that two trajectories that resemble to each other qualitatively, can be very different in terms of their quantitative covariance evolution performance. As a result, the OG-based trajectories, which are insensitive to any noise parameters of the problem, can often show poor performance. Other application areas to the stochastic analysis can be in [219, 220].

Figures 10.7, 10.8, and 10.9 show the results of optimization problem (17) for all the measures of the OG that are indicated in Section 10.2.1 for range-only, bearing-only, and range-squared-only observation models, respectively. As it is shown in these figures, different measures of the observation Gramian can result in very different trajectories. For instance, Figs. 10.7a, 10.7b, 10.7d, and 10.7f suggest maneuvering towards the observation source in this situation, whereas Fig. 10.7c is indifferent to the observation source and Fig. 10.7e suggests a more complicated maneuver. Simulations for other observation models also indicate similar conjecture. Nevertheless, it is very important to note that depending on the noise parameters of the problem, any of these results can become misleading, undesirable, or even contradicting with the optimal maneuvers according to the covariance measure's development along the trajectory.

Figure 10.10 shows the simulation results of problem (18) for all three models and two different observation noise covariances. Moreover, in this figure, the initial noise is correlated in different directions. As it is seen in this figure, the optimal trajectory can change significantly based on the observation noise intensity. This is more obvious for the range-squared observation model, where the observation covariance decreases from 0.01 in Fig. 10.10c to 0.00001 in Fig. 10.10f.

Similarly, Fig. 10.11 shows the results of simulations of problem (18) for all observation models, however, in this case, the observation covariance is fixed and the intensity of the initial and process noise covariances in different directions is changed to show the effect of different levels of noise in different directions. In particular, for the set of figures in the first row, the process and initial noise are more intense in $x$ direction; for the second row, the intensity of the process noise is more in $y$ direction, while the intensity of initial noise is more in $x$ direction; and, for the third raw, the noise intensities are the same in both directions. Once again, unlike these figures, the OG-based optimization would be insensitive to any of these changes, which can lead to poor performances.

Last, Tables 10.2, 10.3, 10.4, 10.6, and 10.5 show the optimization problems' results of Figs. 10.7, 10.8, 10.9, 10.10, and 10.11.

*Remark:* Note that we used very simple models and scenarios to focus specifically on the effect of changing the measure and avoid complications caused by complex models, environments, or even higher dimensionality. Nevertheless, our results show that even in very simple situations the quantitative performance of the system can change dramatically based on the qualitative changes of the trajectory, and only the theoretically sound and proven method of planning based on problem (18) should be utilized in these problems for optimal, desirable and safe performance of the system.

Table 10.2: The OG-based planning (range-only): Initial and final costs and constraint satisfaction for simulations of Fig. 10.7.

| | Fig. 10.7a | Fig. 10.7b | Fig. 10.7c | Fig. 10.7d | Fig. 10.7e | Fig. 10.7f |
|---|---|---|---|---|---|---|
| Initial Cost | 0.14553 | 1.01874 | -7 | 0.84689 | 0.17184 | 4.9282 |
| Final Cost | 0.08163 | 0.57142 | -7 | 0.28571 | 0.14792 | 1 |
| Initial Constraint | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 |
| Final Constraint | 1.8e-07 | 4.3e-11 | -0.0997 | -0.0022 | 4.8e-12 | -0.0230 |

Table 10.3: The OG-based planning (bearing-only): Initial and final costs and constraint satisfaction for simulations of Fig. 10.8.

| | Fig. 10.8a | Fig. 10.8b | Fig. 10.8c | Fig. 10.8d | Fig. 10.8e | Fig. 10.8f |
|---|---|---|---|---|---|---|
| Initial Cost | 1.15314 | 2.98925 | -2.5922 | 2.53422 | 0.45502 | 5.5693 |
| Final Cost | 2.6e-10 | 1.2e-10 | -4e+10 | 1.7e-10 | 7.1e-13 | 1 |
| Initial Constraint | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 |
| Final Constraint | -4.6e-05 | -0.0282 | -0.0008 | -0.0073 | 9.5e-11 | -0.0085 |

Table 10.4: The OG-based planning (range-squared-only): Initial and final costs and constraint satisfaction for simulations of Fig. 10.9.

| | Fig. 10.9a | Fig. 10.9b | Fig. 10.9c | Fig. 10.9d | Fig. 10.9e | Fig. 10.9f |
|---|---|---|---|---|---|---|
| Initial Cost | 0.00099 | 0.08823 | -88.39 | 0.07490 | 0.01332 | 5.6216 |
| Final Cost | 7.1e-05 | 0.02040 | -374.24 | 0.01315 | 0.00289 | 1 |
| Initial Constraint | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 |
| Final Constraint | 1.3e-06 | 4.3e-08 | 2.9e-10 | 2.1e-08 | 2.8e-07 | -0.0006 |

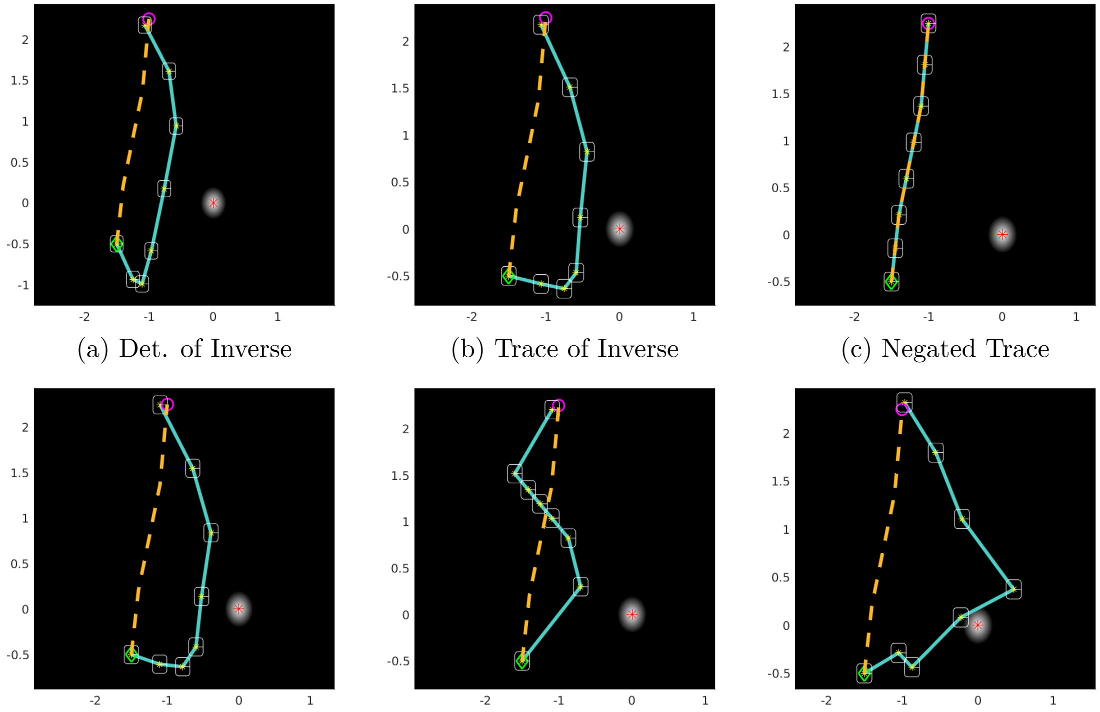
## 10.6 Calculations of the Observability Gramian

First let us provide some calculations related to $2 \times 2$ matrices which are used to calculate the specific formulas provided in the chapter. Let us assume matrix $\mathbf{A}$ is given as follows:

$$\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Table 10.5: The covariance-based planning: Initial and final costs and constraint satisfaction for simulations of Fig. 10.10.

| | Fig. 10.10a | Fig. 10.10b | Fig. 10.10c | Fig. 10.10d | Fig. 10.10b | Fig. 10.10c |
|---|---|---|---|---|---|---|
| Initial Cost | 0.07543 | 0.08893 | 0.04441 | 0.03409 | 0.03516 | 0.03395 |
| Final Cost | 0.06088 | 0.02678 | 0.02622 | 0.01230 | 0.01268 | 0.01231 |
| Initial Constraint | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 |
| Final Constraint | 2.6e-06 | -2.0e-14 | 8.2e-10 | 3.0e-09 | 1.2e-09 | 6.5e-08 |



(a) Det. of Inverse

(b) Trace of Inverse

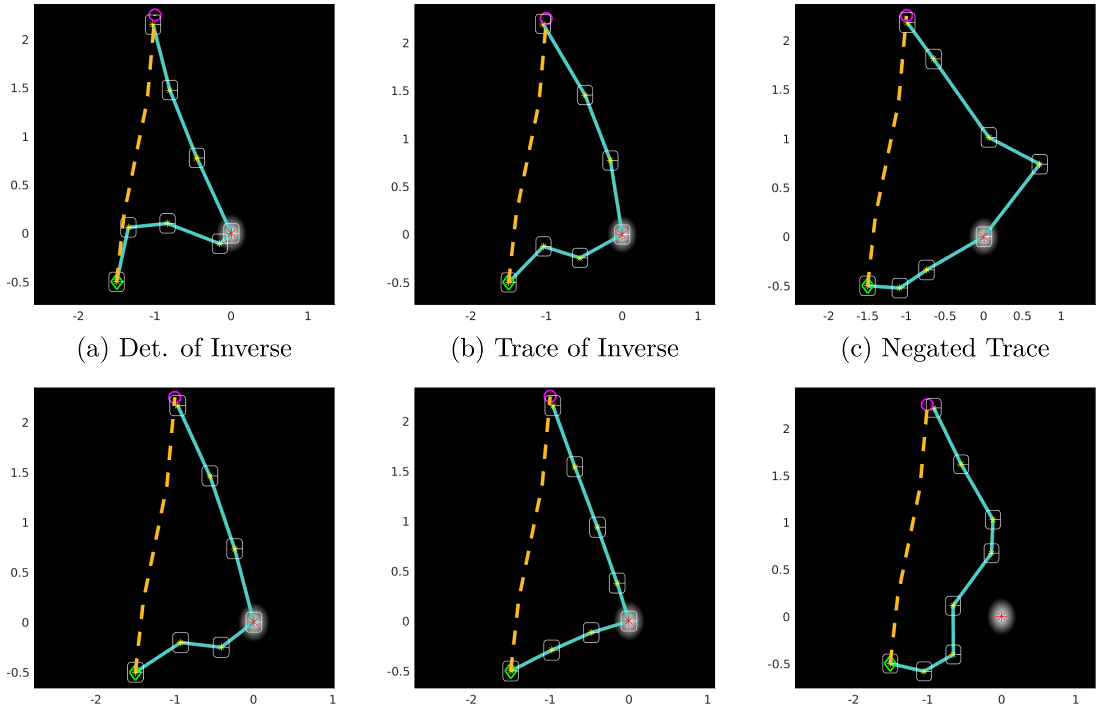
(c) Negated Trace

(d) Inverse of Min E.V.

(e) Inverse of Max E.V.

(f) Condition Number

Figure 10.7: The OG-based planning (range-only): Simulation results for range-only observation model, where the caption indicates the measure of Gramian used in the planning problem (17). The information sources are located at the centers of the light areas. The dashed orange line represents the initial trajectory, while the solid cyan line shows the optimized trajectory.



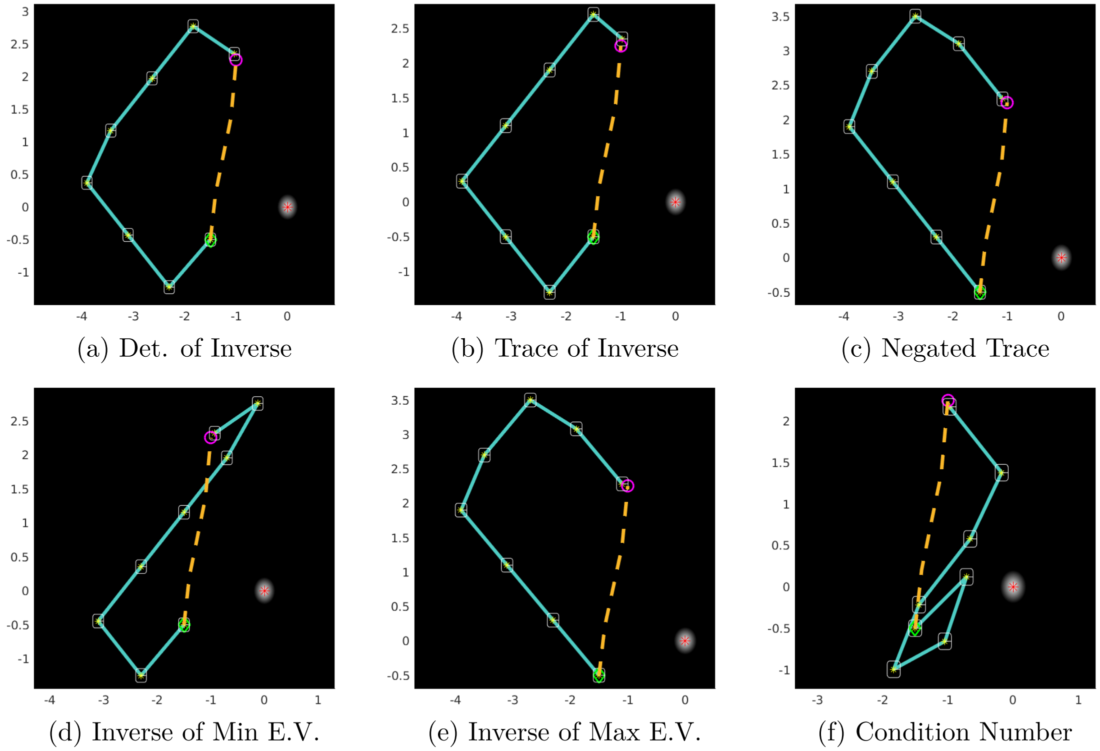
(a) Det. of Inverse (b) Trace of Inverse (c) Negated Trace

(d) Inverse of Min E.V. (e) Inverse of Max E.V. (f) Condition Number

Figure 10.8: The OG-based planning (bearing-only): Simulation results for bearing-only observation model, where the caption indicates the measure of Gramian used in the planning problem (17).The information sources are located at the centers of the light areas. The dashed orange line represents the initial trajectory, while the solid cyan line shows the optimized trajectory.

Table 10.6: The covariance-based planning: Initial and final costs and constraint satisfaction for simulations of Fig. 10.11.

| | Fig. 10.11a | Fig. 10.11b | Fig. 10.11c | Fig. 10.11d | Fig. 10.11e | Fig. 10.11f | Fig. 10.11g | Fig. 10.11h | Fig. 10.11i |
|---|---|---|---|---|---|---|---|---|---|
| Initial Cost | 0.08307 | 0.11604 | 0.05603 | 0.08279 | 0.11366 | 0.04595 | 0.07431 | 0.09116 | 0.04479 |
| Final Cost | 0.07040 | 0.03591 | 0.03410 | 0.06840 | 0.03937 | 0.02927 | 0.06145 | 0.02694 | 0.02746 |
| Initial Constraint | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 | -0.0997 |
| Final Constraint | 1.6e-05 | 4.5e-07 | 4.1e-09 | 2.6e-05 | 1.4e-09 | 3.8e-09 | 4.3e-08 | 1.5e-10 | 1.3e-08 |



(a) Det. of Inverse

(b) Trace of Inverse

(c) Negated Trace

(d) Inverse of Min E.V.

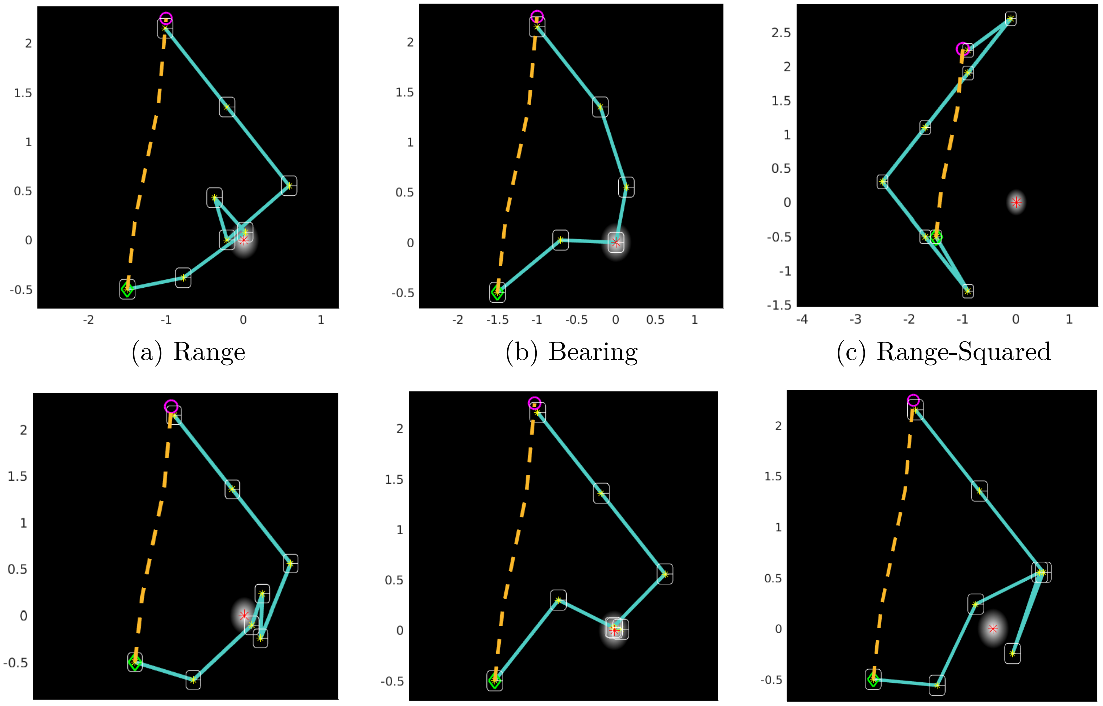
(e) Inverse of Max E.V.

(f) Condition Number

Figure 10.9: The OG-based planning (range-squared-only): Simulation results for range-squared-only observation model, where the caption indicates the measure of Gramian used in the planning problem (17). The information sources are located at the centers of the light areas. The dashed orange line represents the initial trajectory, while the solid cyan line shows the optimized trajectory.

Then, the determinant of $\mathbf{A}$ is $\det(\mathbf{A}) = ad - bc$, and its trace equals $\text{tr}(\mathbf{A}) = a + d$. Moreover, the eigenvalues of $\mathbf{A}$, denoted by $\lambda_1$ and $\lambda_2$ can be calculated using the equations $\mathbf{A}\mathbf{x} = \lambda_i \mathbf{x}$, for $i = 1, 2$. As a result, it can be easily proven that the eigenvalues of $\mathbf{A}$ are the roots of the characteristic equation of $\lambda^2 - (a+d)\lambda + ad - bc = 0$, and therefore are:

$$\lambda_1 = \frac{1}{2}(a+d) + \frac{1}{2}\sqrt{(a+d)^2 - 4(ad-bc)},$$
$$\lambda_2 = \frac{1}{2}(a+d) - \frac{1}{2}\sqrt{(a+d)^2 - 4(ad-bc)},$$



(a) Range

(b) Bearing

(c) Range-Squared

(d) Range

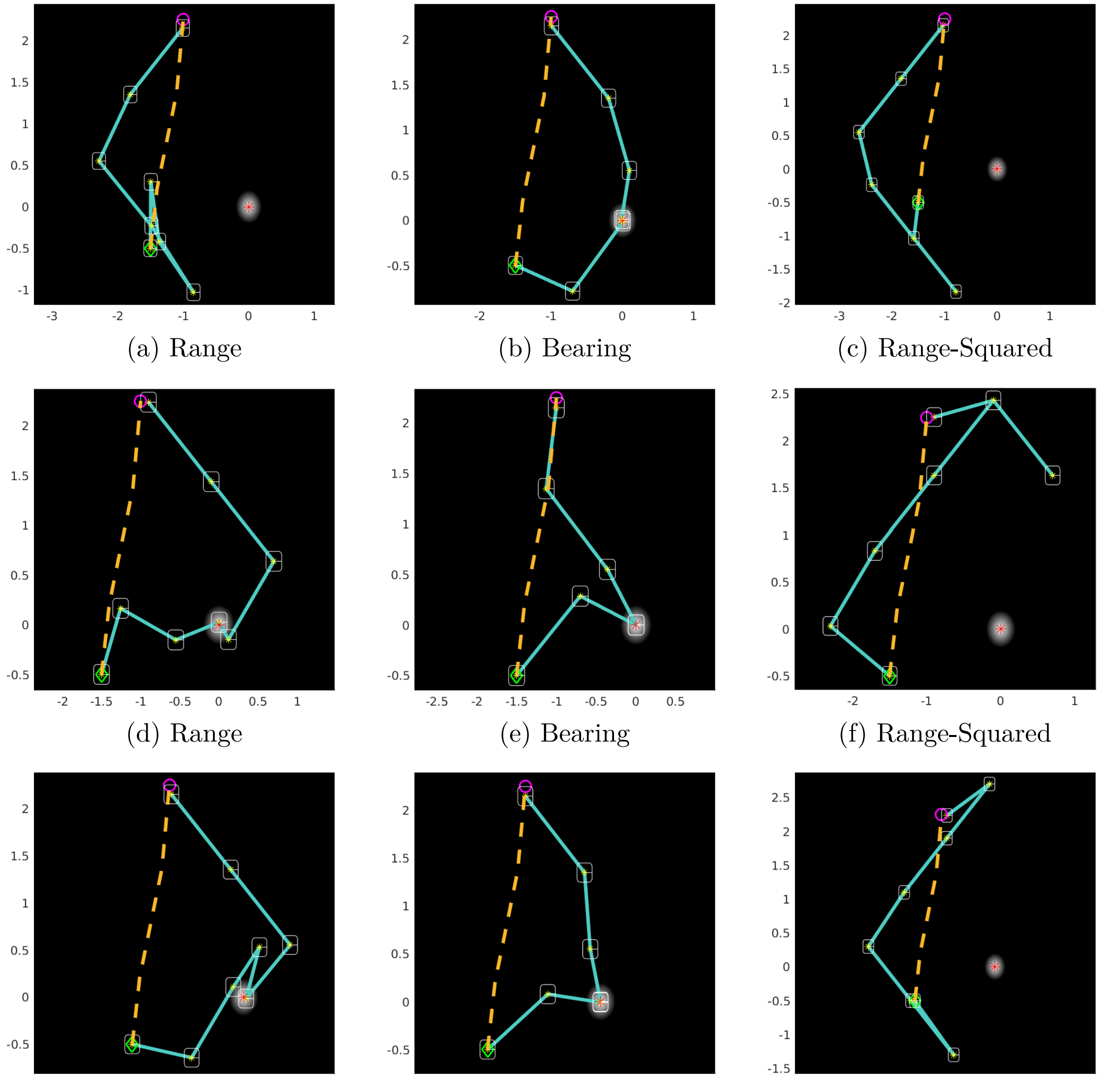
(e) Bearing

(f) Range-Squared

Figure 10.10: Covariance-based planning (correlated initial noise, effect of observation noise covariance): Simulation results for all three observation models and two different observation noise covariances, and a correlated initial covariance, where the caption indicates the observation model used in the planning problem (18). For both rows, $\mathbf{\Sigma}_{\mathbf{x}_0} = (0.005, 0.001; 0.001, 0.005)$, and $\mathbf{\Sigma}_{\mathbf{w}_t} = \text{diag}(0.001, 0.001)$. For the first row, $\Sigma_{v_t} = 0.01$ and for the second row, $\Sigma_{v_t} = 0.00001$. The information sources are located in the centers of the light areas. The dashed orange line represents the initial trajectory, while the solid cyan line shows the optimized trajectory.

where $\lambda_1 > \lambda_2$. Therefore, the condition number of $\mathbf{A}$, is given as follows:

$$\begin{aligned}\kappa(\mathbf{A}) &= \frac{\lambda_{\max}}{\lambda_{\min}} \\ &= \frac{(a+d) + \sqrt{(a+d)^2 - 4(ad-bc)}}{(a+d) - \sqrt{(a+d)^2 - 4(ad-bc)}},\end{aligned}$$



(a) Range (b) Bearing (c) Range-Squared

(d) Range (e) Bearing (f) Range-Squared

(g) Range (h) Bearing (i) Range-Squared

Figure 10.11: Covariance-based planning (effect of initial and process noise covariances): Simulation results for all three observation models and various different initial and process noise covariances, where the caption indicates the observation model used in the planning problem (18). For the first row, $\mathbf{\Sigma}_{\mathbf{x}_0} = \text{diag}(0.005, 0.002)$, $\mathbf{\Sigma}_{\mathbf{w}_t} = \text{diag}(0.003, 0.001)$. For the second row, $\mathbf{\Sigma}_{\mathbf{x}_0} = \text{diag}(0.005, 0.001)$, $\mathbf{\Sigma}_{\mathbf{w}_t} = \text{diag}(0.001, 0.003)$. For the third row, $\mathbf{\Sigma}_{\mathbf{x}_0} = \text{diag}(0.005, 0.005)$, $\mathbf{\Sigma}_{\mathbf{w}_t} = \text{diag}(0.005, 0.005)$. For the entire simulation, $\Sigma_{v_t} = 0.01$. The information sources are located in the centers of the light areas. The dashed orange line represents the initial trajectory, while the solid cyan line shows the optimized trajectory.

where $\kappa(\mathbf{A}) \geq 1$. Therefore, in order to make $\kappa(\mathbf{A}) = 1$, we must have $(a+d)^2 = 4(ad{-}bc)$ or for 2 by 2 matrix, $(\mathrm{tr}(\mathbf{A})/2)^2 = \det(\mathbf{A})$. On the other hand, if $\det(\mathbf{A}) > 0$ the inverse of $\mathbf{A}$ is given as follows:

$$\mathbf{A}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

Therefore, for a 2 by 2 matrix, $\det(\mathbf{A}^{-1}) = 1/\det(\mathbf{A}) = 1/(ad-bc)$ and $\mathrm{tr}(\mathbf{A}^{-1}) = (a+d)/(ad-bc) = \mathrm{tr}(\mathbf{A})/\det(\mathbf{A})$. Next, we provide the calculations related to the specific formulas provided in the chapter.

### *10.6.1 Range-Only Model*

*The OG calculations:* Let us use the definition of the OG in equation (10.3) to calculate the OG for this system model as:

$$\begin{aligned}
\mathbf{Q}^p_{K+1} &= \sum_{t=0}^{K} \tilde{\mathbf{A}}_t^T \mathbf{H}_t^T \mathbf{H}_t \tilde{\mathbf{A}}_t \\
&= \sum_{t=0}^{K} (\frac{x_t}{r_t}, \frac{y_t}{r_t})^T (\frac{x_t}{r_t}, \frac{y_t}{r_t}) \\
&= \sum_{t=0}^{K} \begin{pmatrix} \frac{x_t^2}{r_t^2} & \frac{x_t y_t}{r_t^2} \\ \frac{x_t y_t}{r_t^2} & \frac{y_t^2}{r_t^2} \end{pmatrix}.
\end{aligned}$$

*Calculations of the trace:*

$$\begin{aligned}
\mathrm{tr}(\mathbf{Q}^p_{K+1}) &= \mathrm{tr}(\sum_{t=0}^{K} \begin{pmatrix} \frac{x_t^2}{r_t^2} & \frac{x_t y_t}{r_t^2} \\ \frac{x_t y_t}{r_t^2} & \frac{y_t^2}{r_t^2} \end{pmatrix}) \\
&= \sum_{t=0}^{K} \mathrm{tr}(\begin{pmatrix} \frac{x_t^2}{r_t^2} & \frac{x_t y_t}{r_t^2} \\ \frac{x_t y_t}{r_t^2} & \frac{y_t^2}{r_t^2} \end{pmatrix})
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{t=0}^{K} (\frac{x_t^2}{r_t^2} + \frac{y_t^2}{r_t^2}) \\
&= K + 1.
\end{aligned}
$$

### *10.6.2 Bearing-Only Model*

*The OG calculations:* Let us use the definition of the OG in equation (10.3) to calculate the OG for this system model as:

$$
\begin{aligned}
\mathbf{Q}_{K+1}^{p} &= \sum_{t=0}^{K} \tilde{\mathbf{A}}_t^T \mathbf{H}_t^T \mathbf{H}_t \tilde{\mathbf{A}}_t \\
&= \sum_{t=0}^{K} (-\frac{y_t}{r_t^2}, \frac{x_t}{r_t^2})^T (-\frac{y_t}{r_t^2}, \frac{x_t}{r_t^2}) \\
&= \sum_{t=0}^{K} \begin{pmatrix} \frac{y_t^2}{r_t^4} & \frac{-x_t y_t}{r_t^4} \\ \frac{-x_t y_t}{r_t^4} & \frac{x_t^2}{r_t^4} \end{pmatrix}.
\end{aligned}
$$

Therefore, the trace of the OG can be calculated as:

$$
\begin{aligned}
\mathrm{tr}(\mathbf{Q}_{K+1}^{p}) &= \mathrm{tr}(\sum_{t=0}^{K} \begin{pmatrix} \frac{y_t^2}{r_t^4} & \frac{-x_t y_t}{r_t^4} \\ \frac{-x_t y_t}{r_t^4} & \frac{x_t^2}{r_t^4} \end{pmatrix}) \\
&= \sum_{t=0}^{K} \mathrm{tr}(\begin{pmatrix} \frac{y_t^2}{r_t^4} & \frac{-x_t y_t}{r_t^4} \\ \frac{-x_t y_t}{r_t^4} & \frac{x_t^2}{r_t^4} \end{pmatrix}) \\
&= \sum_{t=0}^{K} (\frac{x_t^2}{r_t^4} + \frac{y_t^2}{r_t^4}) \\
&= \sum_{t=0}^{K} \frac{1}{r_t^2},
\end{aligned}
$$

### 10.6.3 Range-Squared-Only

*The OG calculations:* Let us calculate the OG for this system as follows:

$$\begin{aligned}
\mathbf{Q}^p_{K+1} &= \sum_{t=0}^{K} \tilde{\mathbf{A}}_t^T \mathbf{H}_t^T \mathbf{H}_t \tilde{\mathbf{A}}_t \\
&= \sum_{t=0}^{K} (x_t, y_t)^T (x_t, y_t) \\
&= \sum_{t=0}^{K} \begin{pmatrix} x_t^2 & x_t y_t \\ x_t y_t & y_t^2 \end{pmatrix}.
\end{aligned}$$

Therefore, the trace of the OG can be calculated as:

$$\begin{aligned}
\operatorname{tr}(\mathbf{Q}^p_{K+1}) &= \operatorname{tr}(\sum_{t=0}^{K} \begin{pmatrix} x_t^2 & x_t y_t \\ x_t y_t & y_t^2 \end{pmatrix}) \\
&= \sum_{t=0}^{K} \operatorname{tr}(\begin{pmatrix} x_t^2 & x_t y_t \\ x_t y_t & y_t^2 \end{pmatrix}) \\
&= \sum_{t=0}^{K} (x_t^2 + y_t^2) \\
&= \sum_{t=0}^{K} r_t^2.
\end{aligned}$$

## 10.7 Calculations of the Covariance Evolution

### 10.7.1 Range-Only Model

First, we calculate $\mathbf{P}_1^-$, which will be used for the other examples, as well:

$$\begin{aligned}
\mathbf{P}_1^- &= \mathbf{A}_0 \mathbf{P}_0^+ (\mathbf{A}_0)^T + \mathbf{G}_0 \mathbf{\Sigma}_{\mathbf{w}_0} (\mathbf{G}_0)^T \\
&= \mathbf{\Sigma}_{\mathbf{x}_0} + \mathbf{\Sigma}_{\mathbf{w}_0} \\
&= \operatorname{diag}(\sigma_0^x + \sigma_{\mathbf{w}}^x, \sigma_0^y + \sigma_{\mathbf{w}}^y). \qquad (10.21)
\end{aligned}$$

Then, we have:

$$\begin{aligned}
\mathbf{S}_1 &= \mathbf{H}_1\mathbf{P}_1^-(\mathbf{H}_1)^T + \mathbf{M}_1\mathbf{\Sigma}_{\mathbf{v}_1}(\mathbf{M}_1)^T \\
&= (\frac{x_t}{r_t}, \frac{y_t}{r_t}) \begin{pmatrix} \sigma_0^x + \sigma_{\mathbf{w}}^x & 0 \\ 0 & \sigma_0^y + \sigma_{\mathbf{w}}^y \end{pmatrix} (\frac{x_t}{r_t}, \frac{y_t}{r_t})^T + \sigma_\nu \\
&= \left((\sigma_0^x + \sigma_{\mathbf{w}}^x)\tfrac{x_t}{r_t}, (\sigma_0^y + \sigma_{\mathbf{w}}^y)\tfrac{y_t}{r_t}\right)(\frac{x_t}{r_t}, \frac{y_t}{r_t})^T + \sigma_\nu \\
&= (\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{x_t^2}{r_t^2} + (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{y_t^2}{r_t^2} + \sigma_\nu,
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{K}_1 &= \mathbf{P}_1^-(\mathbf{H}_1)^T S_1^{-1} \\
&= S_1^{-1} \begin{pmatrix} \sigma_0^x + \sigma_{\mathbf{w}}^x & 0 \\ 0 & \sigma_0^y + \sigma_{\mathbf{w}}^y \end{pmatrix} (\frac{x_t}{r_t}, \frac{y_t}{r_t})^T \\
&= S_1^{-1} \left((\sigma_0^x + \sigma_{\mathbf{w}}^x)\tfrac{x_t}{r_t}, (\sigma_0^y + \sigma_{\mathbf{w}}^y)\tfrac{y_t}{r_t}\right)^T,
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{P}_1^+ &= (\mathbf{I}_2 - \mathbf{K}_1\mathbf{H}_1)\mathbf{P}_1^- \\
&= [\mathbf{I}_2 - S_1^{-1} \begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{x_t}{r_t} \\ (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{y_t}{r_t} \end{pmatrix} (\frac{x_t}{r_t}, \frac{y_t}{r_t})]\mathbf{P}_1^- \\
&= (\mathbf{I}_2 - S_1^{-1} \begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{x_t^2}{r_t^2} & (\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{x_t y_t}{r_t^2} \\ (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{x_t y_t}{r_t^2} & (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{y_t^2}{r_t^2} \end{pmatrix})\mathbf{P}_1^- \\
&= \begin{pmatrix} 1 - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)}{S_1}\frac{x_t^2}{r_t^2} & -\frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)}{S_1}\frac{x_t y_t}{r_t^2} \\ -\frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t y_t}{r_t^2} & 1 - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)}{S_1}\frac{y_t^2}{r_t^2} \end{pmatrix}
\end{aligned}$$

$$\times \begin{pmatrix} \sigma_0^x + \sigma_{\mathbf{w}}^x & 0 \\ 0 & \sigma_0^y + \sigma_{\mathbf{w}}^y \end{pmatrix}$$

$$= \begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)^2}{S_1}\frac{x_t^2}{r_t^2} & -\frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)(\sigma_0^y+\sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t y_t}{r_t^2} \\ -\frac{(\sigma_0^x+\sigma_{\mathbf{w}}^x)(\sigma_0^y+\sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t y_t}{r_t^2} & (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y+\sigma_{\mathbf{w}}^y)^2}{S_1}\frac{y_t^2}{r_t^2} \end{pmatrix}.$$

Therefore, the trace of the updated covariance at time-step one is:

$$\begin{aligned}
\mathrm{tr}(\mathbf{P}_1^+) &= (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)^2}{S_1}\frac{x_t^2}{r_t^2} \\
&\quad + (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)^2}{S_1}\frac{y_t^2}{r_t^2} \\
&= (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)^2 x_t^2}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu r_t^2} \\
&\quad + (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)^2 y_t^2}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu r_t^2} \\
&= \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + (\sigma_0^x + \sigma_{\mathbf{w}}^x)\sigma_\nu r_t^2}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu r_t^2} \\
&\quad + \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu r_t^2}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu r_t^2} \\
&= \frac{[(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y) + (\sigma_0^x + \sigma_{\mathbf{w}}^x + \sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu] r_t^2}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu r_t^2} \\
&= \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y) + (\sigma_0^x + \sigma_{\mathbf{w}}^x + \sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{x_t^2}{r_t^2} + (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{y_t^2}{r_t^2} + \sigma_\nu}.
\end{aligned}$$

### *10.7.2 Bearing-Only Model*

Similar to the range-only example, the $\mathbf{P}_1^-$ is given as equation (10.21). The calculation of $\mathbf{S}_1$ is:

$$\begin{aligned}
\mathbf{S}_1 &= \mathbf{H}_1\mathbf{P}_1^-(\mathbf{H}_1)^T + \mathbf{M}_1\mathbf{\Sigma}_{\mathbf{v}_1}(\mathbf{M}_1)^T \\
&= (\frac{-y_t}{r_t^2}, \frac{x_t}{r_t^2}) \begin{pmatrix} \sigma_0^x + \sigma_{\mathbf{w}}^x & 0 \\ 0 & \sigma_0^y + \sigma_{\mathbf{w}}^y \end{pmatrix} (\frac{-y_t}{r_t^2}, \frac{x_t}{r_t^2})^T + \sigma_\nu
\end{aligned}$$

$$
\begin{aligned}
&= \left( (\sigma_0^x + \sigma_{\mathbf{w}}^x)\tfrac{-y_t}{r_t^2}, (\sigma_0^y + \sigma_{\mathbf{w}}^y)\tfrac{x_t}{r_t^2} \right) (\frac{-y_t}{r_t^2}, \frac{x_t}{r_t^2})^T + \sigma_\nu \\
&= (\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{y_t^2}{r_t^4} + (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{x_t^2}{r_t^4} + \sigma_\nu .
\end{aligned}
$$

The Kalman gain is calculated as:

$$
\begin{aligned}
\mathbf{K}_1 &= \mathbf{P}_1^-(\mathbf{H}_1)^T S_1^{-1} \\
&= S_1^{-1} \begin{pmatrix} \sigma_0^x + \sigma_{\mathbf{w}}^x & 0 \\ 0 & \sigma_0^y + \sigma_{\mathbf{w}}^y \end{pmatrix} (\frac{-y_t}{r_t^2}, \frac{x_t}{r_t^2})^T \\
&= S_1^{-1} \left( (\sigma_0^x + \sigma_{\mathbf{w}}^x)\tfrac{-y_t}{r_t^2}, (\sigma_0^y + \sigma_{\mathbf{w}}^y)\tfrac{x_t}{r_t^2} \right)^T .
\end{aligned}
$$

The updated covariance is:

$$
\begin{aligned}
\mathbf{P}_1^+ &= (\mathbf{I}_2 - \mathbf{K}_1\mathbf{H}_1)\mathbf{P}_1^- \\
&= [\mathbf{I}_2 - S_1^{-1} \begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{-y_t}{r_t^2} \\ (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{x_t}{r_t^2} \end{pmatrix} (\frac{-y_t}{r_t^2}, \frac{x_t}{r_t^2})]\mathbf{P}_1^- \\
&= (\mathbf{I}_2 - S_1^{-1} \begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{y_t^2}{r_t^4} & (\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{-x_t y_t}{r_t^4} \\ (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{-x_t y_t}{r_t^4} & (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{x_t^2}{r_t^4} \end{pmatrix})\mathbf{P}_1^- \\
&= \begin{pmatrix} 1 - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)}{S_1}\frac{y_t^2}{r_t^4} & \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)}{S_1}\frac{x_t y_t}{r_t^4} \\ \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t y_t}{r_t^4} & 1 - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t^2}{r_t^4} \end{pmatrix} \\
&\quad \times \begin{pmatrix} \sigma_0^x + \sigma_{\mathbf{w}}^x & 0 \\ 0 & \sigma_0^y + \sigma_{\mathbf{w}}^y \end{pmatrix} \\
&= \begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)^2}{S_1}\frac{y_t^2}{r_t^4} & \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t y_t}{r_t^4} \\ \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y)}{S_1}\frac{x_t y_t}{r_t^4} & (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)^2}{S_1}\frac{x_t^2}{r_t^4} \end{pmatrix} .
\end{aligned}
$$

Lastly, the trace of the updated covariance at time-step one is:

$$\begin{aligned}
\mathrm{tr}(\mathbf{P}_1^+) &= (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)^2}{S_1}\frac{y_t^2}{r_t^4} \\
&+ (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)^2}{S_1}\frac{x_t^2}{r_t^4} \\
&= (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)^2 y_t^2}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)y_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)x_t^2 + \sigma_\nu r_t^4} \\
&+ (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)^2 x_t^2}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)y_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)x_t^2 + \sigma_\nu r_t^4} \\
&= \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y)x_t^2 + (\sigma_0^x + \sigma_{\mathbf{w}}^x)\sigma_\nu r_t^4}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)y_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)x_t^2 + \sigma_\nu r_t^4} \\
&+ \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)(\sigma_0^x + \sigma_{\mathbf{w}}^x)y_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu r_t^4}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)y_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)x_t^2 + \sigma_\nu r_t^4} \\
&= \frac{[(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y) + (\sigma_0^x + \sigma_{\mathbf{w}}^x + \sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu]r_t^4}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)y_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)x_t^2 + \sigma_\nu r_t^4} \\
&= \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y) + (\sigma_0^x + \sigma_{\mathbf{w}}^x + \sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)\frac{y_t^2}{r_t^4} + (\sigma_0^y + \sigma_{\mathbf{w}}^y)\frac{x_t^2}{r_t^4} + \sigma_\nu}.
\end{aligned}$$

### *10.7.3 Range-Squared-Only*

Similar to the range-only example, the $\mathbf{P}_1^-$ is given as equation (10.21). The calculation of $\mathbf{S}_1$ is given as:

$$\begin{aligned}
\mathbf{S}_1 &= \mathbf{H}_1\mathbf{P}_1^-(\mathbf{H}_1)^T + \mathbf{M}_1\mathbf{\Sigma}_{\mathbf{v}_1}(\mathbf{M}_1)^T \\
&= (x_t, y_t)\begin{pmatrix} \sigma_0^x + \sigma_{\mathbf{w}}^x & 0 \\ 0 & \sigma_0^y + \sigma_{\mathbf{w}}^y \end{pmatrix}(x_t, y_t)^T + \sigma_\nu \\
&= \Big((\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t, (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t\Big)(x_t, y_t)^T + \sigma_\nu \\
&= (\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu.
\end{aligned}$$

The Kalman gain is calculated as follows:

$$\begin{aligned}
\mathbf{K}_1 &= \mathbf{P}_1^-(\mathbf{H}_1)^T S_1^{-1} \\
&= S_1^{-1}\begin{pmatrix} \sigma_0^x + \sigma_{\mathbf{w}}^x & 0 \\ 0 & \sigma_0^y + \sigma_{\mathbf{w}}^y \end{pmatrix}(x_t, y_t)^T \\
&= S_1^{-1}\Big((\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t, (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t\Big)^T.
\end{aligned}$$

The updated covariance is:

$$\begin{aligned}
\mathbf{P}_1^+ &= (\mathbf{I}_2 - \mathbf{K}_1\mathbf{H}_1)\mathbf{P}_1^- \\
&= [\mathbf{I}_2 - S_1^{-1}\begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t \\ (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t \end{pmatrix}(x_t, y_t)]\mathbf{P}_1^- \\
&= (\mathbf{I}_2 - S_1^{-1}\begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 & (\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t y_t \\ (\sigma_0^y + \sigma_{\mathbf{w}}^y)x_t y_t & (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 \end{pmatrix})\mathbf{P}_1^- \\
&= \begin{pmatrix} 1 - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)}{S_1}x_t^2 & -\frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)}{S_1}x_t y_t \\ -\frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)}{S_1}x_t y_t & 1 - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)}{S_1}y_t^2 \end{pmatrix} \\
&\quad \times \begin{pmatrix} \sigma_0^x + \sigma_{\mathbf{w}}^x & 0 \\ 0 & \sigma_0^y + \sigma_{\mathbf{w}}^y \end{pmatrix} \\
&= \begin{pmatrix} (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)^2}{S_1}x_t^2 & -\frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y)}{S_1}x_t y_t \\ -\frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y)}{S_1}x_t y_t & (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)^2}{S_1}y_t^2 \end{pmatrix}.
\end{aligned}$$

Last, the trace of the updated covariance at time-step one is:

$$\begin{aligned}
\mathrm{tr}(\mathbf{P}_1^+) &= (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)^2}{S_1}x_t^2 \\
&\quad + (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)^2}{S_1}y_t^2
\end{aligned}$$

$$
\begin{aligned}
&= (\sigma_0^x + \sigma_{\mathbf{w}}^x) - \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)^2 x_t^2}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu} \\
&\quad + (\sigma_0^y + \sigma_{\mathbf{w}}^y) - \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)^2 y_t^2}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu} \\
&= \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + (\sigma_0^x + \sigma_{\mathbf{w}}^x)\sigma_\nu}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu} \\
&\quad + \frac{(\sigma_0^y + \sigma_{\mathbf{w}}^y)(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu} \\
&= \frac{(\sigma_0^x + \sigma_{\mathbf{w}}^x)(\sigma_0^y + \sigma_{\mathbf{w}}^y)r_t^2 + (\sigma_0^x + \sigma_{\mathbf{w}}^x + \sigma_0^y + \sigma_{\mathbf{w}}^y)\sigma_\nu}{(\sigma_0^x + \sigma_{\mathbf{w}}^x)x_t^2 + (\sigma_0^y + \sigma_{\mathbf{w}}^y)y_t^2 + \sigma_\nu}.
\end{aligned}
$$

## 10.8 Conclusion

In this chapter, we have investigated a well-known heuristic employing the observability Gramian in planning under observation uncertainty. We have utilized two common observation models and shown that, in general, the observability Gramian (and the closely-related standard Fisher information matrix) fail to capture many aspects of the models including the initial, process, and observation uncertainties. As a result, based on changes in those models, we showed using analytic and numerical examples that planning based on the observability Gramian can provide trajectories that are very different in terms of the estimation performance from the optimal plans based on the estimation covariance of the problem.

# 11. CONVEX BELIEF SPACE PLANNING UNDER NON-GAUSSIAN UNCERTAINTY

In this chapter, we provide an alternative belief space planning method than the previous chapters, based on our prior work.

Part (I) of this Dissertation provides a rigorous method of analyzing a stochastic optimal control problem for nonlinear systems with additive Gaussian perturbations. In practice, there are situations that the added noise itself is not Gaussian. Note that in general for a nonlinear system, regardless of the Gaussianity or non-Gaussianity of the added noise, the conditional distribution is non-Gaussian. Our analysis of Part (I) obtains the situations where a linear Gaussian system can be good enough to approximate a near-optimal control policy for a nonlinear system with additive Gaussian noise.

In a situation where the additive noise is non-Gaussian, the analysis of Part (I) can be extended as well, utilizing the Wentzell-Freidlin large deviations theory [171]. In this case, the linear surrogate system will not be Gaussian either. However, still the best linear filter to use is Kalman filter. Our future research will explore the extensions of Part (I)'s analysis to these types of problems.

In this chapter, we consider a partially-observed system with additive non-Gaussian noise. Note that even for a linear system with additive non-Gaussian noise, nonlinear filters, such as particle filters, which are Monte-Carlo sampling-based approximations of the Bayesian filtering, can provide better results, particularly in situations where multi-modality of the distribution with distant modes can arise. However, as opposed to the Kalman filter, a closed-form evolution of the estimation covariance does not exits in general for non-Gaussian filters. One method is to try to construct a nominal non-Gaussian system as that of the belief space variant of the T-LQG. Yet even this is not possible in general. Therefore, heuristic approaches emerge in this class of problems.

In this chapter, we explore the problem of planning under non-Gaussian uncertainty from a heuristic point of view that can provide insights regarding the issues existent in the non-Gaussian problems. Particularly, we present an alternative belief space planning method that utilizes particle filters to predict the covariance of possible observations of the system, and plans for trajectories that optimizes the predicted covariance of observations. Note that as opposed to the rigorously proven T-LQG approach, this methods relies on practical heuristics for computationally faster path planning with more general forms of uncertainties. We reduce the problem to a convex program implemented using MPC strategy. Because of convexity of this problem, and the small size of the optimization problem, with features such as independence of the optimization problem's dimension from the number of particles, this method is computationally much efficient than similar state-of-the-art approaches.

In some situations, due to the fact that the T-LQG approach requires the computation of the Riccati recursions (as its bottle-neck), the heuristic method of this chapter can provide faster re-planning, as well. Nevertheless, the analysis of the previous chapter showed that, heuristic methods based on optimizing measures of the Observability Gramian (OG) are not reliable measures for planning under uncertainty. Moreover, although the cost function of this chapter is not directly a measure of the OG, one might find similarities. In retrospect, the difference are the usage of particles to predict the covariance of possible observations, incorporating the initial uncertainty, and usage of a weighting matrix to tune the cost function to more desirable situations and to convexify the problem. We test the accuracy of this method by comparing it to the state-of-the-art methods, and our results show the correctness of the plans. Regardless, the T-LQG-based analysis for similar problems (with Gaussian uncertainty) should be the benchmark for reliability the planned trajectories. That is, untested heuristics can lead to failure. Therefore, in this chapter we provide situations and models where the results are close to T-LQG results in comparable situations, but with better computations.

We propose a trajectory-optimization method in here which also considers optimizing nominal performance; however, unlike the T-LQG which considers the nominal performance of estimation and the control effort, this method only considers the nominal performance of the observation covariance and the control effort. In essence, we have shown in the previous chapters that in order to optimize the nominal estimation performance, as in T-LQG, the Riccati equations should be solved. In fact, even with non-Gaussian additive uncertainty, Riccati equations or covariance evolution provides the linear approximate of the original covariance evolution, and thus, it is the least that should be optimized. Therefore, the method of this chapter does not make such claims. Indeed, the analysis of Chapter 10 also experimentally confirmed that using surrogates are not reliable. Instead, this chapter optimizes the nominal observation covariance evolution.

For a convex environment, we propose an optimization-based open-loop optimal control problem coupled with receding horizon control strategy to plan for high quality trajectories along which the uncertainty of the state localization is reduced while the system reaches a goal state with minimum control effort. In a static environment with non-convex state constraints, the optimization is modified by defining barrier functions to obtain collision-free paths while maintaining the previous goals. By initializing the optimization with trajectories in different homotopy classes and comparing the resultant costs, we improve the quality of the solution in the presence of action and measurement uncertainties. In dynamic environments with time-varying constraints such as moving obstacles or changing banned areas, the approach is extended to find collision-free trajectories. In this chapter, the underlying spaces are continuous, and distributions are non-Gaussian. Without obstacles, the optimization is a globally convex problem, while in the presence of obstacles it becomes locally convex. We demonstrate the performance of the method on different scenarios.

The method of this chapter utilizes,a stochastic MPC for planning in the belief space. Samples of an initial non-Gaussian belief are mapped into observation samples by applying the observation model to them. Then a cost function is designed with the objective of obtaining a more compressed ensemble of the predicted observation trajectories. Therefore, the Riccati equation is avoided during the planning stage. Hence, the goal of planning is also not estimation, rather, it is high quality observations. Our experiments show the usefulness of this method in practice. Additionally, the MLO assumption is not used. The core problem in a convex environment is convexified for common nonlinear observation models. Moreover, non-convex constraints are incorporated using the OPF method of the earlier chapters. In a static environment, we apply the proposed optimization over trajectories in different homotopy classes to find a collision-free trajectory with the lowest cost in the homotopy classes. Moreover, the simulations show the OPF method's quality where the optimization can be initialized with some tolerance of infeasibility (i.e., passing through obstacles but not through the local minima of the OPFs).

Dynamic environments are also considered with time-varying OPFs. As a result, in neither the static nor dynamic situations does the optimization vector size change and the decision variables remain solely as the control variables. This approach, can be used as an on-line planner due to its relatively low computational burden. The flexibility of the MPC also allows incorporating dynamic environments, which makes the algorithm suitable for on-line planning. Moreover, the low computation allows to consider different homotopy classes, thereby moving from locally-optimal solutions towards a better approximation of a globally-optimal approach by applying the algorithm over multiple homotopy classes.

## 11.1 Particle-Filter-Based Belief Space Planning

In this chapter, since we are not using the conditional distribution of the system for estimation, and we are just utilizing a finite-vector representation of it by means of particles, we will use the term belief to refer to the approximations of the conditional distribution. Kalman filters maintain a mean and covariance evolution of the estimates of the system. Whereas, particle filters utilize a Monte-Carlo sampling representation of the conditional distribution and propagate the samples utilizing sampling-based approximations of the Bayesian update equation. The most well-known type of these filters is the bootstrap or Sampling Importance Resampling (SIR) filter [14], which will be described here.

*Particle representation of belief:* We use a non-Gaussian particle filter representation of belief state $b_t$ at time step $t$ by taking a number $N$ of state samples $\{\mathbf{x}_t^i\}_{i=1}^N$ with importance weights $\{w_t^i\}_{i=1}^N$ [14, 221]. Here, each particle $\mathbf{x}_t^i$ is an $n_x-$dimensional vector, whereas its corresponding weight, $w_t^i$, is a scalar number. Therefore $b_t(\mathbf{x}) \approx \sum_{i=1}^N w_t^i \delta(\mathbf{x} - \mathbf{x}_t^i)$, where $\delta(\cdot)$ denotes the Dirac delta mass.

*Bootstrap filter:* In order to obtain the belief updates, we use a standard particle filter known as the SIR filter [14]. It can be proven that as the number of particles increases to infinity, the distribution of the particles tends to the true filtering distribution [221, 222]. An overall description of the SIR filter is in Algorithm 3. In steps 2 to 6 of this algorithm, new state samples are obtained using the previous set of samples and the prediction pdf, such that every previous particle generates a new particle and its corresponding weight is assigned using the likelihood function.

In steps 7 to 9, weights are normalized to make a probability distribution. Steps 10 to 13, describe the resampling part of the algorithm in which replicas of higher probability samples take place of some of the lowest weight particles. Overall, steps 1 to 9 correspond to the prediction step of the filtering process, whereas steps 10 to 13, correspond to the update procedure.

**Algorithm 3:** Particle Filtering Algorithm SIR approach

```
   Input  : Set of particles at $t-1$, $\mathcal{X}_{t-1}$, Observation at $t$, $\mathbf{z}_t$, Transition
            function, $p_{\mathbf{X}_{t+1}|\mathbf{U}_t,\mathbf{X}_t}(\cdot|\cdot,\cdot)$, Likelihood function, $p_{\mathbf{Z}_t|\mathbf{X}_t}(\cdot|\cdot)$
   Output: Set of particles at $t$, $\mathcal{X}_t$
1  $\bar{\mathcal{X}}_t = \mathcal{X}_t \leftarrow \phi$;
2  for $i = 1 : N$ do
3      sample $x_t^i \sim p_{\mathbf{X}_{t+1}|\mathbf{U}_t,\mathbf{X}_t}(\cdot|\mathbf{u},\mathbf{x}_{t-1}^i)$;
4      $\tilde{w}_t^i \leftarrow p_{\mathbf{Z}_t|\mathbf{X}_t}(\mathbf{z}_t|\mathbf{x}_t^i)$;
5      $\bar{\mathcal{X}}_t \leftarrow \bar{\mathcal{X}}_t \cup \langle \mathbf{x}_t^i, \tilde{w}_t^i \rangle$;
6  end
7  for $i = 1 : N$ do
8      $w_t^i = w_t^i / \sum_{j=1}^{N} \tilde{w}_t^j$;
9  end
10 for $i = 1 : N$ do
11     draw $i$ with probability $\propto w_t^i$;
12     $\mathcal{X}_t \leftarrow \mathcal{X}_t \cup \mathbf{x}_t^i$;
13 end
14 return $\mathcal{X}_t$
```

*System equations and linearizations:* We provide the linearizations of the process and observation models around the nominal trajectory of the system similar to the previous chapters. The equations are:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{G}_t \mathbf{w}_t, \tag{11.1a}$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{M}_t \mathbf{v}_t, \tag{11.1b}$$

$$\mathbf{x}^p_{t+1} = \mathbf{f}(\mathbf{x}^p_t, \mathbf{u}^p_t), \tag{11.1c}$$

$$\mathbf{z}^p_t = \mathbf{h}(\mathbf{x}^p_t), \tag{11.1d}$$

$$\tilde{\mathbf{x}}_{t+1} = \mathbf{A}_t\tilde{\mathbf{x}}_t + \mathbf{B}_t\tilde{\mathbf{u}}_t + \mathbf{G}_t\mathbf{w}_t, \tag{11.1e}$$

$$\tilde{\mathbf{z}}_t = \mathbf{H}_t\tilde{\mathbf{x}}_t + \mathbf{M}_t\mathbf{v}_t, \tag{11.1f}$$

where $\mathbf{x}^p_0 := \mathbb{E}[\mathbf{x}_0]$, $\mathbf{A}_t := \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x},\mathbf{u})|_{\mathbf{x}^p_t,\mathbf{u}^p_t}$, $\mathbf{B}_t := \nabla_{\mathbf{u}}\mathbf{f}(\mathbf{x},\mathbf{u})|_{\mathbf{x}^p_t,\mathbf{u}^p_t}$, $\mathbf{G}_t$ is time-dependent constant matrix, $\mathbf{H}_t = \nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x})|_{\mathbf{x}^p_t}$, and $\tilde{\mathbf{x}}_t := \mathbf{x}_t - \mathbf{x}^p_t$, $\tilde{\mathbf{u}}_t := \mathbf{u}_t - \mathbf{u}^p_t$, and $\tilde{\mathbf{z}}_t := \mathbf{z}_t - \mathbf{z}^p_t$ denote the state, control and observation errors, respectively. For holonomic systems and under saturation constraints, a linear model suffices for planning purposes. This is because, these systems can track a given trajectory without insignificant morphing.

*Cost function:* Using the incremental cost $c(\cdot,\cdot) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$, we define the cost function as:

$$\mathbb{E}[\sum_{t=1}^{K} c(\mathbf{x}_t, \mathbf{u}_t)] = \sum_{t=1}^{K} \mathbb{E}[\tilde{\mathbf{z}}_t^T \mathbf{R}_t \tilde{\mathbf{z}}_t + \mathbf{u}_{t-1}^T \mathbf{V}^u_t \mathbf{u}_{t-1}], \tag{11.2}$$

where $\mathbf{V}^u_t \succ 0$ is positive definite matrices, and $\mathbf{R}_t(\mathbf{x}^p_t) : \mathbb{R} \times \mathbb{X} \to \mathbb{R}^{n_z \times n_z}$ is a proper weighting matrix, to be defined later. This cost seeks to reduce the dispersion in the ensemble of the observation trajectories in terms of the weighted covariance. In other words, the minimization seeks to reduce the uncertainty in the predicted observation, which translates itself to shrinking the support of belief distribution. In addition, it considers reducing the control effort, as well.

*Connections of the cost to the observation covariance:* Note that we have:

$$\mathbb{E}[\tilde{\mathbf{z}}_t^T \mathbf{R}_t \tilde{\mathbf{z}}_t] = \mathrm{tr}(\mathbb{E}[\mathbf{R}_t^{1/2} \tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^T (\mathbf{R}_t^{1/2})^T]) = \mathrm{tr}(\mathbf{R}_t^{1/2} \mathbb{E}[\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^T] (\mathbf{R}_t^{1/2})^T) \tag{11.3a}$$

$$= \mathbb{E}[(\mathbf{H}_t\tilde{\mathbf{x}}_t)^T \mathbf{R}_t \mathbf{H}_t \tilde{\mathbf{x}}_t] + \mathbb{E}[(\mathbf{M}_t\mathbf{v}_t)^T \mathbf{R}_t \mathbf{M}_t \mathbf{v}_t] \tag{11.3b}$$

$$= \mathbb{E}[\tilde{\mathbf{x}}_t^T \mathbf{H}_t^T \mathbf{R}_t \mathbf{H}_t \tilde{\mathbf{x}}_t] + \mathbb{E}[\mathbf{v}_t^T \mathbf{M}_t^T \mathbf{R}_t \mathbf{M}_t \mathbf{v}_t] \tag{11.3c}$$

$$= \mathbb{E}[\tilde{\mathbf{x}}_t^T \mathbf{W}_t \tilde{\mathbf{x}}_t] + \mathbb{E}[\mathbf{v}_t^T \mathbf{M}_t^T \mathbf{R}_t \mathbf{M}_t \mathbf{v}_t] \tag{11.3d}$$

$$= \mathbb{E}[\tilde{\mathbf{x}}_t^T \mathbf{W}_t \tilde{\mathbf{x}}_t] + \mathrm{tr}(\mathbf{R}_t^{1/2} \mathbf{M}_t \mathbb{E}[\mathbf{v}_t \mathbf{v}_t^T] \mathbf{M}_t^T (\mathbf{R}_t^{1/2})^T) \tag{11.3e}$$

$$= \mathrm{tr}(\mathbf{W}_t^{1/2} \mathbb{E}[\tilde{\mathbf{x}}_t \tilde{\mathbf{x}}_t^T] (\mathbf{W}_t^{1/2})^T) + \mathrm{tr}(\mathbf{R}_t^{1/2} \mathbf{M}_t \mathbf{\Sigma}_{\mathbf{v}_t} \mathbf{M}_t^T (\mathbf{R}_t^{1/2})^T), \tag{11.3f}$$

where $\mathbf{\Sigma}_{\mathbf{v}_t} := \mathbb{E}[\mathbf{v}_t \mathbf{v}_t^T]$ is the covariance of the observation noise. Note that, if $\mathbf{R} = \mathbf{I}_{n_z}$ where $\mathbf{I}_{n_z}$ is the $n_z$-dimensional identity matrix, this term becomes $\mathrm{tr}(\mathrm{Cov}[(\mathbf{z}_t - \mathbf{z}_t^p)])$. Otherwise it is a weighted observation variance. where $(\mathbf{z}_t - \mathbf{z}_t^p)$ is the predicted error of the observation from its nominal observation at time-step $t$. Therefore, conceptually, the minimum of this cost occurs over the state trajectories along where the covariance dispersion in the ensemble of the observation trajectories is reduced. This means that the minimization seeks to reduce the uncertainty in the observations, which can potentially lead to better trajectories.

## 11.2 Approximation of the Cost

In this subsection, we use the particle filter representation of the belief to obtain more tractable approximations of the cost function.

*Utilizing the T-LQG concepts:* First, note that using the belief space variant of the T-LQG method, assuming the application of a linear feedback law, and using the interpretation given in (11.3f) for the equivalence of the first term of the cost function with a quadratic cost in the state error, we expect that the cost function can be approximated by its nominal counter part. That us we assume heuristically that even for this case (the non-Gaussian perturbations), the first order expected error of the cost function is zero. Note that this we have not proven this, and this is heuristic. Also note that in Chapter 6, we showed that the results hold even if $\mathbf{L}_t = \mathbf{0}$. However, due to the change in the transfer function, the linearization's validity probability changes dramatically, requiring a much smaller noise intensity. For now, we will assume that the feedback gain that we are gonna use is $\mathbf{L}_t = \mathbf{0}$. We also heuristically use this result for the current analysis. Therefore,

$$\tilde{\mathbf{x}}_t \approx \tilde{\mathbf{A}}_{0:t-1}\tilde{\mathbf{x}}_0 + \sum_{r=0}^{t-1} \tilde{\mathbf{A}}_{r+1:t-1}\mathbf{G}_r\mathbf{w}_r, \tag{11.4}$$

where where $\tilde{\mathbf{A}}_{t_1:t_2} := \prod_{\tau=t_1}^{t_2} \mathbf{A}_\tau = \mathbf{A}_{t_2}\mathbf{A}_{t_2-1}\cdots\mathbf{A}_{t_1}, t_1 \leq t_2$, otherwise it is identity matrix. Now, we approximate $\mathbb{E}[\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^T]$ as follows:

$$\mathbb{E}[\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^T] \approx \tilde{\mathbf{A}}_{0:t-1}\mathbb{E}[\tilde{\mathbf{x}}_0\tilde{\mathbf{x}}_0^T]\tilde{\mathbf{A}}_{0:t-1}^T + \sum_{r=0}^{t-1} \tilde{\mathbf{A}}_{r+1:t-1}\mathbf{G}_r\mathbb{E}[\mathbf{w}_r\mathbf{w}_r^T]\mathbf{G}_r^T\tilde{\mathbf{A}}_{r+1:t-1}^T \tag{11.5a}$$

$$= \tilde{\mathbf{A}}_{0:t-1}\boldsymbol{\Sigma}_{\mathbf{x}_0}\tilde{\mathbf{A}}_{0:t-1}^T + \sum_{r=0}^{t-1} \tilde{\mathbf{A}}_{r+1:t-1}\mathbf{G}_r\boldsymbol{\Sigma}_{\mathbf{w}_r}\mathbf{G}_r^T\tilde{\mathbf{A}}_{r+1:t-1}^T, \tag{11.5b}$$

where $\boldsymbol{\Sigma}_{\mathbf{x}_0} := \mathbb{E}[\tilde{\mathbf{x}}_0\tilde{\mathbf{x}}_0^T]$ is the initial covariance, and $\boldsymbol{\Sigma}_{\mathbf{w}_t} := \mathbb{E}[\mathbf{w}_t\mathbf{w}_t^T]$ is the process noise covariance. Moreover, since we have used the feedback gain as zero, we replace the control effort with its nominal counterpart. Therefore, the nominal cost is defined as follows:

$$\begin{aligned} J^p := \sum_{t=1}^{K} \Big[ &\mathrm{tr}(\mathbf{W}_t^{1/2}\tilde{\mathbf{A}}_{0:t-1}\boldsymbol{\Sigma}_{\mathbf{x}_0}\tilde{\mathbf{A}}_{0:t-1}^T(\mathbf{W}_t^{1/2})^T) \\ &+ \sum_{r=0}^{t-1} \mathrm{tr}(\mathbf{W}_t^{1/2}\tilde{\mathbf{A}}_{r+1:t-1}\mathbf{G}_r\boldsymbol{\Sigma}_{\mathbf{w}_r}\mathbf{G}_r^T\tilde{\mathbf{A}}_{r+1:t-1}^T(\mathbf{W}_t^{1/2})^T) \\ &+ \mathrm{tr}(\mathbf{R}_t^{1/2}\mathbf{M}_t\boldsymbol{\Sigma}_{\mathbf{v}_t}\mathbf{M}_t^T(\mathbf{R}_t^{1/2})^T) + (\mathbf{u}_{t-1}^p)^T\mathbf{V}_t^u\mathbf{u}_{t-1}^p \Big], \end{aligned} \tag{11.6}$$

*Initial covariance:* Given a set of particles $\{\mathbf{x}_0^i\}_{i=1}^k$ at time-step 0, we approximate the initial covariance as follows:

$$\boldsymbol{\Sigma}_{\mathbf{x}_0} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_0^i - \mathbf{x}_0^p)(\mathbf{x}_0^i - \mathbf{x}_0^p)^T, \tag{11.7}$$

As a result, the first term in the nominal cost can be written as follows:

$$\frac{1}{N}\sum_{i=1}^{N}[(\mathbf{x}_0^i - \mathbf{x}_0^p)^T \tilde{\mathbf{A}}_{0:t-1}^T \mathbf{W}_t \tilde{\mathbf{A}}_{0:t-1}(\mathbf{x}_0^i - \mathbf{x}_0^p)]. \tag{11.8}$$

Next, we define the optimization problem, resulting from this approximation. Note that since we have ignored the feedback's correction but assumed its existence, we use an MPC strategy to obtain feedback.

**Problem 19 Deterministic Open-Loop Problem***: Given an initial state* $\bar{\mathbf{x}}_0$*, we begin by determining an optimal open-loop sequence for*

$$\begin{aligned}\min_{\mathbf{u}_{0:K-1}} \sum_{t=1}^{K} \Big[ & \mathrm{tr}(\mathbf{W}_t^{1/2} \tilde{\mathbf{A}}_{0:t-1} \mathbf{\Sigma}_{\mathbf{x}_0} \tilde{\mathbf{A}}_{0:t-1}^T (\mathbf{W}_t^{1/2})^T) \\ & + \sum_{r=0}^{t-1} \mathrm{tr}(\mathbf{W}_t^{1/2} \tilde{\mathbf{A}}_{r+1:t-1} \mathbf{G}_r \mathbf{\Sigma}_{\mathbf{w}_r} \mathbf{G}_r^T \tilde{\mathbf{A}}_{r+1:t-1}^T (\mathbf{W}_t^{1/2})^T) \\ & + \mathrm{tr}(\mathbf{R}_t^{1/2} \mathbf{M}_t \mathbf{\Sigma}_{\mathbf{v}_t} \mathbf{M}_t^T (\mathbf{R}_t^{1/2})^T) + (\mathbf{u}_{t-1}^p)^T \mathbf{V}_t^u \mathbf{u}_{t-1}^p \Big] \qquad (11.9) \\ s.t.\ \mathbf{x}_{t+1}^p = \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p). & \qquad (11.10)\end{aligned}$$

## 11.3 Special Case: Linear Process Model

Let us assume that the process model is linear. Moreover, for simplification of the cost function, let us ignore the effects of $\mathbf{\Sigma}_{\mathbf{w}_t}$ and $\mathbf{\Sigma}_{\mathbf{v}_t}$. That is, we ignore the second order effects of the perturbations. This way, we loose accuracy, instead we gain computational efficiency. Next, we provide the simplified form of the cost function.

*Simplified cost function:* Let us define a vector $\mathbf{e}_t := (\mathbf{e}_t^{1^T}, \mathbf{e}_t^{2^T}, \cdots, \mathbf{e}_t^{N^T})^T \in \mathbb{R}^{Nn_x}$, where $\mathbf{e}_t^i := \frac{1}{\sqrt{N}}\tilde{\mathbf{A}}_{0:t-1}(\mathbf{x}_0^i - \mathbf{x}_0^p) \in \mathbb{R}^{n_x}$ for $1 \leq i \leq N$, and a matrix $\bar{\mathbf{W}}(\mathbf{x}_t^p) :=$ BlockDiag$(\mathbf{W}(\mathbf{x}_t^p))$ with $N$ equal diagonal blocks of $\mathbf{W}(\mathbf{x}_t^p)$. The cost function simplifies to:

$$\sum_{t=1}^{K}[\sum_{i=1}^{N}[\mathbf{e}_t^{iT}\mathbf{W}(\mathbf{x}_t^p)\mathbf{e}_t^i] + \mathbf{u}_{t-1}^T\mathbf{V}_t^u\mathbf{u}_{t-1}] = \sum_{t=1}^{K}[\mathbf{e}_t^T\bar{\mathbf{W}}(\mathbf{x}_t^p)\mathbf{e}_t + \mathbf{u}_{t-1}^T\mathbf{V}_t^u\mathbf{u}_{t-1}],$$

where $\mathbf{e}_t$ is a constant vector at each time-step $t$. Moreover, let $\mathbf{f}_t^p := \mathbf{f}(\mathbf{x}_t^p, \mathbf{u}_t^p) - \mathbf{A}_t^p\mathbf{x}_t^p - \mathbf{B}_t^p\mathbf{u}_t^p$, then $\mathbf{x}_t^p = \tilde{\mathbf{A}}_{0:t-1}\mathbf{x}_0^p + \sum_{s=0}^{t-1}\tilde{\mathbf{A}}_{s+1:t-1}(\mathbf{B}_s\mathbf{u}_s + \mathbf{f}_s^p)$ which is the noiseless prediction of the initial mean of the estimate.

Next, we discuss how to convexify this cost function.

### *11.3.1 Convexifying the Cost Function*

In this subsection, we convexify the cost function through the proper design of the $\mathbf{R}$ matrix. First, we consider a situation with one scalar observation, then we extend it for a general case.

**Lemma 9 (Scalar observation)** *Suppose* $\mathbf{d} = (d_1, \cdots, d_{n_x})^T \in \mathbb{R}^{n_x}$ *and* $h(\mathbf{x}) : \mathbb{X} \to \mathbb{R}$ *is differentiable. If* $l : \mathbb{X} \to \mathbb{R}$ *defined as* $l(\mathbf{x}) := \sqrt{R(\mathbf{x})}\mathbf{H}(\mathbf{x})\mathbf{d}$*, is convex or concave in* $\mathbf{x}$*, then* $g : \mathbb{X} \to \mathbb{R}_{\geq 0}$*, where* $g(\mathbf{x}) := \mathbf{d}^T\mathbf{H}(\mathbf{x})^T R(\mathbf{x})\mathbf{H}(\mathbf{x})\mathbf{d}$ *is a convex function of* $\mathbf{x}$*, where* $\mathbf{H}(\mathbf{x}) := \nabla h(\mathbf{x})|_{\mathbf{x}}$ *is the Jacobian of* $h$*.*

**Proof 27** *The Jacobian of* $h$ *is* $\mathbf{H}(\mathbf{x}) = \begin{bmatrix} H_1(\mathbf{x}), \cdots, H_{n_x}(\mathbf{x}) \end{bmatrix}$*, where* $H_i(\mathbf{x}) := \dfrac{\partial \mathbf{h}(\mathbf{x})}{\partial x_i}$*, for* $1 \leq i \leq n_x$*. Thus,* $\mathbf{H}(\mathbf{x})^T\mathbf{H}(\mathbf{x}) = \begin{bmatrix} H^TH(\mathbf{x})_{ij} \end{bmatrix}$*, which is a symmetric matrix and* $H^TH(\mathbf{x})_{ij} := H_i(\mathbf{x})H_j(\mathbf{x})$ *, for* $1 \leq i, j \leq n_x$*. Next, we can express* $\mathbf{B}(\mathbf{x}) := \mathbf{d}^T\mathbf{H}(\mathbf{x})^T\mathbf{H}(\mathbf{x})$ *as* $\mathbf{B}(\mathbf{x}) = \begin{bmatrix} B_1(\mathbf{x}), \cdots, B_{n_x}(\mathbf{x}) \end{bmatrix}$*, where* $B_j(\mathbf{x}) = \begin{bmatrix} \sum_{i=1}^{n_x} c_i H^TH(\mathbf{x})_{ij} \end{bmatrix}$*, for* $1 \leq j \leq n_x$*. Therefore,* $\mathbf{d}^T\mathbf{H}(\mathbf{x})^T\mathbf{H}(\mathbf{x})\mathbf{d}$ *can be written as:*

$$\begin{aligned}\mathbf{d}^T\mathbf{H}(\mathbf{x})^T\mathbf{H}(\mathbf{x})\mathbf{d} &= \sum_{j=1}^{n_x} B_j(\mathbf{x})d_j = \sum_{j=1}^{n_x}\sum_{i=1}^{n_x} d_i H^TH(\mathbf{x})_{ij}d_j = \sum_{j=1}^{n_x}\sum_{i=1}^{n_x} d_i H_i(\mathbf{x})H_j(\mathbf{x})d_j \\ &= (\sum_{i=1}^{n_x} d_i H_i(\mathbf{x}))^2 = (\mathbf{d} \cdot \mathbf{H}(\mathbf{x})^T)^2 = (\mathbf{H}(\mathbf{x})\mathbf{d})^2\end{aligned}$$

*Thus, $g(\mathbf{x})$ is nothing but $g(\mathbf{x}) = (l(\mathbf{x}))^2$. Therefore, if $l(\mathbf{x})$ is a convex or concave function of $\mathbf{x}$, $g(\mathbf{x})$ will be a convex function of $\mathbf{x}$, and $g(\mathbf{x}) \geq 0$.*

*Multiple observations*: We extend the results derived for the scalar observation to the case where there are multiple vector observations. Particularly, we show that the convexity and all the desired features remain unchanged, as long as the design feature of the Lemma 9 are followed. For an observation vector $\mathbf{z} = [z_1, \cdots, z_{n_z}]^T$ in $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{M}_t\mathbf{v}$, and given the differentiable function $\mathbf{h}(\mathbf{x}) = [h_j(\mathbf{x})]$, and its Jacobian $\mathbf{H}(\mathbf{x}) = [\mathbf{H}_j(\mathbf{x})]$, where $\mathbf{H}_j(\mathbf{x}) = \left[\dfrac{\partial h_j(\mathbf{x})}{\partial x_1}, \cdots, \dfrac{\partial h_j(\mathbf{x})}{\partial x_{n_x}}\right]$ for $1 \leq j \leq n_z$, if $\mathbf{R}(\mathbf{x}) = \mathrm{diag}(R_j(\mathbf{x}))$ is the diagonal matrix of $R_i(\mathbf{x})$'s corresponding to (uncorrelated) observations, extending the definition of $g$ to include matrix $\mathbf{R}$ we have:

$$\begin{aligned}\mathbf{d}^T\mathbf{H}(\mathbf{x})^T\mathbf{R}(\mathbf{x})\mathbf{H}(\mathbf{x})\mathbf{d} &= \sum_{j=1}^{n_z} \mathbf{d}^T\mathbf{H}_j(\mathbf{x})^T R_j(\mathbf{x})\mathbf{H}_j(\mathbf{x})\mathbf{d} \\ &= \sum_{j=1}^{n_z} R_j(\mathbf{x}) \sum_{k=1}^{n_x} (d_k\mathbf{H}_{jk}(\mathbf{x}))^2 \\ &= \sum_{j=1}^{n_z} R_j(\mathbf{x})(\mathbf{H}_j(\mathbf{x})\mathbf{d})^2,\end{aligned}$$

which is a sum of positive convex functions as determined in Lemma 9. Therefore, in the case of multiple observations, the same results still hold.

*Another representation of the cost function:* In our cost function, vector $\mathbf{d}$ represents any of the vectors $\mathbf{e}_t^i$ for $1 \leq i \leq N$ and any $t$. Therefore, we can re-write the cost function as:

$$\sum_{t=1}^{K}[(\sum_{i=1}^{N}\sum_{j=1}^{n_z} R_j(\mathbf{x}_t^p)(\mathbf{H}_j(\mathbf{x}_t^p)\mathbf{e}_t^i)^2) + \mathbf{u}_{t-1}^T\mathbf{V}_t^u\mathbf{u}_{t-1}]. \tag{11.11}$$

*Designing the desired convex cost:* Design of $\mathbf{R}(\mathbf{x})$ is heuristic, but we in fact tailor it such that the cost function becomes convex. A common heuristic is to consider the design of $\mathbf{R}(\mathbf{x})$ to be a function of some distance measure from known states that are informative. This is specially desirable when the observation covariance reduces at a closer range to the information source. This is the case in e.g., a light-dark, GPS, or beacon models. Although, this is not generalizable. In fact, we devoted the previous chapter to this issue. Another class of problems in which this can be beneficial is where it is desirable to design trajectories along which the state gets closer to some known regions (or even sink states), in addition to achieving other goals.

Therefore, the physical conditions of the problem can be utilized to design the $\mathbf{R}(\mathbf{x})$ matrix that makes the problem more tractable and give desired features to the problem. For our purposes, we design $\mathbf{R}$ to have the desired features of Lemma 9. Examples of such designs are provided in Section 11.3.2.

### *11.3.2 Examples of Observation Models*

In this subsection, we provide some of the most common observation models in the literature and show that we can obtain the goals described in the previous section. Particularly, we will design the $\mathbf{R}$ matrices that convexify each cost function based on those observation models.

**Example 1 (The range-based measurements)** *In a range based measurement from known landmarks,* $\mathbf{h}(\mathbf{x}) = \|\mathbf{x} - \mathbf{L}\|_2$*, where,* $\|\cdot\|_2$ *denotes the Euclidean norm, and* $\mathbf{L} \in \mathbb{R}$ *is a known state (landmark). Therefore, the Jacobian's ith component is* $\mathbf{H}_i(\mathbf{x}) = [(x_i - L_i)/\|\mathbf{x} - \mathbf{L}\|_2]$ *for* $1 \leq i \leq n_x$*. Moreover,* $R(\mathbf{x}) = \|\mathbf{x} - \mathbf{L}\|_2^2$ *has the desired properties discussed above. Thus, we have* $g(\mathbf{x}) = (\sum_{i=1}^{n_x} d_i(x_i - L_i))^2 = (\mathbf{d} \cdot (\mathbf{x} - \mathbf{L}))^2 = ((\mathbf{x} - \mathbf{L})^T \mathbf{d})^2$*, which is convex in* $\mathbf{x}$*.*

**Example 2 (The bearing-based measurements)** *Given a state vector* $\mathbf{x} = [x, y, \theta]^T$*, and* $\mathbf{L} = [L_x, L_y]^T$*, in a bearing measurement from this landmark, we have* $\mathbf{h}(\mathbf{x}) = atan2(y - L_y, x - L_x) - \theta$*. Hence, the Jacobian is formed as* $\mathbf{H}(\mathbf{x}) = [\frac{-(y-L_y)}{r^2}, \frac{(x-L_x)}{r^2}, -1]$*,* *where* $r = \sqrt{(x-L_x)^2+(y-L_y)^2}$ *is the range from the landmark. Thus, using* $R(\mathbf{x}) = ((x-L_x)^2+(y-L_y)^2)^2$ *we have* $g(\mathbf{x}) = (d_1(x-L_x)+d_2(y-L_y)-d_3((x-L_x)^2+(y-L_y)^2))^2$*, which is a convex function in* $\mathbf{x}$*.*

**Example 3 (Measurements with exponential decay of covariance)** *Let the observation model be linear* $\mathbf{h}(\mathbf{x}) = \mathbf{D}(\mathbf{x}-\mathbf{L})$*, where* $\mathbf{D} = [\mathbf{D}_1^T, \cdots, \mathbf{D}_{n_z}^T]^T$ *is a constant* $n_z \times n_x$ *matrix. With* $\mathbf{R}(\mathbf{x}) = \exp((\eta_L\|\mathbf{x}-\mathbf{L}\|_2+\sigma_b))\mathbf{I}_{n_z}$*, where* $\eta_L$*, and* $\sigma_b$ *are positive constants and* $\mathbf{I}_{n_z}$ *is the* $n_z$*-dimensional identity matrix,* $g(\mathbf{x}) = \exp(\eta_L\|\mathbf{x}-\mathbf{L}\|_2+\sigma_b)\sum\limits_{j=1}^{n_z}(\mathbf{d}\cdot\mathbf{D}_j)^2$ *is a convex function in* $\mathbf{x}$*.*

**Example 4 (Light-dark environment in literature [109])** *In a light-dark environment with an observation model of* $\mathbf{h}(\mathbf{x}) = \mathbf{D}(\mathbf{x}-\mathbf{L})$ *and* $\mathbf{R}(\mathbf{x}) = \sqrt{(\eta x_i+\sigma_b)}\mathbf{I}_{n_z}$ *for some* $1 \leq i \leq n_x$*, where* $\eta$ *and* $\sigma_b$ *are positive constants and* $\mathbf{D}$ *is defined as before, we have,* $g(\mathbf{x}) = (\sum\limits_{j=1}^{n_z}(\mathbf{d}\cdot\mathbf{D}_j)^2)(\eta x_i+\sigma_b) > 0$ *which is a convex function in* $\mathbf{x}$*. Another instance is where* $\mathbf{R}(\mathbf{x}) = (\eta x_i+\sigma_b)\mathbf{I}_{n_z}$*,* $(\eta x_i+\sigma_b) > 0$*, then* $g(\mathbf{x}) = (\sum\limits_{j=1}^{n_z}(\mathbf{d}\cdot\mathbf{D}_j)^2)(\eta x_i+\sigma_b)^2$ *which is convex in* $\mathbf{x}$ *(in the defined domain), as well.*

**Example 5 (Single Beam model in literature [109])** *In a similar observation model where* $\mathbf{h}(\mathbf{x}) = \mathbf{D}(\mathbf{x}-\mathbf{L})$*, and* $\mathbf{R}(\mathbf{x}) = \sqrt{\eta'_L/(d_M-\eta_L\|\mathbf{x}-\mathbf{L}\|_2+\sigma_b)}\mathbf{I}_{n_z}$*, where* $\eta_L, \eta'_L, d_M$*, and* $\sigma_b$ *are positive constants and* $\mathbf{D}$ *is defined as before, we have* $g(\mathbf{x}) = (\sum\limits_{j=1}^{n_z}(\mathbf{d}\cdot\mathbf{D}_j)^2)\eta'_L/(d_M-\eta_L\|\mathbf{x}-\mathbf{L}\|_2+\sigma_b) > 0$ *which is a convex function in* $\mathbf{x}$*. Another instance is where* $\mathbf{R}(\mathbf{x}) = \eta'_L/(d_M-\eta_L\|\mathbf{x}-\mathbf{L}\|_2+\sigma_b)\mathbf{I}_{n_z}$*,* $(d_M-\eta_L\|\mathbf{x}-\mathbf{L}\|_2+\sigma_b) > 0$*, where* $\eta$*, then* $g(\mathbf{x}) = (\sum\limits_{j=1}^{n_z}(\mathbf{d}\cdot\mathbf{D}_j)^2)(\eta'_L)^2/(d_M-\eta_L\|\mathbf{x}-\mathbf{L}\|_2+\sigma_b)^2$ *which is convex in* $\mathbf{x}$ *(in the defined domain), as well.*

The following examples are more trivial yet common in the field.

**Example 6 (Signed distance with range-proportional covariance)** *Let the observation model be linear* $\mathbf{h}(\mathbf{x}) = \mathbf{D}(\mathbf{x}-\mathbf{L})$ *with* $\mathbf{R}(\mathbf{x}) = (\eta_L\|\mathbf{x}-\mathbf{L}\|_2+\sigma_b)\mathbf{I}_{n_z}$*, where*

*$\eta_L$, and $\sigma_b$ are positive constants, and $\mathbf{D}$ is defined as before. Note that this is usually coupled with $d_M - \|\mathbf{x} - \mathbf{L}\|_2 > 0$ which provides a maximum field of view for the sensor through $d_M > 0$, However, mathematically this might be omitted as well in this case. Then, $g(\mathbf{x}) = (\sum_{j=1}^{n_z} (\mathbf{d} \cdot \mathbf{D}_j)^2)\eta_L^2(\|\mathbf{x} - \mathbf{L}\|_2 + \sigma_b)^2$ is a convex function in $\mathbf{x}$.*

**Example 7 (Absolute bearing)** *Similar to the previous example on bearing measurement, given a state vector $\mathbf{x} = [x, y, \theta]^T$, and $\mathbf{L} = [L_x, L_y]^T$, in a bearing measurement from this landmark, if we make observations like $\mathbf{h}(\mathbf{x}) = \arctan((y - L_y)/(x - L_x))$, then, $\mathbf{H}(\mathbf{x}) = [\frac{-(y-L_y)}{r^2}, \frac{(x-L_x)}{r^2}]$, where $r = \sqrt{(x - L_x)^2 + (y - L_y)^2}$ is the range from the landmark. Thus, using $R(\mathbf{x}) = ((x - L_x)^2 + (y - L_y)^2)^2$ we have $g(\mathbf{x}) = (d_1(x - L_x) + d_2(y - L_y))^2$, which is a convex function in $\mathbf{x}$.*

**Example 8 (GPS-like observations)** *Finally, we mention the more trivial example of GPS-like observations where the state is directly observed with some background noise. We model this noise to have a constant covariance. Thus, let the observation model be linear $\mathbf{h}(\mathbf{x}) = \mathbf{D}\mathbf{x}$, where $\mathbf{D}$ is defined as before. With $\mathbf{R}(\mathbf{x}) = \sigma_b \mathbf{I}_{n_z}$, $\sigma_b > 0$. Then, $g(\mathbf{x}) = \sigma_b^2 \sum_{j=1}^{n_z} (\mathbf{d} \cdot \mathbf{D}_j)^2$ is a convex function in $\mathbf{x}$. Note that, since in this case the observation model is completely indifferent to the specific state that the observation is made from, it will not matter for the controller to move to any specific state to make 'better' or 'more accurate' observations from. This is correctly reflected in the $g(\mathbf{x})$ which is trivially a constant independent of the state the observation is being obtained in. However, if a GPS is used in a covered area with poor connectivity, then the objective can be designed in a similar fashion to previous examples such that the agent seeks proximity to states with better coverage, such as near the windows.*

Last, as a design objective, note that unless we make the two terms of the cost function within the same order of magnitude, one term will be dominant. It is usually desirable to make one term slightly dominant, say by one order of magnitude.

However, in order to have a numerically sound optimization, particularly with time- (or state-)dependent weight matrices, the tuning of the weights is very important especially for practical implementation purposes.

### *11.3.3 Convex Optimization Problem*

Finally, we define the open-loop optimization problem of this section for a convex environment. Define the cost of information as $\mathrm{cost}_{info}(\mathbf{x}_t^p) := \mathbf{e}_t^T \bar{\mathbf{W}}(\mathbf{x}_t^p)\mathbf{e}_t$ and cost of control effort as $\mathrm{cost}_{eff}(\mathbf{u}_t) := \mathbf{u}_t^T \mathbf{V}_{t+1}^u \mathbf{u}_t$. Hence, the core convex problem is given below.

**Problem 20 (Core convex problem in convex feasible space)** *Under the assumptions of linear (holonomic) system and convex environment and given the initial re-sampled set of particles at time step* $0$*,* $\{\mathbf{x}_0^i\}_{i=1}^k$*, and a goal state* $\mathbf{x}_g$*, the core convex problem is:*

$$\begin{aligned} &\min_{\mathbf{u}_{0:K-1}} \sum_{t=1}^{K} [\mathrm{cost}_{info}(\mathbf{x}_t^p) + \mathrm{cost}_{eff}(\mathbf{u}_{t-1})] \\ &s.t. \;\; \mathbf{x}_K^p = \mathbf{x}_g, \end{aligned}$$

*where* $\mathbf{x}_t^p = \tilde{\mathbf{A}}_{0:t-1}\mathbf{x}_0^p + \sum_{s=0}^{t-1} \tilde{\mathbf{A}}_{s+1:t-1}(\mathbf{B}_s\mathbf{u}_s + \mathbf{f}_s^p)$.

Note that for some of the observation models that are considered in the previous section, the above problem reduces to a quadratic program, which even has a closed-form solution.

### *11.3.4 Static Environment with Non-Convex Constraints*

In this subsection, we extend the solution of the previous subsection to include non-convex constraints on the state, such as obstacles and banned areas in static environment with a known map of the environment. For this purpose, we use the obstacle barrier function method of previous chapters. The optimization in such a case reduces to a locally convex optimization. Similar to any nonlinear program, we need to initialize the optimization with a trajectory. Note that if we use the OPF method, starting from a feasible trajectory is more desirable. Moreover, in that situation, the optimization avoids entering non-feasible states. However, if we use the OPF method, this trajectory does not need to be feasible, and cans sightly violate some constraints. In this situation, the penalty function needs to be tuned. In fact, the OPF method has lower computation and in practice is more desirable. Furthermore, by initialing the optimization with trajectories in different homotopy classes, we find the locally optimal trajectories in different homotopy classes. We discuss the benefit of doing this towards the end of this subsection.

**Problem 21 (Locally convex problem in a static environment)** *Given* $\{\mathbf{x}_0^i\}_{i=1}^k$, $\mathbf{x}_g$ *and obstacle parameters* $(\mathcal{P}, \mathcal{C})$*, the static environment problem for a holonomic system is:*

$$\begin{aligned} &\min_{\mathbf{u}_{0:0+K-1}} \sum_{t=0+1}^{0+K} [\text{cost}_{info}(\mathbf{x}_t^p) + \text{cost}_{eff}(\mathbf{u}_{t-1}) + \text{cost}_{obst}(\mathbf{x}_{t-1}^p, \mathbf{x}_t^p)] \\ &s.t. \;\; \mathbf{x}_K^p = \mathbf{x}_g. \end{aligned} \tag{11.12}$$

*where the cost of obstacles is defined in the previous chapters.*

Moreover, we add convex saturation constraints of the type $\|\mathbf{u}_t\| \leq \max_u$ based on the specific robot model.

Next, we proceed to optimize towards a better approximation among different homotopy classes while reaching predefined goals, such as uncertainty reduction, collision avoidance, and reaching the final destination with minimal energy effort.

*Homotopy classes and optimal trajectory:* There are several methods to find the trajectories in homotopy classes [187, 188]. For instance, in low dimensions one can construct the visibility graph considering the pure motion planning problem and find trajectories in different homotopy classes that connect the start state to the goal state by pruning the non-unique paths. These methods provide such paths for different purposes such as finding the shortest path. However, usually the uncertainty or dynamics of the system are not considered. We initialize our optimization with non-looped trajectories in different homotopy classes [188]. The optimizer considers the cost of uncertainty, effort, and collision-avoidance along with the linearized dynamics of the (holonomic) system and morphs the initial trajectory towards a locally optimal trajectory. Our barrier function model of the obstacles prevents the trajectory from entering the banned regions. These barrier functions, along with a optimization tuned through the saturation constraints, a long enough optimization horizon (determined by the time-discretization step of the initial trajectory), and a limited step size of the line-search in optimization [223, 35], keep the trajectory in its initial homotopy class. Moreover, since the optimization is locally convex, it finds the local optimal trajectory of that homotopy class under the imposed constraints and conditions starting from a trajectory in that class. Therefore, by comparing the total costs obtained in different cases, we obtain the lowest cost smooth trajectory considering all the predefined costs, and most significant of all, uncertainty reduction. This is the closest output trajectory of our algorithm to the globally optimal trajectory in the existence of uncertainties.

### *11.3.5 Problem: Dynamic Environment with Time-Varying Constraints*

Now that we have specified all the machinery needed to find the optimal path in terms of the defined cost in a static environment, we extend our method to an environment that is not fully static.

*Incorporating dynamic obstacles:* If some of the obstacles are moving, the state constraints become time-varying. In such a case, we modify the optimization problem by altering the obstacle cost so it includes the dynamic obstacles as follows:

$$\Phi_t^{(\hat{\mathcal{P}}_t,\hat{\mathcal{C}}_t)}(\mathbf{x}):=M\sum_{i=1}^{n_b}[\exp(-[(\mathbf{x}-\hat{\mathbf{c}}_t^i)^T\hat{\mathbf{P}}_t^i(\mathbf{x}-\hat{\mathbf{c}}_t^i)]^p)$$
$$+\sum_{\theta=0:\epsilon_m:1}(\|\mathbf{x}-(\theta\hat{\zeta}_t^{i,1}+(1-\theta)\hat{\zeta}_t^{i,2})\|_2^{-2}+\|\mathbf{x}-(\theta\hat{\xi}_t^{i,1}+(1-\theta)\hat{\xi}_t^{i,2})\|_2^{-2})],$$

where $\hat{\mathbf{c}}_t^i$, $\hat{\mathbf{P}}_t^i$, $\hat{\zeta}_t^{i,1}$, $\hat{\zeta}_t^{i,2}$, $\hat{\xi}_t^{i,1}$ and $\hat{\xi}_t^{i,2}$ are the estimated parameters of the $i$th obstacle at time step $t$ given by a separate estimator that tracks the obstacles. Note that if the $i$th obstacle is moving but not changing shape, then at time $0 > t$, $\hat{\mathbf{c}}_0^i = \hat{\mathbf{c}}_t^i + \hat{\mathbf{v}}^i(0-t)$ and $\hat{\mathbf{P}}_0^i = R_{\hat{\alpha}}^i\hat{\mathbf{P}}_t^i$ where $\hat{\mathbf{v}}^i$ is a constant estimated velocity vector, and $R_{\hat{\alpha}}^i$ is an estimated rotation matrix by $\hat{\alpha}$ degrees. However, if there is also a change of shape in the obstacle or appearance of new obstacles, we run the MVEE algorithm to find the parameters of that obstacle. For our planning purposes, we assume there is a separate estimator that tracks and estimates the obstacles' parameters, and our planner only uses the results obtained by that tracker to find the optimized trajectory. Moreover, since the algorithm is implemented in an RHC fashion, if there is a change in the estimates of the obstacles, for the next step the optimization uses the new estimates of the obstacle parameters. Moreover, the obstacle cost is modified as follows:

$$\text{cost}_{obst}(\mathbf{x}_{t1},\mathbf{x}_{t2},t):=\int_{\mathbf{x}_{t1}}^{\mathbf{x}_{t2}}\Phi_t^{(\hat{\mathcal{P}}_t,\hat{\mathcal{C}}_t)}(\mathbf{x}')d\mathbf{x}'.$$

**Problem 22 (Dynamic environment)** *For a linear system, given $\{\mathbf{x}_0^i\}_{i=1}^k$, $\mathbf{x}_g$ and estimates of the obstacle parameters for the entire lookahead horizon $\{(\hat{\mathcal{P}}_t,\hat{\mathcal{C}}_t)\}_{t=0}^{K+1}$,* *the dynamic environment problem is defined as:*

$$\min_{\mathbf{u}_{0:K-1}} \sum_{t=1}^{K} [\text{cost}_{info}(\mathbf{x}_t^p) + \text{cost}_{eff}(\mathbf{u}_{t-1}) + \text{cost}_{obst}(\mathbf{x}_{t-1}^p, \mathbf{x}_t^p, t-1)]$$
$$s.t.\ \ \mathbf{x}_K^p = \mathbf{x}_g. \tag{11.13}$$

If there is a sudden appearance of a new obstacle in part of the trajectory, only that part of the trajectory is changed provided there is still a feasible path between the two points immediately outside and on the other side of that obstacle. Otherwise, the entire algorithm runs again from the current state to the goal state. It should be added that, unlike a static environment, in a stochastic problem with dynamic environment, unless the planning horizon is very small, there is not much that can be said regarding the homotopy paths discussed in Section 11.3.4. This is an ongoing research.

Now that we have provided our solution for all the three cases, we proceed to discuss the implementation strategy.

### *11.3.6 Receding Horizon Control (RHC) Implementation*

The overall feedback control loop is shown in Fig. 11.1. The system initiates from a non-Gaussian distribution in the feasible state space that constitutes the initial belief. In the case of a dynamic environment, the most complicated case of our problems, given the current belief, $b_t$, estimates of the obstacles' parameters, $\{(\hat{\mathcal{P}}_t, \hat{\mathcal{C}}_t)\}_{t=0}^{0+K+1}$, lookahead time horizon, $K$, and the goal state, $\mathbf{x}_g$, the RHC policy function $\pi : \mathbb{B} \times \mathbb{R}^{n_x \times n_b \times (K+1)} \times \mathbb{R}^{n_x^2 \times n_b \times (K+1)} \times \mathbb{R} \times \mathbb{X} \rightarrow \mathbb{U}$ generates an optimal action $\mathbf{u}_t = \pi(b_t, \hat{\mathcal{P}}_{t:t+K+1}, \hat{\mathcal{C}}_{t:t+K+1}, K, \mathbf{x_g})$, which is the first element of the open-loop optimal sequence of actions generated in different variants of problem (1). The agent executes $\mathbf{u}_t$ transitioning the state of the system from $\mathbf{x}_t$ in $\mathbf{x}_{t+1}$ where a new observation $\mathbf{z}_{t+1}$ is obtained by the sensors. The estimator updates the belief as $b_{t+1} = \tau(b_t, \mathbf{u}_t, \mathbf{z}_{t+1})$ and the policy is fed the updated belief to close the loop. Meanwhile, on another separate loop, the obstacle trackers measure the current state of the obstacles and the obstacle parameter estimators obtain the estimates of the obstacles. As mentioned above, the estimates are fed into the policy function immediately before the controller plans its next action. In the case of the static environment, the policy function is fed the parameters of the obstacles that remain the same for the entire horizon. Similarly, in the case of a convex environment, the general boundaries and convex constraints take the place of the obstacle parameters in the planning problem.

*Stopping execution:* The algorithm stops when the probability of reaching the goal, calculated as the area under the belief density over the goal region, exceeds a

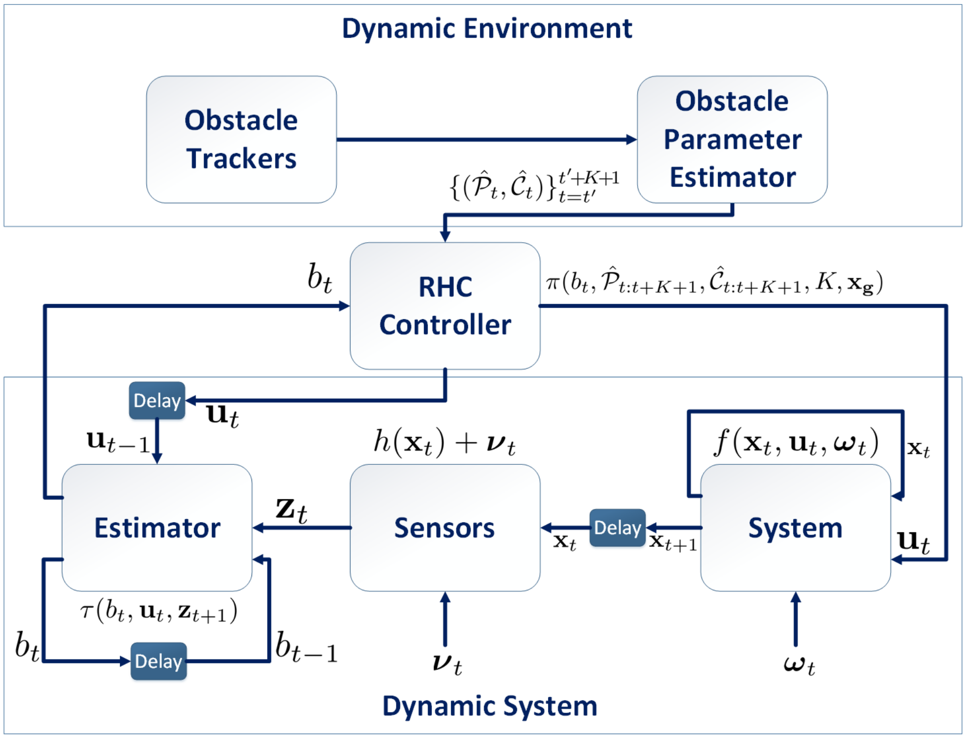

Dynamic Environment
Obstacle Trackers
Obstacle Parameter Estimator
$\{(\hat{\mathcal{P}}_t, \hat{\mathcal{C}}_t)\}_{t=t'}^{t'+K+1}$
$b_t$
RHC Controller
$\pi(b_t, \hat{\mathcal{P}}_{t:t+K+1}, \hat{\mathcal{C}}_{t:t+K+1}, K, \mathbf{x_g})$
Delay
$\mathbf{u}_t$
$\mathbf{u}_{t-1}$
$h(\mathbf{x}_t) + \boldsymbol{\nu}_t$
$f(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\omega}_t)$
$\mathbf{x}_t$
Estimator
$\mathbf{z}_t$
Sensors
$\mathbf{x}_t$
Delay
$\mathbf{x}_{t+1}$
System
$\mathbf{u}_t$
$\tau(b_t, \mathbf{u}_t, \mathbf{z}_{t+1})$
$b_t$
Delay
$b_{t-1}$
$\boldsymbol{\nu}_t$
$\boldsymbol{\omega}_t$
Dynamic System

Figure 11.1: The overall feedback control loop.

predefined value [61].

The planning algorithm is in Algorithm 4.

```
Algorithm 4: Planning Algorithm
Input: Initial belief state $\mathbf{b}_0$, Goal state $\mathbf{x}_g$, Planning horizon $K$, Belief
       dynamics $\tau$, Obstacle parameters $\{(\hat{\mathcal{P}}_t, \hat{\mathcal{C}}_t)\}_{t=0}^{K+1}$
while $\mathcal{P}(\mathbf{b}_t, r, \mathbf{x}_g) \leq \breve{w}_{th}$ do
    $\mathbf{u}_t \leftarrow \pi(\mathbf{b}_t, \hat{\mathcal{P}}_{t:t+K+1}, \hat{\mathcal{C}}_{t:t+K+1}, K, \mathbf{x_g})$;
    execute $\mathbf{u}_t$, perceive $\mathbf{z}_t$;
    $\mathbf{b}_{t+1}(\mathbf{x}) \leftarrow \boldsymbol{\tau}_t(\mathbf{b}_t(\mathbf{x}), \mathbf{u}_t, \mathbf{z}_t)$;
end
```

### *11.3.7 A Discussion and Comparison on Complexity*

*Comparison of our method with traditional approaches:* Figure 11.2 graphically compares our method with traditional methods in the literature that tackle the *open-loop* problem. In order to perform the filtering equation, a previous belief and action, and a current observation are required. In the planning stage, where the controller obtains the best sequence of future actions, a current belief is given; however, all that is known about the future observation is a likelihood distribution. As shown in this figure, in classic methods, the initial belief is propagated using finitely many samples of the observation obtained from the likelihood distribution. Therefore, a decision tree on the future predicted beliefs is constructed so that the optimizer can obtain the best action for each height of the tree. Overall, the first method is computationally intensive. In the second popular class of methods, only the most likely observation is utilized to perform the filtering equations and propagate the belief. This method can be less accurate than the latter, and although it provides a less expensive optimization, the filtering equation is part of the optimization constraints

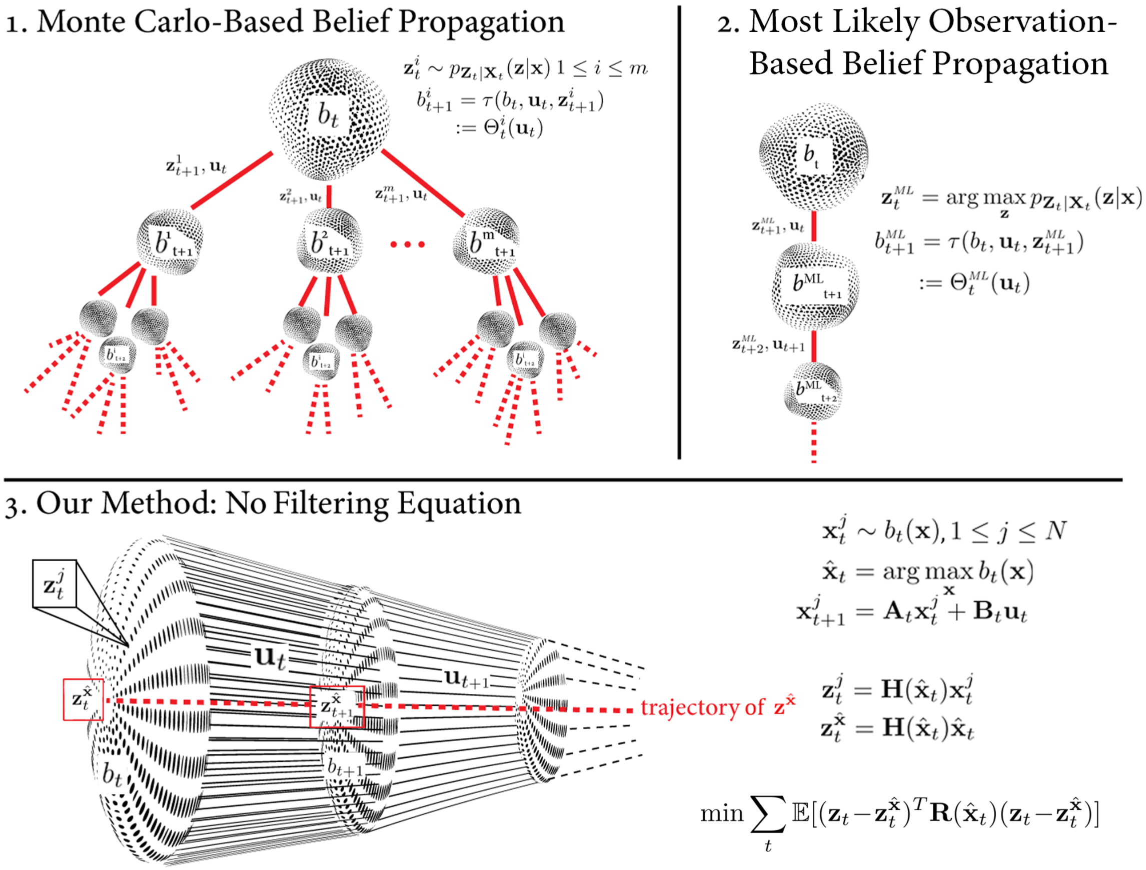

1. Monte Carlo-Based Belief Propagation
$\mathbf{z}_t^i \sim p_{\mathbf{Z}_t|\mathbf{X}_t}(\mathbf{z}|\mathbf{x})\ 1 \le i \le m$
$b_{t+1}^i = \tau(b_t, \mathbf{u}_t, \mathbf{z}_{t+1}^i)$
$:= \Theta_t^i(\mathbf{u}_t)$
$b_t$
$\mathbf{z}_{t+1}^1, \mathbf{u}_t$
$\mathbf{z}_{t+1}^2, \mathbf{u}_t$
$\mathbf{z}_{t+1}^m, \mathbf{u}_t$
$b^1_{t+1}$
$b^2_{t+1}$
$b^m_{t+1}$
2. Most Likely Observation-Based Belief Propagation
$b_t$
$\mathbf{z}_t^{ML} = \arg\max_{\mathbf{z}} p_{\mathbf{Z}_t|\mathbf{X}_t}(\mathbf{z}|\mathbf{x})$
$b_{t+1}^{ML} = \tau(b_t, \mathbf{u}_t, \mathbf{z}_{t+1}^{ML})$
$:= \Theta_t^{ML}(\mathbf{u}_t)$
$\mathbf{z}_{t+1}^{ML}, \mathbf{u}_t$
$b^{ML}_{t+1}$
$\mathbf{z}_{t+2}^{ML}, \mathbf{u}_{t+1}$
$b^{ML}_{t+2}$
3. Our Method: No Filtering Equation
$\mathbf{x}_t^j \sim b_t(\mathbf{x}), 1 \le j \le N$
$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}} b_t(\mathbf{x})$
$\mathbf{x}_{t+1}^j = \mathbf{A}_t\mathbf{x}_t^j + \mathbf{B}_t\mathbf{u}_t$
$\mathbf{z}_t^j$
$\mathbf{u}_t$
$\mathbf{u}_{t+1}$
$\mathbf{z}_t^{\hat{\mathbf{x}}}$
$\mathbf{z}_{t+1}^{\hat{\mathbf{x}}}$
trajectory of $\mathbf{z}^{\hat{\mathbf{x}}}$
$b_t$
$b_{t+1}$
$\mathbf{z}_t^j = \mathbf{H}(\hat{\mathbf{x}}_t)\mathbf{x}_t^j$
$\mathbf{z}_t^{\hat{\mathbf{x}}} = \mathbf{H}(\hat{\mathbf{x}}_t)\hat{\mathbf{x}}_t$
$\min \sum_t \mathbb{E}[(\mathbf{z}_t - \mathbf{z}_t^{\hat{\mathbf{x}}})^T \mathbf{R}(\hat{\mathbf{x}}_t)(\mathbf{z}_t - \mathbf{z}_t^{\hat{\mathbf{x}}})]$

Figure 11.2: Comparison of method of this chapter with traditional belief propagation methods.

which makes it computationally intense. However, in the proposed method of this chapter, once the samples of the initial belief are propagated via the predicted model of the system, they are converted into observation particles by a proper usage of the observation model. Thus, a rope-like bundle of propagated observation particle strands is constructed using the initial belief samples and with the advantage of a particular defined cost function, the dispersion in the strands is minimized. Hence, the optimization not only morphs the rope towards regions that provide observations, but also seeks to compress the bundle towards the end of the horizon. As a result of reduced uncertainty in observation bundle, the belief itself shrinks and the same results are obtained without performing the filtering equation. Therefore, using this idea, the main computational burden of the problem is broken and the much cheaper optimization yields the desired results. We provide more details in the sequel.

*Computational complexity:* The core optimization problem in a convex environment as defined in problem (20) is a convex program that does not necessarily require an initial solution. The number of decision variables is $Kn_u$, and there is one linear equality constraint, plus, the robots saturation inequality constraints, which can be $Kn_u$ at most. Therefore, the optimization involves the minimum number of decision variables. Let us assume for simplicity that the sizes of $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{z}$ vectors are all $O(n)$. Thus, utilizing a common method, such as center of gravity for convex optimization [190] to obtain a *globally optimal* solution with $\epsilon$ confidence, the algorithm requires $\Omega(Knlog(1/\epsilon))$ calls to the oracle [191]. On the other hand, in equation (11.8), $\tilde{\mathbf{A}}^T_{0:t-1}\mathbf{W}(\mathbf{x}^p_t)\tilde{\mathbf{A}}_{0:t-1}$ requires a multiplication of $O(n) \times O(n)$ matrices $O(K)$ times, which takes $O(Kn^3)$. However, the multiplication of the vectors $(\mathbf{x}^i_0 - \mathbf{x}^p_0)$ to a $\mathbb{R}^{n \times n}$ matrix involves $O(Nn^2)$ time. The outer sum also takes $K$ time. All the other operations, such as calculation of $\mathbf{x}^p_t$ and constraints, take less time. Hence, the time complexity of the computations is $O(Kn^3 + Nn^2)$. In LQG-based belief space methods that construct a trajectory, the method described in [128] involves a non-convex optimization, which takes $O(Kn^6)$ computations with a second-order convergence rate to a locally optimal solution. Another RHC-based method particle filter-based method is described in [61], where the core problem is a convex problem in $Kn_u + N$ number of decision variables with $N(K-1)+1$ number of inequality constraints. The algorithm assumes a linear process model with Gaussian noise and a linear measurement model with a Gaussian noise whose covariance is state-dependent. The solution is categorized in the second class of methods in Fig. 11.2. Moreover, to include more than one observation source, the algorithm requires a modification with integer programming, such that at each time step, there could only be one observation. Although the analysis of time complexity is not given, to the best of our knowledge we assess it to be $O(NK(Kn^3 + Nn^2))$ without integer programming. Moreover, the near-convergence needs $\Omega((N + Kn)log(1/\epsilon))$ calls.

In the presence of obstacles, the size of our optimization does not change; however, the rate of convergence reduces to the rate of gradient descent methods. Furthermore, the solution becomes locally optimal starting with an infeasible solution whose immediate gradient is not towards the local minima of the obstacles. Theoretically, if the $\epsilon_m$ of OBF tends to zero, there is no local minima of the barriers; nevertheless, practically, starting from a semi-feasible trajectory, a tuned optimization results in convergence to a locally optimal feasible solution. In [128], in the presence of obstacles, the convergence rate and computational cost do not change, but the (tuned) optimization must start with a feasible path. In [61], obstacles are modeled with a chance-constrained method that involves the introduction of additional variables and integer programming with iterative applications of the algorithm. This limits the scalability of that method in complex environments.

## 11.4 Simulation Results

In this section, we show some applications for the method of this section. We perform all our simulations in MATLAB 2015b in a 2.90 GHz CORE i7 machine with dual core technology and 8 GB of RAM. First, we perform a comparison test on an example from the literature and analyze the solutions of two algorithms with various parameters. Then, we introduce a scenario that consists of guiding a robot between two walls. Our last experiment is a simulation where a robot is in a complex scenario in a house with several features to localize with respect to and reach a goal. Next, we perform a comparative simulation between trajectories in different homotopy classes in which we compare the results of our algorithm for an static environment with and without information sources. Then, we perform an experiment for a KUKA youBot in static environment. Finally, we perform two simulations for dynamic environments. In the first one, obstacles only move in simple translational movements, and in the second, an object moves in a helix that makes the robot escape from its trajectory in a more complex scenario.

### *11.4.1 Comparison Test in a Convex Scenario*

In this experiment, we consider the light-dark example introduced in [109]. We compare our results with the algorithm presented in [109]. Since we did not have access to the author's code, we implemented the method of [109] in MATLAB to the best of our ability. Note that in this scenario, we assume that there is no obstacle in the environment. It is important to note that essentially the two methods are different from each other, but we solve the same problem for the same systems and same initial and final states. Therefore, the optimization tuning parameters are different and have different meanings. The state, observation and action spaces are two-dimensional continuous spaces. The process model is linear with $\mathbf{A} = \mathbf{B} = \mathbf{I}_2$, and the observation model is linear with nonlinear observation covariance, modeled as $\mathbf{R}(x) = \text{diag}(1/(2x_1 + 1), 1/(2x_1 + 1))$, where $x_1 > 0$ is the first element of state. Therefore, as the robot gets further to bigger values of $x_1$ it can localize better with less noisy observations. This is shown in Fig. 11.3 with lighter background on th right side. One can verify that the problem is convex in both methods (with different shapes of cost functions). Figure 11.3 shows the results of the optimized trajectory for time 0 with 1000 particles and a time horizon of 20. Moreover, to avoid the control saturation, we add a constraint to bound the control inpu0s magnitude at each step to 3.16. The initial distribution is a mixture of two Gaussians with equal variances of 0.0625 and means at (1.75, 0) and (2, 0.5). The solid line shows the results for our problem with $\mathbf{V}_t^u = 0.065$. It should be noted that, in our simulation, changes $\mathbf{V}_t^u$ does not impose unexpected behavior in the trajectory. Rather, by increasing the values of $\mathbf{V}_t^u$, the agent acts more conservatively in terms of the consumed energy effort.

*Sensitivity of solution to number of particles:* We increase the number of particles from 50 to 1000, 10000, and 100000 particles and analyze the optimization size and required time for the optimization. In our method, by increasing the number of particles, the optimization vector size does not increase. Neither are additional constraints introduced by increasing the number of particles. Therefore, as shown in Table 11.1 the required time for optimization does not increase significantly. However, in [109], the optimization vector size is dependent on the number of particles, particularly, it is equal to $(Kn_u + N)$, while in our method, it is only $Kn_u$. Moreover, in their method, upon addition of one particle, $K$ new inequality constraints are added to the optimization problem, whereas in our method, there is no such constraint and the number of optimization constraints is independent from the number of samples. The results of Table 11.1 show that our method is scalable in the number of particles. As stated in the Table, for $N = 10000$ and $N = 100000$, we could not perform the optimization for the method in [109] because of large memory requirement.

*Sensitivity of solution to time horizon:* Lastly, we perform the optimization for lookahead time horizon $K = 10, 20, 50$ and 100 and report the required time in table. Once again, since the number of optimization variables is $Kn_u$ which is $2K$, and there is no added constraint for addition of time horizon, the optimization time does not explode in our method. Whereas, in [109], increasing the time horizon, increases the solution time significantly. The results reflected in table 11.1 show that our method is scalable with long time horizon as well. However, for $K = 50$ and $K = 100$, we could not perform the optimization for method of [109] because of large memory requirement.

### *11.4.2 Robot Within Two Walls*

In this section, we simulate a case where there are non-convex constraints in the state space. Figure 11.4 depicts the results in a case where the system starts with a distribution about its initial state and wants to reach the goal state while minimizing the localization error and spending low energy. The green and red lines show the solution of the convex problem where there is no walls, and the problem with added walls, respectively. As it is seen, there are three information sources in that are shown with lighter spots in Fig. 11.4. The observation model is range based as described in example 1. To obtain the green trajectory, the convex optimization problem (which is initialized with an arbitrary solution) is solved. Then, the green trajectory (which is not feasible for the case with walls) is used as the initial trajectory for the optimization with OPF to obtain the red trajectory which avoids the walls, as well.

Table 11.1: The results of comparative simulations for several time horizons and particle numbers in a convex light-dark scenario.

| | Time horizon ($K$) | 20 | | | | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|---|---|---|---|
| | Number of Particles ($N$) | 100 | 1000 | 10000 | 100000 | 1000 | | | |
| Our Method | Time (s) | 0.24 | 0.33 | 1.11 | 10.37 | 0.16 | 0.33 | 2.30 | 9.22 |
| | # of Iterations | 288 | 288 | 288 | 288 | 170 | 288 | 1013 | 2215 |
| | Function Tolerance | 2e-03 | 2e-03 | 2e-03 | 2e-03 | 2e-03 | 2e-03 | 2e-03 | 2e-03 |
| | Constraint Tolerance | 5.551e-16 | 8.882e-16 | 5.551e-16 | 2.331e-15 | 1.110e-15 | 8.882e-16 | 3.839e-11 | 1.883e-11 |
| | # of Opt. Vars.[†] ($Kn_u$) | 40 | 40 | 40 | 40 | 20 | 40 | 100 | 200 |
| | # of Constrs.[†] (1) | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Method of [109] | Time (s) | 49.0 | 311.32 | * | * | 80.22 | 311.32 | * | * |
| | # of Iterations | 40000 | 40000 | | | 40000 | 40000 | | |
| | Function Tolerance | 2e-02 | 2e-02 | | | 2e-02 | 2e-02 | | |
| | Constraint Tolerance | 3.509e-04 | 5.853e-04 | | | 5.671e-04 | 5.853e-04 | | |
| | Required Memory (GB) | | | 15.0 | 1490.7 | | | 37.6 | 76.0 |
| | # of Opt. Vars.[†] ($Kn_u$+$N$) | 140 | 1040 | 10,040 | 100,040 | 1020 | 1040 | 1100 | 1200 |
| | # of Constrs.[†] ($N(K$–1)+1) | 1901 | 19,001 | 190,001 | 1,900,001 | 9001 | 19,001 | 49,001 | 99,001 |

*: Unable to allocate enough memory to solve the problem.

[†]: '# of Opt. Vars.' specifies the number of optimization variables, and '# of Constrs.' specifies the number of optimization problem's constraints.

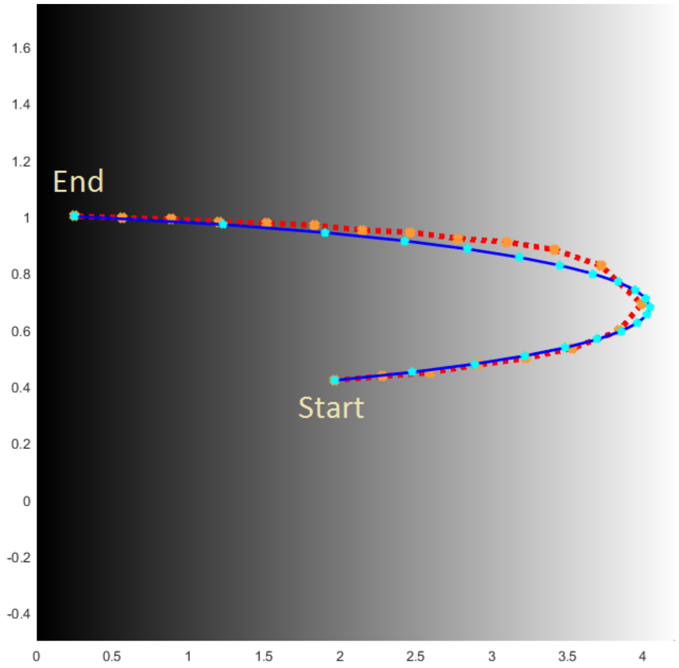


Figure 11.3: Light-dark example. Lighter states on the right signify lesser observation noise. The solid blue and red dotted lines show the results of our method and the implementation of [109], respectively. The axes' units are in meters.

### *11.4.3 Complex Scenario in a Room*

*Robot in a house:* Figure 11.5 depicts the results in two cases where the objective is similar to the previous example. In the first case, the robot is put in a room and wants to reach a room in the other side of the house. Given an initial trajectory, shown by red dots, the optimization provides the optimized trajectory that seeks for the information sources in every house, and the penalty functions perform the task of keeping the robot away from the obstacles. In this case, the lookahead time horizon is set to $K = 100$. In the second case, the start and final goal of the robot is in one room, and therefore, the optimization can solve the problem with any arbitrary trajectory in that room like the straight line.

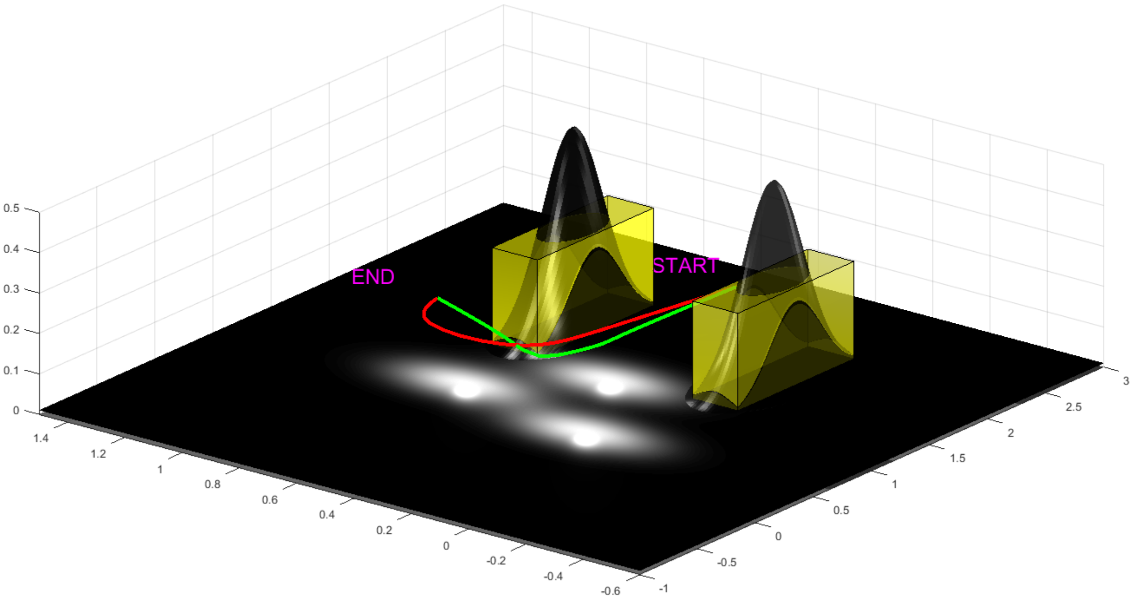


Figure 11.4: Robot within two walls. The OPF is visualized within the walls. The green and red lines show the results for optimization with and without considering the walls. The axes' units are in meters.

### *11.4.4 Comparison Test Between Homotopy Classes*

Figure 11.6 shows an environment with three obstacles forming a connected obstacle. The banned areas are enclosed with three MVEEs. In this experiment, we use the visibility graph to find initial trajectories in different homotopy classes. Moreover, instead of using the polygons, we use the ellipsoids that enclose them. Since our optimization utilizes a gradient descent method, we only consider the straight lines between the nodes and ignore the collision of the straight line with the ellipsoid that the node is lying on. This increases the speed of finding the visibility graph and coupled with optimization over the output paths, the minor collisions do not hurt the algorithm.

Next, each of the two paths is discretized to satisfy the tuning properties described in Section 11.3.4. They are then fed into the optimization function $\pi$ to produce the

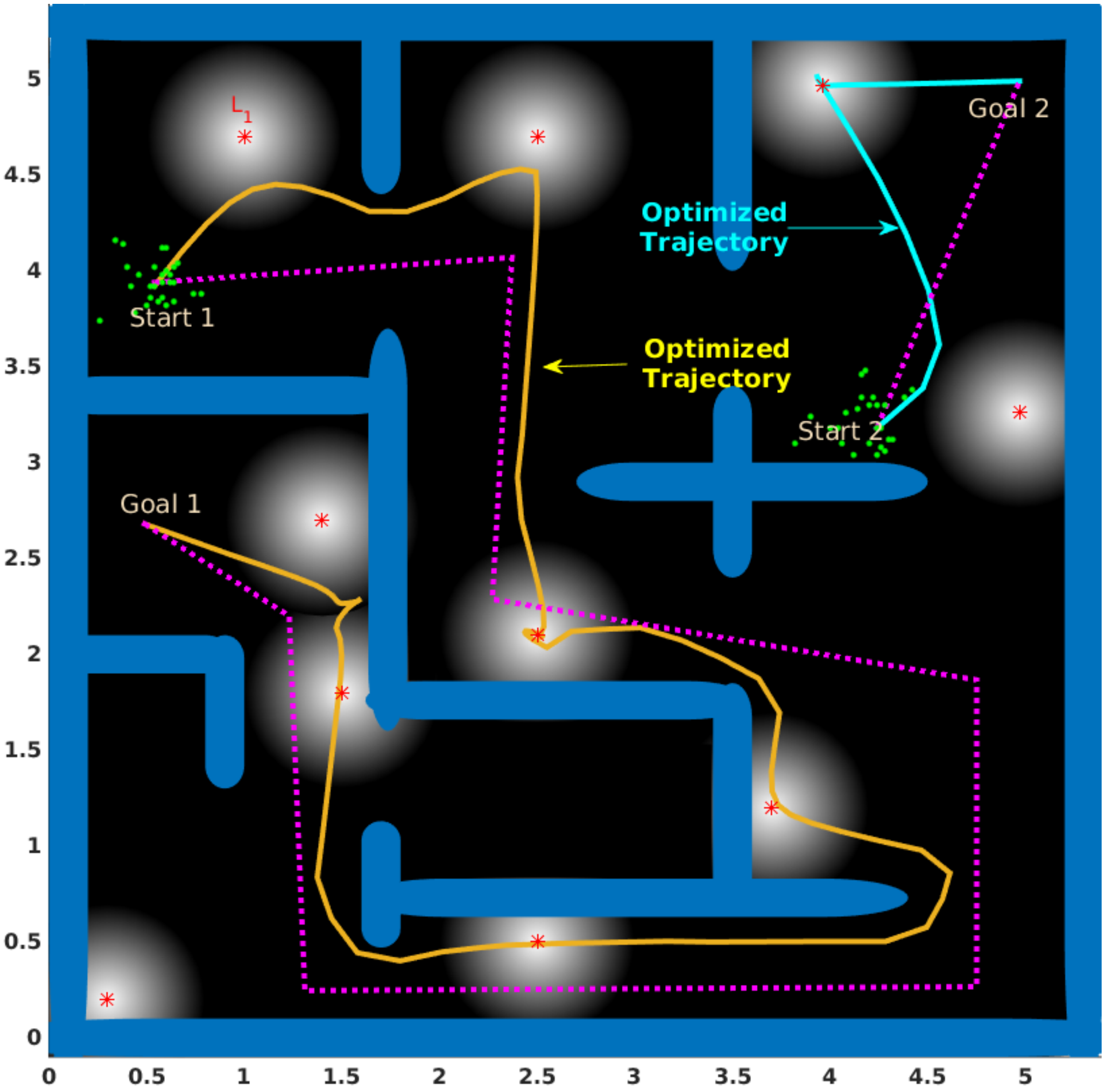


Figure 11.5: A holonomic system in a complex scenario. Solid lines show the optimal trajectories, dotted lines show the initial trajectory, for two different scenarios. The longer trajectory includes obstacles, and the other, no obstacles. The dots around the start points show the initial particles. Landmarks are marked as stars and information is coded with color (lighter means more information). Lookahead time horizon for the longer and shorter trajectories is 100 seconds and 30 seconds, respectively. The axes' units are in meters.

optimized smooth collision-free paths. We have produced two sets of results; in the first set, we do not consider the cost of information (as if we are considering the motion planning problem to generate smooth collision-free paths); in the second, we add a landmark as the information source, and consider the optimization with cost of information, to compare the results. As seen in Fig. 11.7d, existence of the landmarks changes the paths of the robot, such that the robot visits them to reduce its uncertainty and then continues its path towards the goal state.

### *11.4.5 KUKA YouBot*

In this section, we use the kinematics equations of KUKA youBot base as described in [192]. Particularly, the state vector can be described by a 3D vector, $\mathbf{x} = [\mathbf{x}_{\mathrm{x}}, \mathbf{x}_{\mathrm{y}}, \mathbf{x}_{\theta}]^T$, describing the position and heading of the robot base, and $\mathbf{x} \in SO(3)$. The control consists of the velocities of the four wheels. It can be shown that the discrete motion model can be written as $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) = \mathbf{x}_t + \mathbf{B}\mathbf{u}_t dt + \mathbf{G}\mathbf{w}_t\sqrt{dt}$, where $\mathbf{B}$ and $\mathbf{G}$ are appropriate constant matrices whose elements depend on the dimensions of the robot as indicated in [199], and $dt$ is the time-discretization period. Inspired by [193], we model the robot with a configuration of a set of points which

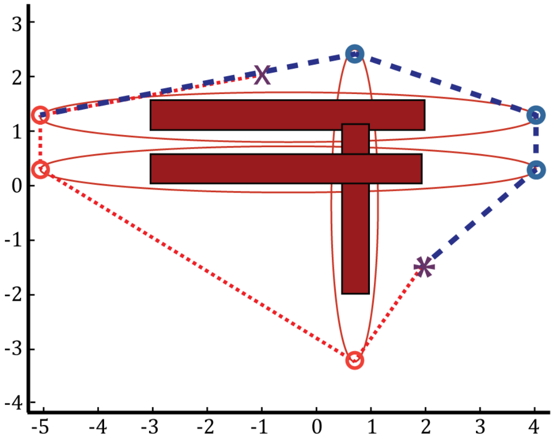
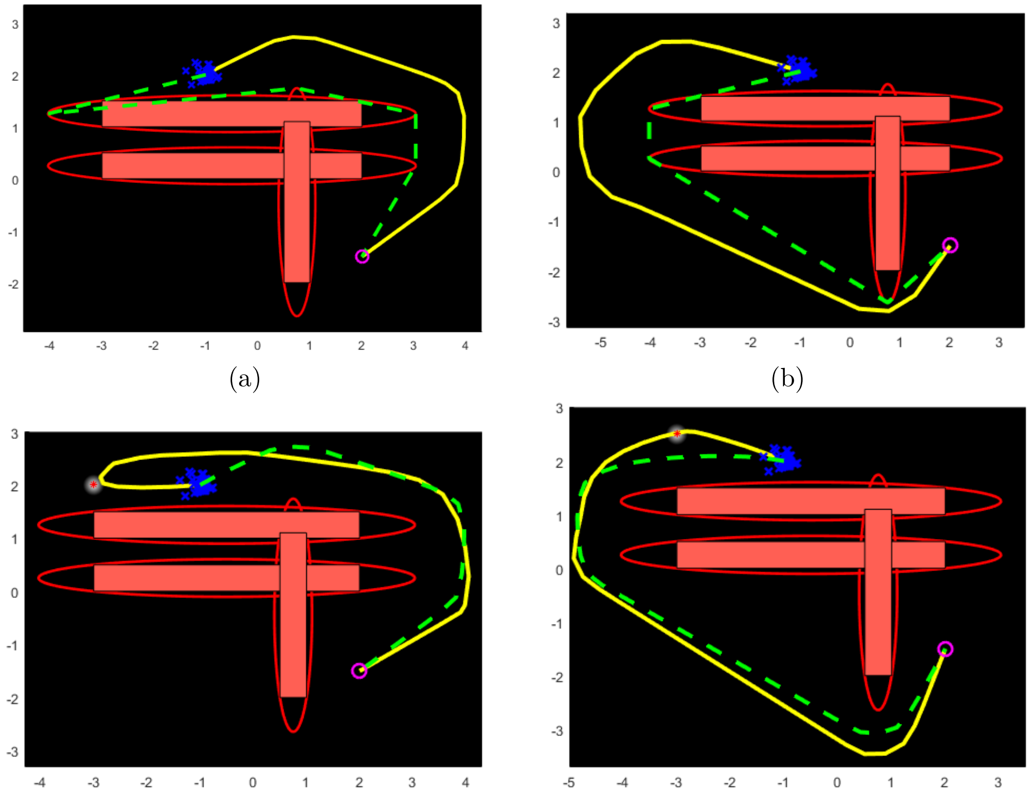


Figure 11.6: Modified visibility graph. There are two homotopy classes between the start and goal points that are found using the visibility graph and are indicated as the red dotted and blue dashed paths. The axes' units are in meters.



(a)

(b)

(c)

(d)

Figure 11.7: Comparison of paths in different homotopy classes. Cases (a) and (b) show the resulting paths generated by optimizing without considering the information sources, whereas cases (c) and (d) consider information sources. The axes' units are in meters.

represent the centers of the balls that cover the body of the robot. In our method, we cover the robot with two balls whose radii are proportional to the width of the robot. 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. The observation model is a range and bearing based model where the corresponding elements of the $\mathbf{R}$ matrix are chosen to be $\|(\mathbf{x}_{\mathrm{x}} - L_x, \mathbf{x}_{\mathrm{y}} - L_y)\|_2^2$ for all observations so as to have the desired features described in Lemma 9. $(L_x, L_y)$ represents the coordinates of a landmark. The results depicted in Fig. 11.8, show that the planned trajectory avoids entering the banned regions bordered by the ellipsoids, so that the robot itself avoids colliding with the three obstacles.

### *11.4.6 Dynamic Environment*

In this scenario, we simulate a case where there are four objects, starting from a common position and moving in different directions downwards and towards the right of the map. The robots starts from a distribution whose mean is at (0,0), and wishes to reach the goal state (2,2) with high probability. As seen in Fig. 11.9, at the beginning of its trajectory, the robot head towards the landmark at (1, 0.5), and as the moving obstacles get closer, it changes its direction to bypass the objects in the opposite direction. In this scenario, the initial trajectory is just the straight line between the most probable initial location of the robot and the final destination shown in the figure with green dashed line, with the planned trajectory of the robot shown as a solid yellow line.

In another scenario shown in Fig. 11.10, an object is moving in a spiral path shown in Fig. 11.10h with the robot trying to avoid colliding with the obstacle, spending most of its time near the information source and reaching the goal in a safe, short and smooth path.

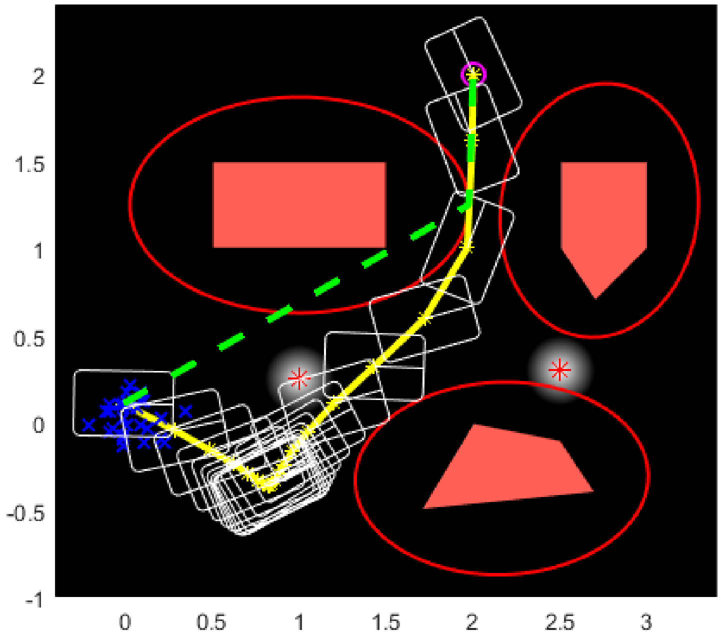
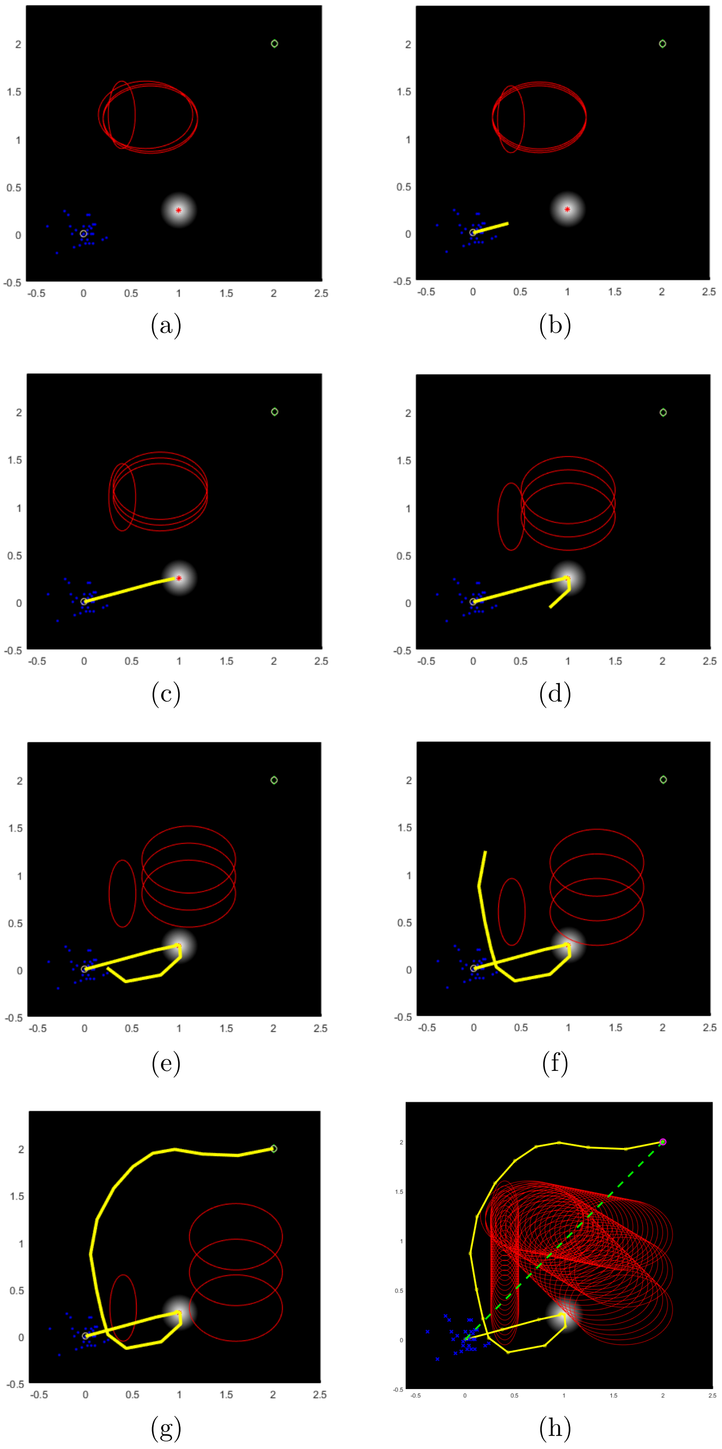
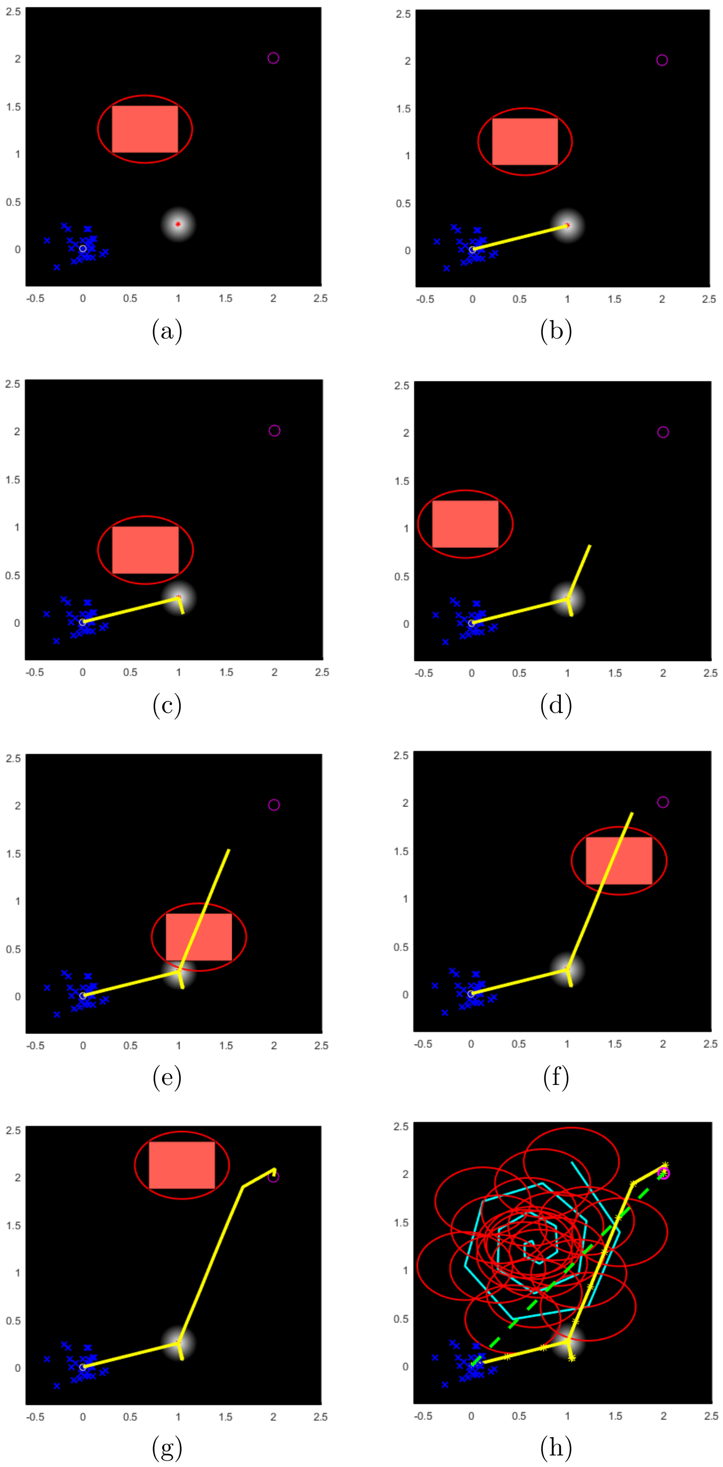


Figure 11.8: Controlling a youBo0s base. There are three obstacles and two landmarks. The robot base is shown by a rectangle with a line at the heading. Initial and planned trajectories are depicted by dashed and solid lines, respectively. The axes' units are in meters.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 11.9: Dynamic environment. The robot heads towards the landmark to reduce its uncertainty, and avoids the moving objects by changing its path to point in a direction opposite to the objects. The axes' units are in meters.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 11.10: Moving object. The robot spends most of its time near the information source and avoids the object, which is moving in a spiral path, and heads towards the goal region safely. The axes' units are in meters.

# 12. CONCLUDING REMARKS

In this chapter, we review the major contributions of this work and discuss its future directions and extensions.

## 12.1 Contributions

*The decoupling principle:* Perhaps the main contribution is this Dissertation is the introduction and proof of the decoupling principle. Since the introduction of the HJB equation for solving the stochastic control problem, there have been many works on providing tractable solutions to obtain policies that are both tractable and have theoretical guarantees. There are very sparse number of results with such properties, and where they exist, they do often consider very limiting assumptions that generally may not be satisfied in practical systems. Except for the linear Gaussian systems where the theory is sound and perfect, for nonlinear systems the general methodology and importantly, the popular methods have been heuristic strategies. This Dissertation has tried its best to avoid unnecessary heuristics. Particularly, the theoretical part of the Dissertation, i.e., Part (I), is mathematically rigorous. Yet, the solutions that are resulted from the four variations of the decoupling principle are all computationally tractable, while retaining the theoretical guarantees. In this retrospect, this is yet the most generic result of this type that has both the tractability feature, similar to the heuristic method, as well as the rigor, similar to the theoretical solution.

*The decoupling principle in a word:* The decoupling principle provides the conditions under which the design of the nominal trajectory of the system and a decentralized feedback policy can be near-optimally decoupled from each other. This result considers the fully- and partially-observed single- and multi-agent situations for nonlinear stochastic systems with additive Gaussian perturbations.

*Linear Gaussian approximation:* For nonlinear systems, linearization has always been “the” practical way to go. However, there has not been a result that has quantified the correctness of this approach. This Dissertations provides an answer to this problem through the decoupling principle for various situations.

*Decentralized solution:* Theoretically sound solution for multi-agent systems results in the application of the HJB equation in a centralized manner. The decoupling principle, provides a theoretically sound, while also tractable, decentralized solution for a multi-agent system.

*Belief space planning:* One of the most important robotic problems is tackled with rigorous tractable algorithms. The T-LQG, and MT-LQG are the partially-observed variants of the T-LQR and MT-LQR algorithms for fully-observed situations. These algorithms are the resultant methods of the decoupling principle.

*Non-convex constraints:* Our methods consider non-convex time-varying dynamic environments and show tractable and reliable solutions or various complex situations. The obstacle penalty function method provides an easy-to-handle method of incorporating the non-convex constraints into the optimization problems.

*The observability Gramian:* While heuristic solutions are helpful, many of them have pitfalls. We found and analyzed the observability Gramian’s shortcomings for robotic path planning and estimation. We showed that optimizing measures of the observability Gramian as a surrogate for estimation performance may provide irrelevant or misleading trajectories for planning under observation uncertainty.

*Non-Gaussian particle-filter-based planning:* Finally, we utilize the insights provided from the results of the decoupling principle for Gaussian perturbations to obtain heuristic solutions for non-Gaussian additive perturbations utilizing particle filters. We also provide a convexified belief space planning method using an MPC strategy for robotic systems with nonlinear measurement models.

## 12.2 Future Extensions

This Dissertation provides various possible directions to continue the advancement of the approach to much advanced situations, such as continuous-time models and models with non-Gaussian perturbations. It also provides a theoretically sound, and yet implementable, benchmark solution where other the performance and correctness of other heuristic methods can be tested and analyzed. Moreover, the solutions of this research can be utilized in other application areas, such as the solution of systems with black-box unknown dynamics models and reinforcement learning techniques for fully- and partially-observed systems with partial differential equation process models, where the solution space is of high degrees of freedom. This line of research has started to blossom its initial results. Last, it is possible to provide further enhancement of the algorithms by considering higher order expansions. Our future work will address some of these issues.

# REFERENCES

[1] M. Rafieisakhaei, A. Tamjidi, S. Chakravorty, and P. Kumar, "Feedback motion planning under non-gaussian uncertainty and non-convex state constraints," *arXiv preprint arXiv:1511.05186*, 2015.

[2] P. R. Kumar and P. P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control.* Englewood Cliffs, NJ: Prentice-Hall, 1986.

[3] K. Astrom, "Optimal control of markov decision processes with incomplete state estimation," *Journal of Mathematical Analysis and Applications*, vol. 10, pp. 174–205, 1965.

[4] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable markov processes over a finite horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, 1973.

[5] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[6] M. I. Freidlin and A. D. Wentzell, *Random Perturbations*, pp. 15–43. New York, NY: Springer US, 1984.

[7] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, MA, 1995.

[8] H. Kushner and P. G. Dupuis, *Numerical methods for stochastic control problems in continuous time*, vol. 24. Springer Science & Business Media, 2013.

[9] R. Bellman, *Dynamic Programming.* Princeton, NJ, USA: Princeton University Press, 1 ed., 1957.

[10] C.-S. Chow and J. N. Tsitsiklis, “The complexity of dynamic programming,” *Journal of complexity*, vol. 5, no. 4, pp. 466–488, 1989.

[11] E. J. Sondik, “The optimal control of partially observable markov processes,” *PhD thesis, Stanford University*, 1971.

[12] D. Bertsekas, *Dynamic Programming and Optimal Control: 3rd Ed.* Athena Scientific, 2007.

[13] D. Bertsekas, *Dynamic Programming and Stochastic Control.* Academic Press, 1976.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics.* MIT Press, 2005.

[15] 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.

[16] J. Pineau, G. Gordon, S. Thrun, *et al.*, “Point-based value iteration: An anytime algorithm for pomdps,” in *IJCAI*, vol. 3, pp. 1025–1032, 2003.

[17] G. Shani, R. I. Brafman, and S. E. Shimony, “Forward search value iteration for pomdps.,” in *IJCAI*, pp. 2619–2624, 2007.

[18] T. Smith and R. Simmons, “Heuristic search value iteration for pomdps,” in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 520–527, AUAI Press, 2004.

[19] M. T. Spaan and N. Vlassis, “Perseus: Randomized point-based value iteration for pomdps,” *Journal of artificial intelligence research*, vol. 24, pp. 195–220, 2005.

[20] T. Smith and R. Simmons, “Point-based pomdp algorithms: improved analysis and implementation,” in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 542–549, AUAI Press, 2005.

[21] A. R. Cassandra and L. P. Kaelbling, “Learning policies for partially observable environments: Scaling up,” in *Machine Learning Proceedings 1995: Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, July 9-12 1995*, p. 362, Morgan Kaufmann, 2016.

[22] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa, “Online planning algorithms for pomdps,” *Journal of Artificial Intelligence Research*, vol. 32, pp. 663–704, 2008.

[23] G. Shani, J. Pineau, and R. Kaplow, “A survey of point-based pomdp solvers,” *Autonomous Agents and Multi-Agent Systems*, pp. 1–51, 2013.

[24] A. Somani, N. Ye, D. Hsu, and W. S. Lee, “Despot: Online pomdp planning with regularization,” in *Advances in neural information processing systems*, pp. 1772–1780, 2013.

[25] D. Silver and J. Veness, “Monte-carlo planning in large pomdps,” in *Advances in neural information processing systems*, pp. 2164–2172, 2010.

[26] 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)*, pp. 2290–2297, IEEE, 2015.

[27] D. Mayne, “Robust and stochastic mpc: Are we going in the right direction?,” *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 1–8, 2015.

[28] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[29] J. N. Tsitsiklis, “Computational complexity in markov decision theory,” *HERMIS-An International Journal of Computer Mathematics and its Applications*, vol. 9, pp. 45–54, 2007.

[30] Y. Le Tallec, *Robust, risk-sensitive, and data-driven control of Markov decision processes.* PhD thesis, Massachusetts Institute of Technology, 2007.

[31] R. E. Kopp, “Pontryagin maximum principle,” *Mathematics in Science and Engineering*, vol. 5, pp. 255–279, 1962.

[32] D. H. Jacobson and D. Q. Mayne, “Differential dynamic programming,” 1970.

[33] E. Theodorou, Y. Tassa, and E. Todorov, “Stochastic differential dynamic programming,” in *American Control Conference (ACC), 2010*, pp. 1125–1132, IEEE, 2010.

[34] E. Todorov and W. Li, “A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems,” in *American Control Conference, 2005. Proceedings of the 2005*, pp. 300–306, IEEE, 2005.

[35] 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.

[36] J. van den Berg, “Extended lqr: locally-optimal feedback control for systems with non-linear dynamics and non-quadratic cost,” in *Robotics Research*, pp. 39–56, Springer, 2016.

[37] P. Benigno and M. Woodford, “Linear-quadratic approximation of optimal policy problems,” *Journal of Economic Theory*, vol. 147, no. 1, pp. 1–42, 2012.

[38] F. A. Oliehoek and C. Amato, *A concise introduction to decentralized POMDPs*. Springer, 2016.

[39] D. V. Pynadath and M. Tambe, “The communicative multiagent team decision problem: Analyzing teamwork theories and models,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 389–423, 2002.

[40] S. Seuken and S. Zilberstein, “Formal models and algorithms for decentralized decision making under uncertainty,” *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 2, pp. 190–250, 2008.

[41] C. Boutilier, “Planning, learning and coordination in multiagent decision processes,” in *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pp. 195–210, Morgan Kaufmann Publishers Inc., 1996.

[42] C. Amato, G. Chowdhary, A. Geramifard, N. K. Ure, and M. J. Kochenderfer, “Decentralized control of partially observable markov decision processes,” in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 2398–2405, IEEE, 2013.

[43] F. A. Oliehoek, “Decentralized pomdps,” *Reinforcement Learning*, pp. 471–503, 2012.

[44] C. V. Goldman and S. Zilberstein, "Decentralized control of cooperative systems: Categorization and complexity analysis," 2004.

[45] D. S. Bernstein, C. Amato, E. A. Hansen, and S. Zilberstein, "Policy iteration for decentralized control of markov decision processes," *Journal of Artificial Intelligence Research*, vol. 34, no. 1, p. 89, 2009.

[46] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of markov decision processes," *Mathematics of operations research*, vol. 27, no. 4, pp. 819–840, 2002.

[47] M. Mundhenk, J. Goldsmith, and E. Allender, "The complexity of policy evaluation for finite-horizon partially-observable markov decision processes," *Mathematical Foundations of Computer Science 1997*, pp. 129–138, 1997.

[48] C. H. Papadimitriou and J. Tsitsiklis, "Intractable problems in control theory," *SIAM journal on control and optimization*, vol. 24, no. 4, pp. 639–654, 1986.

[49] C. Papadimitriou and J. N. Tsitsiklis, "The complexity of markov decision processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, 1987.

[50] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella, "Taming decentralized pomdps: Towards efficient policy computation for multiagent settings," in *IJCAI*, vol. 3, pp. 705–711, 2003.

[51] C. Amato, F. A. Oliehoek, *et al.*, "Scalable planning and learning for multiagent pomdps.," in *AAAI*, pp. 1995–2002, 2015.

[52] J. V. Messias, M. T. Spaan, and P. U. Lima, “Multiagent pomdps with asynchronous execution,” in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 1273–1274, International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[53] J. V. Messias, M. Spaan, and P. U. Lima, “Efficient offline communication policies for factored multiagent pomdps,” in *Advances in Neural Information Processing Systems*, pp. 1917–1925, 2011.

[54] C. Boutilier and D. Poole, “Computing optimal policies for partially observable decision processes using compact representations,” in *Proceedings of the National Conference on Artificial Intelligence*, pp. 1168–1175, 1996.

[55] E. A. Hansen and Z. Feng, “Dynamic programming for pomdps using a factored state representation.,” in *AIPS*, pp. 130–139, 2000.

[56] S. Omidshafiei, A. a. Agha-mohammadi, C. Amato, and J. P. How, “Decentralized control of partially observable markov decision processes using belief space macro-actions,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5962–5969, May 2015.

[57] 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.

[58] S. Patil, G. Kahn, M. Laskey, J. Schulman, K. Goldberg, and P. Abbeel, “Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation,” in *Algorithmic Foundations of Robotics XI*, pp. 515–533, Springer, 2015.

[59] 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.

[60] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using differential dynamic programming in belief space," in *Robotics Research*, pp. 473–490, Springer, 2017.

[61] R. Platt, "Convex receding horizon control in non-gaussian belief space," in *Algorithmic Foundations of Robotics X*, pp. 443–458, Springer, 2013.

[62] M. Rafieisakhaei, A. Tamjidi, S. Chakravorty, and P. Kumar, "Feedback motion planning under non-gaussian uncertainty and non-convex state constraints," *International Conference on Robotics and Automation (ICRA)*, 2016.

[63] H. J. Kushner, "Near optimal control in the presence of small stochastic perturbations," *Journal of Basic Engineering*, vol. 87, no. 1, pp. 103–108, 1965.

[64] W. H. Fleming, "Stochastic control for small noise intensities," *SIAM Journal on Control*, vol. 9, no. 3, pp. 473–517, 1971.

[65] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, "A near-optimal separation principle for nonlinear stochastic systems arising in robotic path planning and control," *arXiv preprint arXiv:1705.08566*, 2017.

[66] C.-P. Tsai, "Perturbed stochastic linear regulator problems," *SIAM Journal on Control and Optimization*, vol. 16, no. 3, pp. 396–410, 1978.

[67] C. J. Holland, "Small noise open loop control," *SIAM Journal on Control*, vol. 12, no. 3, pp. 380–388, 1974.

[68] H. Cruz-Suárez and R. Ilhuicatzi-Roldán, "Stochastic optimal control for small noise intensities: the discrete-time case," *WSEAS Transactions on Mathematics*, vol. 9, no. 2, pp. 120–129, 2010.

[69] Y. Kifer, "A discrete-time version of the wentzell-friedlin theory," *The Annals of Probability*, pp. 1676–1692, 1990.

[70] D. Grass, T. Kiseleva, and F. Wagener, "Small-noise asymptotics of hamilton–jacobi–bellman equations and bifurcations of stochastic optimal control problems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 22, no. 1, pp. 38–54, 2015.

[71] W. Fleming and P. Souganidis, "Asymptotic series for solutions to the dynamic programming equation for diffusions with small noise," in *Decision and Control, 1985 24th IEEE Conference on*, vol. 24, pp. 1343–1344, IEEE, 1985.

[72] R. S. Liptser, W. Runggaldier, and M. Taksar, "Deterministic approximation for stochastic control problems," *SIAM journal on control and optimization*, vol. 34, no. 1, pp. 161–178, 1996.

[73] W. Wonham, "On the separation theorem of stochastic control," *SIAM Journal on Control*, vol. 6, no. 2, pp. 312–326, 1968.

[74] H. J. Kushner, *Introduction to stochastic control.* Holt, Rinehart and Winston New York, 1971.

[75] H. S. Witsenhausen, “Separation of estimation and control for discrete time systems,” *Proceedings of the IEEE*, vol. 59, no. 11, pp. 1557–1566, 1971.

[76] A. N. Atassi and H. K. Khalil, “A separation principle for the stabilization of a class of nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 44, no. 9, pp. 1672–1687, 1999.

[77] J. E. Potter, *A guidance-navigation separation theorem.* Massachusetts Institute of Technology, Experimental Astronomy Laboratory, 1964.

[78] A. E. Lim, J. B. Moore, and L. Faybusovich, “Separation theorem for linearly constrained lqg optimal control,” *Systems & control letters*, vol. 28, no. 4, pp. 227–235, 1996.

[79] R. Curry, “Separation theorem for nonlinear measurements,” *IEEE Transactions on Automatic Control*, vol. 14, no. 5, pp. 561–564, 1969.

[80] M. Arcak, “A global separation theorem for a new class of nonlinear observers,” in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 1, pp. 676–681, IEEE, 2002.

[81] A. E. Lim and J. B. Moore, “A quasi-separation theorem for lqg optimal control with iq constraints,” *Systems & control letters*, vol. 32, no. 1, pp. 21–33, 1997.

[82] W. H. Fleming and É. Pardoux, “Optimal control for partially observed diffusions,” *SIAM Journal on Control and Optimization*, vol. 20, no. 2, pp. 261–285, 1982.

[83] J.-M. Bismut, “Partially observed diffusions and their control,” *SIAM Journal on Control and Optimization*, vol. 20, no. 2, pp. 302–309, 1982.

[84] A. Bensoussan, *Stochastic control of partially observable systems.* Cambridge University Press, 2004.

[85] C. Charalmbous and F. Rezaei, "Optimization of stochastic uncertain systems: Large deviations and robustness for partially observable diffusions," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 4, pp. 3152–3157, IEEE, 2004.

[86] W. H. Fleming and E. Pardoux, "Piecewise monotone filtering with small observation noise," *SIAM journal on control and optimization*, vol. 27, no. 5, pp. 1156–1181, 1989.

[87] W. H. Fleming and Q. Zhang, "Piecewise monotone filtering with small observation noise: Numerical simulations," in *Applied Stochastic Analysis*, pp. 108–120, Springer, 1992.

[88] S. Peng, "A general stochastic maximum principle for optimal control problems," *SIAM Journal on control and optimization*, vol. 28, no. 4, pp. 966–979, 1990.

[89] H. Kushner, "Necessary conditions for continuous parameter stochastic optimization problems," *SIAM Journal on Control*, vol. 10, no. 3, pp. 550–565, 1972.

[90] F. Chighoub, B. Djehiche, and B. Mezerdi, "The stochastic maximum principle in optimal control of degenerate diffusions with non-smooth coefficients," *Random Operators and Stochastic Equations*, vol. 17, no. 1, pp. 37–54, 2009.

[91] U. G. Haussmann, *A stochastic maximum principle for optimal control of diffusions.* John Wiley & Sons, Inc., 1986.

[92] U. Haussmann, “Some examples of optimal stochastic controls or: the stochastic maximum principle at work,” *SIAM Review*, vol. 23, no. 3, pp. 292–307, 1981.

[93] U. Haussmann, “The maximum principle for optimal control of diffusions with partial information,” *SIAM journal on control and optimization*, vol. 25, no. 2, pp. 341–361, 1987.

[94] C. D. Charalambous and J. L. Hibey, “Necessary conditions of optimization for partially observed controlled diffusions,” *SIAM Journal on Control and Optimization*, vol. 37, no. 6, pp. 1676–1700, 1999.

[95] 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.

[96] J. Pineau, G. Gordon, and S. Thrun, “Point-based value iteration: An anytime algorithm for POMDPs,” in *International Joint Conference on Artificial Intelligence*, pp. 1025–1032, 2003.

[97] T. Smith and R. Simmons, “Point-based pomdp algorithms: Improved analysis and implementation,” in *Proceedings of Uncertainty in Artificial Intelligence*, 2005.

[98] M. Spaan and N. Vlassis, “Perseus: Randomized point-based vallue iteration for pomdps,” *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, 2005.

[99] H. Kurniawati, D. Hsu, and W. Lee, “SARSOP: Efficient point-based pomdp planning by approximating optimally reachable belief spaces,” in *Proceedings of Robotics: Science and Systems*, 2008.

[100] J. M. Porta, N. Vlassis, M. T. J. Spaan, and P. Poupart, “Point-based value iteration for continuous POMDPs,” *Journal of Machine Learning Research*, vol. 7, pp. 2329–2367, Nov. 2006.

[101] H. Bai, D. Hsu, W. S. Lee, and V. A. Ngo, “Monte carlo value iteration for continuous-state pomdps.,” in *WAFR*, vol. 68 of *Springer Tracts in Advanced Robotics*, pp. 175–191, Springer, 2010.

[102] S. C. W. Ong, S. W. Png, D. Hsu, and W. S. Lee, “Planning under uncertainty for robotic tasks with mixed observability.,” *International Journal of Robotics Research*, vol. 29, no. 8, pp. 1053–1068, 2010.

[103] H. Bai, D. Hsu, W. Lee, and V. Ngo, “Monte carlo value iteration for continuous-state pomdps,” *Algorithmic foundations of robotics IX*, pp. 175–191.

[104] Z. Lim, W. Lee, and D. Hsu, “Monte carlo value iteration with macro-actions,” *Advances in Neural Information Processing Systems*, pp. 1287–1295.

[105] R. Zhou and E. A. Hansent, “An improved grid-based approximation algorithm for pomdps,” *IJCAI*, 2001.

[106] O. Madani, S. Hanks, and A. Condon, “On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems,” in *Proceedings of the Sixteen Conference on Artificial Intelligence (AAAI)*, pp. 541–548, 1999.

[107] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, “Efficient planning in non-gaussian belief spaces and its application to robot grasping,” *IntâĂŹl Symposium on Robotics Research*, 2011.

[108] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, “Belief space planning assuming maximum likelihood observations,” in *Robotics: Science and Systems (RSS)*, 2010.

[109] R. Platt, “Convex receding horizon control in non-gaussian belief space,” in *International Workshop on Algorithmic Foundations of Robotics*, 2012.

[110] A. Agha-mohammadi, S. Chakravorty, and N. Amato, “FIRM: Feedback controller-based Information-state RoadMap -a framework for motion planning under uncertainty-,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[111] S. Prentice and N. Roy., “The belief roadmap: Efficient planning in linear pomdps by factoring the covariance,” 2008.

[112] A. Bry and N. Roy, “Rapidly-exploring random belief trees for motion planning under uncertainty,” in *IEEE IntâĂŹl Conf. on Robotics and Automation*, 2011.

[113] A. Censi, D. Calisi, A. D. Luca, and G. Oriolo, “A Bayesian framework for optimal motion planning with uncertainty,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (Pasadena, CA), May 2008.

[114] R. Platt, L. Kaelbling, T. Lozano-Perez, , and R. Tedrake, “Efficient planning in non-gaussian belief spaces and its application to robot grasping,” in *Proc. of International Symposium of Robotics Research, (ISRR)*, 2011.

[115] S. Prentice and N. Roy, “The belief roadmap: Efficient planning in belief

space by factoring the covariance," *International Journal of Robotics Research*, vol. 28, October 2009.

[116] V. Huynh and N. Roy, "icLQG: combining local and global optimization for control in information space," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[117] L. Kavraki, P. Švestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[118] L. Kavraki, M. Kolountzakis, and J. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 166–171, February 1998.

[119] A. Ladd and L. Kavraki, "Measure theoretic analysis of probabilistic path planning," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 229–242, April 2004.

[120] L. Kavraki, J. Latombe, R. Motwani, and P. Raghavan, "Randomized query processing in robot motion planning," in *Proc. ACM Symp. Theory of Computing*, pp. 353–362, 1995.

[121] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of robot motion: theory, algorithms, and implementations.* MIT Press, 2005.

[122] N. D. Toit and J. W. Burdick, "Robotic motion planning in dynamic, cluttered, uncertain environments," in *ICRA*, May 2010.

[123] A. Bry and N. Roy, “Rapidly-exploring random belief trees for motion planning under uncertainty.,” in *ICRA*, pp. 723–730, 2011.

[124] S. Lavalle and J. Kuffner, “Randomized kinodynamic planning,” *International Journal of Robotics Research*, vol. 20, no. 378-400, 2001.

[125] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *International Journal of Robotics Research*, vol. 30, pp. 846–894, June 2011.

[126] M. P. Vitus and C. J. Tomlin, “Closed-loop belief space planning for linear, Gaussian systems.,” in *ICRA*, pp. 2152–2159, 2011.

[127] W. Li and E. Todorov, “Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system,” *International Journal of Control*, vol. 80, no. 9, pp. 1439–1453, 2007.

[128] J. Van den Berg, S. Patil, and R. Alterovitz, “Motion planning under uncertainty using differential dynamic programming in belief space,” in *Proc. of International Symposium of Robotics Research, (ISRR)*, 2011.

[129] A.-A. Agha-Mohammadi, *Feedback-based Information Roadmap (FIRM): Graph-based Estimation and Control of Robotic Systems Under Uncertainty.* PhD thesis, Texas A&M University, 2014.

[130] C. E. Garcia, D. M. Prett, and M. Morari, “Model predictive control: theory and practiceâĂŤa survey,” *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[131] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[132] D. Q. Mayne and H. Michalska, “Receding horizon control of nonlinear systems,” *Automatic Control, IEEE Transactions on*, vol. 35, no. 7, pp. 814–824, 1990.

[133] H. Michalska and D. Q. Mayne, “Robust receding horizon control of constrained nonlinear systems,” *Automatic Control, IEEE Transactions on*, vol. 38, no. 11, pp. 1623–1633, 1993.

[134] S. a. Keerthi and E. G. Gilbert, “Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations,” *Journal of optimization theory and applications*, vol. 57, no. 2, pp. 265–293, 1988.

[135] E. S. Meadows, M. A. Henson, J. W. Eaton, and J. B. Rawlings, “Receding horizon control and discontinuous state feedback stabilization,” *International Journal of Control*, vol. 62, no. 5, pp. 1217–1229, 1995.

[136] S. J. Qin and T. A. Badgwell, “An overview of industrial model predictive control technology,” in *AIChE Symposium Series*, vol. 93, pp. 232–256, New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997.

[137] G. Zames, “Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses,” *Automatic Control, IEEE Transactions on*, vol. 26, no. 2, pp. 301–320, 1981.

[138] J. C. Doyle, K. Glover, P. P. Khargonekar, B. Francis, *et al.*, “State-space solutions to standard h 2 and hâĹđ control problems,” *Automatic Control, IEEE Transactions on*, vol. 34, no. 8, pp. 831–847, 1989.

[139] G. Zames, B. Francis, *et al.*, “Feedback, minimax sensitivity, and optimal robustness,” *Automatic Control, IEEE Transactions on*, vol. 28, no. 5, pp. 585–601, 1983.

[140] D. Mayne, “Optimization in model predictive control,” in *Methods of Model Based Process Control*, pp. 367–396, Springer, 1995.

[141] E. Polak, *Optimization: algorithms and consistent approximations*, vol. 124. Springer Science & Business Media, 2012.

[142] R. Bellman, “A markovian decision process,” tech. rep., DTIC Document, 1957.

[143] E. B. Lee and L. Markus, “Foundations of optimal control theory,” tech. rep., DTIC Document, 1967.

[144] K. J. Åström, “Theory and applications of adaptive controlâĂŤa survey,” *Automatica*, vol. 19, no. 5, pp. 471–486, 1983.

[145] D. Mayne and J. Rawlings, “Model predictive control: theory and design,” *Madison, WI: Nob Hill Publishing, LCC*, 2009.

[146] S. Yu, M. Reble, H. Chen, and F. Allgöwer, “Inherent robustness properties of quasi-infinite horizon mpc,” in *18th IFAC World Congress, Milano*, 2011.

[147] D. L. Marruedo, T. Alamo, and E. Camacho, “Input-to-state stable mpc for constrained discrete-time nonlinear systems with bounded additive uncertainties,” in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 4, pp. 4619–4624, IEEE, 2002.

[148] Z.-P. Jiang and Y. Wang, “Input-to-state stability for discrete-time nonlinear systems,” *Automatica*, vol. 37, no. 6, pp. 857–869, 2001.

[149] D. Limon, T. Alamo, D. Raimondo, D. M. de la Pena, J. Bravo, A. Ferramosca, and E. Camacho, “Input-to-state stability: a unifying framework for robust model predictive control,” in *Nonlinear model predictive control*, pp. 1–26, Springer, 2009.

[150] M. Lazar, W. Heemels, and A. Teel, “Further input-to-state stability subtleties for discrete-time systems,” *Automatic Control, IEEE Transactions on*, vol. 58, pp. 1609–1613, June 2013.

[151] E. D. Sontag and Y. Wang, “On characterizations of the input-to-state stability property,” *Systems & Control Letters*, vol. 24, no. 5, pp. 351–359, 1995.

[152] G. Pin, D. M. Raimondo, L. Magni, and T. Parisini, “Robust model predictive control of nonlinear systems with bounded and state-dependent uncertainties,” *Automatic Control, IEEE Transactions on*, vol. 54, no. 7, pp. 1681–1687, 2009.

[153] D. P. Bertsekas and I. B. Rhodes, “Recursive state estimation for a set-membership description of uncertainty,” *Automatic Control, IEEE Transactions on*, vol. 16, no. 2, pp. 117–128, 1971.

[154] D. P. Bertsekas and I. B. Rhodes, “On the minimax reachability of target sets and target tubes,” *Automatica*, vol. 7, no. 2, pp. 233–247, 1971.

[155] L. Chisci, J. A. Rossiter, and G. Zappa, “Systems with persistent disturbances: predictive control with restricted constraints,” *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.

[156] D. Bernardini and A. Bemporad, “Scenario-based model predictive control of stochastic constrained linear systems,” in *Decision and Control, 2009 held*

*jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 6333–6338, IEEE, 2009.

[157] N. Kantas, J. Maciejowski, and A. Lecchini-Visintini, “Sequential monte carlo for model predictive control,” in *Nonlinear Model Predictive Control*, pp. 263–273, Springer, 2009.

[158] G. C. Calafiore and M. C. Campi, “The scenario approach to robust control design,” *Automatic Control, IEEE Transactions on*, vol. 51, no. 5, pp. 742–753, 2006.

[159] P. Kumar *et al.*, “Control: a perspective,” *Automatica*, vol. 50, no. 1, pp. 3–43, 2014.

[160] V. D. Blondel and J. N. Tsitsiklis, “A survey of computational complexity results in systems and control,” *Automatica*, vol. 36, no. 9, pp. 1249–1274, 2000.

[161] M. Rafieisakhaei, S. Chakravorty, and P. R. Kumar, “A Near-Optimal Decoupling Principle for Nonlinear Stochastic Systems Arising in Robotic Path Planning and Control,” in *56th IEEE Conference on Decision and Control (CDC)*, IEEE, 2017.

[162] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, “On the use of the observability gramian for partially observed robotic path planning problems,” in *56th IEEE Conference on Decision and Control (CDC)*, IEEE, 2017.

[163] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, “Mt-lqg: Multi-agent planning in belief space via trajectory-optimized lqg,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5583–5590, IEEE, 2017.

[164] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, “T-lqg: Closed-loop belief space planning via trajectory-optimized lqg,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 649–656, IEEE, 2017.

[165] 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.

[166] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, “Belief space planning simplified: Trajectory-optimized lqg (t-lqg),” *arXiv preprint arXiv:1608.03013*, 2016.

[167] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, “Near-optimal belief space planning via t-lqg,” *arXiv preprint arXiv:1705.09415*, 2017.

[168] D. Yu, M. Rafieisakhaei, and S. Chakravorty, “Stochastic feedback control of systems with unknown nonlinear dynamics,” *arXiv preprint arXiv:1705.09761*, 2017.

[169] M. Rafieisakhaei, A. Tamjidi, and S. Chakravorty, “On-line mpc-based stochastic planning in the non-gaussian belief space with non-convex constraints,” 2015.

[170] D. Yu, M. Rafieisakhaei, and S. Chakravorty, “A separation-based design to data-driven control for large-scale partially observed systems,” 2017.

[171] A. D. Wentzell, *Limit theorems on large deviations for Markov stochastic processes*, vol. 38. Springer Science & Business Media, 2012.

[172] A. Dembo and O. Zeitouni, *Large deviations techniques and applications*, vol. 38. Springer Science & Business Media, 2009.

[173] H. Cruz-Suárez and R. Ilhuicatzi-Roldán, “Stochastic optimal control for small noise intensities: The discrete-time case,” *WSEAS Trans. Math.*, vol. 9, pp. 120–129, Feb. 2010.

[174] J. D. Perkins and R. W. H. Sargent, *Nonlinear optimal stochastic control — some approximations when the noise is small*, pp. 820–830. Berlin, Heidelberg: Springer Berlin Heidelberg, 1976.

[175] J. Perkins and R. Sargent, “Nonlinear optimal stochastic controlâĂŤsome approximations when the noise is small,” in *IFIP Technical Conference on Optimization Techniques*, pp. 820–830, Springer, 1975.

[176] C. J. Holland, “An approximation technique for small noise open-loop control problems,” *Optimal Control Applications and Methods*, vol. 2, no. 1.

[177] S. S. Varadhan and S. S. Varadhan, *Large deviations and applications*, vol. 46. SIAM, 1984.

[178] cardinal (https://math.stackexchange.com/users/7003/cardinal), “Proof of upper-tail inequality for standard normal distribution.” Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/28754 (version: 2011-03-24).

[179] S. Lavalle, *Planning algorithms.* Cambridge University Press, 2006.

[180] R. C. James, *Advanced calculus.* Wadsworth Pub. Co., 1966.

[181] 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.

[182] W. Sun, S. Patil, and R. Alterovitz, "High-frequency replanning under uncertainty using parallel sampling-based motion planning," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 104–116, 2015.

[183] D. S. Watkins, *Fundamentals of matrix computations*, vol. 64. John Wiley & Sons, 2004.

[184] J. Duchi, "Derivations for linear algebra and optimization," *Berkeley, California*, 2007.

[185] N. Moshtagh, "Minimum volume enclosing ellipsoid," *Convex Optimization*, vol. 111, p. 112, 2005.

[186] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[187] S. Bhattacharya, V. Kumar, and M. Likhachev, "Search-based path planning with homotopy class constraints," in *Third Annual Symposium on Combinatorial Search*, 2010.

[188] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Autonomous Robots*, vol. 33, no. 3, pp. 273–290, 2012.

[189] 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.

[190] S. Bubeck, "Theory of convex optimization for machine learning," *arXiv preprint arXiv:1405.4980*, 2014.

[191] A. Nemirovsky, “Problem complexity and method efficiency in optimization.,”

[192] zakharov, “zakharov youbot model,” 2011.

[193] 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.

[194] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization.* Dover Publications, Inc. NY, 1998.

[195] M. Rafieisakhaei, S. Chakravorty, and P. R. Kumar, “Belief space planning simplified: Trajectory-optimized lqg (t-lqg),” 2016 (Submitted).

[196] M. Rafieisakhaei, S. Chakravorty, and P. Kumar, “Belief space planning simplified: Trajectory-optimized lqg (t-lqg),” *arXiv preprint arXiv:1608.03013*, 2016.

[197] A. Agha-mohammadi, S. Agarwal, S. Chakravorty, and N. M. Amato, “Simultaneous localization and planning for physical mobile robots via enabling dynamic replanning in belief space,” *CoRR*, vol. abs/1510.07380, 2015.

[198] S. Omidshafiei, A.-a. Agha-mohammadi, C. Amato, S.-Y. Liu, J. P. How, and J. Vian, “Graph-based cross entropy method for solving multi-robot decentralized pomdps,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5395–5402, IEEE, 2016.

[199] Youbot-store.com, “Youbot 3d model - youbot wiki,” 2016.

[200] P. S. Maybeck, *Stochastic models, estimation, and control*, vol. 3, pp. 45–48, 238–241. Academic press, 1982.

[201] K. Yasuda and R. E. Skelton, “Assigning controllability and observability gramians in feedback control,” *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 5, pp. 878–885, 1991.

[202] U. Vaidya, “Observability gramian for nonlinear systems,” in *Decision and Control, 2007 46th IEEE Conference on*, pp. 3357–3362, IEEE, 2007.

[203] B. Southall, B. F. Buxton, and J. A. Marchant, “Controllability and observability: Tools for kalman filter design.,” in *BMVC*, pp. 1–10, 1998.

[204] A. J. Krener and K. Ide, “Measures of unobservability,” in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 6401–6406, IEEE, 2009.

[205] D. Georges, “Energy minimization and observability maximization in multi-hop wireless sensor networks,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 13918–13923, 2011.

[206] B. T. Hinson, *Observability-based guidance and sensor placement.* PhD thesis, University of Washington, 2014.

[207] B. T. Hinson, M. K. Binder, and K. A. Morgansen, “Path planning to optimize observability in a planar uniform flow field,” in *American Control Conference (ACC), 2013*, pp. 1392–1399, IEEE, 2013.

[208] J. D. Quenzer and K. A. Morgansen, “Observability based control in range-only underwater vehicle localization,” in *American Control Conference (ACC), 2014*, pp. 4702–4707, IEEE, 2014.

[209] M. Travers and H. Choset, “Use of the nonlinear observability rank condition for improved parametric estimation,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 1029–1035, IEEE, 2015.

[210] L. DeVries and D. A. Paley, “Wake sensing and estimation for control of autonomous aircraft in formation flight,” *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 1, pp. 32–41, 2015.

[211] L. DeVries, S. J. Majumdar, and D. A. Paley, “Observability-based optimization of coordinated sampling trajectories for recursive estimation of a strong, spatially varying flowfield,” *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 527–544, 2013.

[212] A. K. Singh and J. Hahn, “Determining optimal sensor locations for state and parameter estimation for stable nonlinear systems,” *Industrial & engineering chemistry research*, vol. 44, no. 15, pp. 5645–5659, 2005.

[213] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, “Non-gaussian belief space planning: Correctness and complexity,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 4711–4717, IEEE, 2012.

[214] P. Tichavsky, C. H. Muravchik, and A. Nehorai, “Posterior cramér-rao bounds for discrete-time nonlinear filtering,” *IEEE Transactions on signal processing*, vol. 46, no. 5, pp. 1386–1396, 1998.

[215] N. Thacker and A. Lacey, “Tutorial: The likelihood interpretation of the kalman filter,” *TINA Memos: Advanced Applied Statistics*, vol. 2, no. 1, pp. 1–11, 1996.

[216] M. Lei, C. Baehr, and P. Del Moral, “Fisher information matrix-based nonlinear system conversion for state estimation,” in *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, pp. 837–841, IEEE, 2010.

[217] G. Casella and R. L. Berger, *Statistical inference*, vol. 2. Duxbury Pacific Grove, CA, 2002.

[218] M. Rafieisakhaei, S. Chakravorty, and P. R. Kumar, “T-LQG: Closed-Loop Belief Space Planning via Trajectory-Optimized LQG,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 649–656, IEEE, 2017.

[219] B. Barazandeh, K. Bastani, M. Rafieisakhaei, S. Kim, Z. Kong, and M. A. Nussbaum, “Robust sparse representation-based classification using online sensor data for monitoring manual material handling tasks,” *IEEE Transactions on Automation Science and Engineering*, 2017.

[220] M. Boloursaz, R. Kazemi, B. Barazandeh, and F. Behnia, “Bounds on compressed voice channel capacity,” in *Communication and Information Theory (IWCIT), 2014 Iran Workshop on*, pp. 1–6, IEEE, 2014.

[221] D. Crisan and A. Doucet, “A survey of convergence results on particle filtering methods for practitioners,” *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 50, no. 3, 2002.

[222] A. Doucet, J. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice.* New York: Springer, 2001.

[223] S. Yakowitz, “Algorithms and computational techniques in differential dynamic programming,” *Control and Dynamical Systems: Advances in Theory and Applications*, vol. 31, pp. 75–91, 2012.