Experiments with Tractable Feedback in Robotic Planning under Uncertainty: Insights over a wide range of noise regimes

Mohamed Naveed Gul Mohamed, Suman Chakravorty, and Dylan A. Shell

Texas A&M University, College Station TX 77843, USA,
mohdnaveed96@gmail.com, schakrav@tamu.edu, dshell@tamu.edu.

Abstract. We consider the problem of robotic planning under uncertainty. This problem may be posed as a stochastic optimal control problem, complete solution to which is fundamentally intractable owing to the infamous curse of dimensionality. We report the results of an extensive simulation study in which we have compared two methods, both of which aim to salvage tractability by using alternative, albeit inexact, means for treating feedback. The first is a recently proposed method based on a near-optimal "decoupling principle" for tractable feedback design, wherein a nominal open-loop problem is solved, followed by a linear feedback design around the open-loop. The second is Model Predictive Control (MPC), a widely-employed method that uses repeated re-computation of the nominal open-loop problem during execution to correct for noise, though when interpreted as feedback, this can only said to be an implicit form. We examine a much wider range of noise levels than have been previously reported and empirical evidence suggests that the decoupling method allows for tractable planning over a wide range of uncertainty conditions without unduly sacrificing performance.

Keywords: Empirical study, Optimization, Optimal Control

1 Introduction

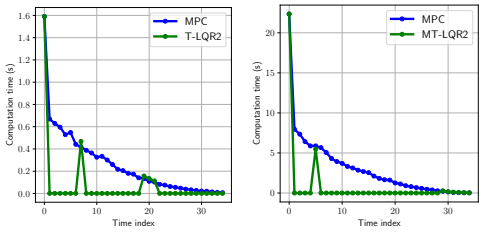
Planning under uncertainty is a central problem in robotics. The space of current methods includes several contenders, each with different simplifying assumptions, approximations, and domains of applicability. This is a natural consequence of the fact that the challenge of dealing with the continuous state, control and observation space problems, for non-linear systems and across long-time horizons with significant noise, and potentially multiple agents, is fundamentally intractable.

Model Predictive Control is one popular means for tackling optimal control problems [13, 19]. The MPC approach solves a finite horizon "deterministic" optimal control problem at every time step given the current state of the process, performs only the first control action and then repeats the planning process at the next time step. In terms of computation, this can be a costly endeavor and,

when a stochastic control problem is well approximated by the deterministic problem (when the noise is meager), much of this computation is superfluous.

In this paper we consider the generalization of a recently proposed method [25] that uses a local feedback to control noise induced deviations from the deterministic (that we term the *nominal*) trajectory. When the deviation is too large for the feedback to manage, replanning is triggered and it computes a fresh nominal. Otherwise, the feedback tames the perturbations during execution and no computation is expended in replanning. Put another way, the method decouples feedback and planning/nominal control but will fall back to replanning when perturbations are excessive. Thus, by considering every deviation to necessitate replanning, this approach will essentially reduce to MPC itself.

We present an empirical investigation of this decoupling approach, exploring dimensions that are important in characterizing its performance—key among these being the triggering of replanning. Hence, the primary focus of the study is on understanding the performance across a wide range of noise conditions with comparison to the "gold standard" of MPC. Figure 1 gives an overall summary of the paper's findings: the areas under the respective curves give the total computational resources consumed—the savings by the decoupling method over MPC are seen to be considerable.



(a) A single agent. (b) Three agents.

Fig. 1: Computation time expended by MPC (in blue) and the decoupling algorithms described herein (in green), at each time step for a sample experiment involving navigation. Both cases result in nearly identical motions by the robot.The peaks in T-LQR2 and MT-LQR2 happen only when replanning takes place. Computational effort decreases for both methods because the horizon diminishes as the agent(s) reach their goals.(To relate to subsequent figures: noise parameter $\epsilon = 0.4$ and the replan threshold = 2% of cost deviation.)

1.1 Related Work

Robotic planning problems under uncertainty can be posed as a stochastic optimal control problem that requires the solution of an associated Dynamic Programming (DP) problem, however, as the state dimension increases, the computational complexity goes up exponentially [4], Bellman's infamous "curse of dimensionality". There has been recent success using sophisticated (Deep) Reinforcement Learning (RL) paradigm to solve DP problems, where deep neural networks are used as the function approximators [2, 10, 11, 22, 23], however, the training time required for these approaches is still prohibitive to permit real-time robotic planning that is considered here.

In the case of continuous state, control and observation space problems, the Model Predictive Control [13, 19] approach has been used with a lot of success

in the control system and robotics community. For deterministic systems, the process results in solving the original DP problem in a recursive online fashion. However, stochastic control problems, and the control of uncertain systems in general, is still an unresolved problem in MPC. As succinctly noted in [13], the problem arises due to the fact that in stochastic control problems, the MPC optimization at every time step cannot be over deterministic control sequences, but rather has to be over feedback policies, which is, in general, difficult to accomplish since a tractable parametrization of such policies to perform the optimization over, is, in general, unavailable. Thus, the tube-based MPC approach, and its stochastic counterparts, typically consider linear systems [7, 14, 20] for which a linear parametrization of the feedback policy suffices but the methods become intractable when dealing with nonlinear systems. In recent work, we have introduced a "decoupling principle" that allows us to tractably solve such stochastic optimal control problems in a near optimal fashion, with applications to highly efficient RL and MPC implementations [17,25]. However, this prior work required a small noise assumption. In this work, we relax this small noise assumption to show, via extensive empirical evaluation, that even when the noise is not small, a *replan-when-necessary* modification of the decoupled planning approach, akin to event-triggered MPC [9, 12], suffices to keep the planning computationally efficient while retaining performance comparable to MPC. We note that event-triggered MPC inherits the same issues mentioned above with respect to the stochastic control problem, and consequently, the techniques are only tractable for linear systems. *Lest it seem that we are being unduly critical of MPC, that is definitely not our intention: we believe that MPC type replanning is unavoidable in uncertain systems, instead we additionally believe that such replanning can be substantially reduced utilizing decoupling while tractably and rigorously extending MPC to stochastic systems, i.e., the decoupled approach is not competition, but rather complimentary, to MPC. Please also see the deeper historical context to this discussion at the end of Section 3.1, after we have presented the basic near-optimality result.*

The problem of multiple agents further and severely compounds the planning problem since now we are also faced with the issue of a control space that grows exponentially with the number of agents in the system. Moreover, since the individual agents never have full information regarding the system state, the observations are partial. Furthermore, the decision making has to be done in a distributed fashion which places additional constraints on the networking and communication resources. In a multi-agent setting, the stochastic optimal problem can be formulated in the space of joint policies. Some variations of this problem have been successfully characterized and tackled based on the level of observability, in/dependence of the dynamics, cost functions and communications [16, 18, 21]. This has resulted in a variety of solutions from fully-centralized [5] to fully-decentralized approaches with many different subclasses [3, 15].

The major concerns of the multi-agent problem are tractability of the solution and the level of communication required during the execution of the policies.

In this paper, we also consider a generalization of the decoupling principle to a multi-agent, fully observed setting. We show that this leads to a spatial decoupling between agents in that they do not need to communicate for long periods of time during execution. Albeit, we do not consider the problem of when and how to replan in this paper, assuming that there exists a (yet to be determined) distributed mechanism that can achieve this, we nonetheless show that there is a highly significant increase in planning efficiency over a wide range of noise levels.

2 Problem Formulation

The problem of robot planning and control under noise can be formulated as a stochastic optimal control problem in the space of feedback policies. We assume here that the map of the environment is known and state of the robot is fully observed. Uncertainty in the problem lies in the system's actions.

2.1 System Model:

For a dynamic system, we denote the state and control vectors by $\mathbf{x}_t \in \mathbb{X} \subset \mathbb{R}^{n_x}$ and $\mathbf{u}_t \in \mathbb{U} \subset \mathbb{R}^{n_u}$ respectively at time t. The motion model $f : \mathbb{X} \times \mathbb{U} \times \mathbb{R}^{n_u} \to \mathbb{X}$ is given by the equation

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

where $\{\mathbf{w}_t\}$ are zero mean independent, identically distributed (i.i.d) random sequences with variance $\mathbf{\Sigma}_{\mathbf{w}_t}$, and ϵ is a small parameter modulating the noise input to the system.

2.2 Stochastic optimal control problem:

The stochastic optimal control problem for a dynamic system with initial state $\mathbf{x}_0$ is defined as:

$$J_{\pi^*}(\mathbf{x}_0) = \min_{\pi}\ \mathbb{E}\left[\sum_{t=0}^{T-1} c(\mathbf{x}_t, \pi_t(\mathbf{x}_t)) + c_T(\mathbf{x}_T)\right], \tag{2}$$

$$s.t.\ \mathbf{x}_{t+1} = f(\mathbf{x}_t, \pi_t(\mathbf{x}_t), \epsilon \mathbf{w}_t), \tag{3}$$

where:

- the optimization is over feedback policies $\pi := \{\pi_0, \pi_1, \ldots, \pi_{T-1}\}$ and $\pi_t(\cdot)$: $\mathbb{X} \to \mathbb{U}$ specifies an action given the state, $\mathbf{u}_t = \pi_t(\mathbf{x}_t)$;
- $J_{\pi^*}(\cdot) : \mathbb{X} \to \mathbb{R}$ is the cost function on executing the optimal policy π^*;
- $c_t(\cdot, \cdot) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ is the one-step cost function;
- $c_T(\cdot) : \mathbb{X} \to \mathbb{R}$ is the terminal cost function;
- T is the horizon of the problem;
- the expectation is taken over the random variable $\mathbf{w}_t$.

3 A Decoupling Principle

Now, we give a brief overview of a "decoupling principle" that allows us to substantially reduce the complexity of the stochastic planning problem given that the parameter ϵ is small enough. We only provide an outline here and the relevant details can be found in our recent work [25]. We shall also present a generalization to a class of multi-robot problems. Finally, we preview the results in the rest of the paper.

3.1 Near-Optimal Decoupling in Stochastic Optimal Control

Let $\pi_t(\mathbf{x}_t)$ denote a control policy for the stochastic planning problem above, not necessarily the optimal policy. Consider now the control actions of the policy when the noise to the system is uniformly zero, and let us denote the resulting "nominal" trajectory and controls as $\overline{\mathbf{x}}_t$ and $\overline{\mathbf{u}}_t$ respectively, i.e., $\overline{\mathbf{x}}_{t+1} = f(\overline{\mathbf{x}}_t, \overline{\mathbf{u}}_t, 0)$, where $\overline{\mathbf{u}}_t = \pi_t(\overline{\mathbf{x}}_t)$. Note that this nominal system is well defined. Further, let us assume that the closed-loop (i.e., with $\mathbf{u}_t = \pi_t(\mathbf{x}_t)$), system equations, and the feedback law are smooth enough that we can expand the feedback law about the nominal as $\pi_t(\mathbf{x}_t) = \overline{\mathbf{u}}_t + \mathbf{K}_t\delta\mathbf{x}_t + \mathbf{R}_t^\pi(\delta\mathbf{x}_t)$, where $\delta\mathbf{x}_t = \mathbf{x}_t - \overline{\mathbf{x}}_t$, i.e., the perturbation from the nominal, $\mathbf{K}_t$ is the linear gain obtained by the Taylor expansion about the nominal in terms of the perturbation $\delta\mathbf{x}_t$, and $\mathbf{R}_t^\pi(\cdot)$ represents the second and higher order terms in the expansion of the feedback law about the nominal trajectory. Further we assume that the closed-loop perturbation state can be expanded about the nominal as: $\delta\mathbf{x}_t = \mathbf{A}_t\delta\mathbf{x}_t + \mathbf{B}_t\mathbf{K}_t\delta\mathbf{x}_t + \mathbf{R}_t^f(\delta\mathbf{x}_t) + \epsilon\mathbf{B}_t\mathbf{w}_t$, where the $\mathbf{A}_t$, $\mathbf{B}_t$ are the system matrices obtained by linearizing the system state equations about the nominal state and control, while $\mathbf{R}_t^f(\cdot)$ represents the second and higher order terms in the closed-loop dynamics in terms of the state perturbation $\delta\mathbf{x}_t$. Moreover, let the nominal cost be given by $\overline{J}^\pi = \sum_{t=0}^{T}\overline{c}_t$, where $\overline{c}_t = c(\overline{\mathbf{x}}_t, \overline{\mathbf{u}}_t)$, for $t \leq T-1$, and $\overline{c}_T = c_T(\overline{\mathbf{x}}_T, \overline{\mathbf{u}}_T)$. Further, assume that the cost function is smooth enough that it permits the expansion $J^\pi = \overline{J} + \sum_t \mathbf{C}_t\delta\mathbf{x}_t + \sum_t \mathbf{R}_t^c(\delta\mathbf{x}_t)$ about the nominal trajectory, where $\mathbf{C}_t$ denotes the linear term in the perturbation expansion and $\mathbf{R}_t^c(\cdot)$ denote the second and higher order terms in the same. Finally, define the exactly linear perturbation system $\delta\mathbf{x}_{t+1}^\ell = \mathbf{A}_t\delta\mathbf{x}_t^\ell + \mathbf{B}_t\mathbf{K}_t\delta\mathbf{x}_t^\ell + \epsilon\mathbf{B}_t\mathbf{w}_t$. Further, let $\delta J_1^{\pi,\ell}$ denote the cost perturbation due to solely the linear system, i.e., $\delta J_1^{\pi,\ell} = \sum_t \mathbf{C}_t\delta\mathbf{x}_t^\ell$. Then, the decoupling result states the following [25]:

Theorem 1. *The closed-loop cost function J^π can be expanded as $J^\pi = \overline{J}^\pi + \delta J_1^{\pi,\ell} + \delta J_2^\pi$. Furthermore, $\mathbb{E}[J^\pi] = \overline{J}^\pi + O(\epsilon^2)$, and $\mathrm{Var}[J^\pi] = \mathrm{Var}[\delta J_1^{\pi,\ell}] + O(\epsilon^4)$, where $\mathrm{Var}[\delta J_1^{\pi,\ell}]$ is $O(\epsilon^2)$.*

Thus, the above result suggest that the mean value of the cost is determined almost solely by the nominal control actions while the variance of the cost is almost solely determined by the linear closed-loop system. Thus, the decoupling result says that the feedback law design can be decoupled into an open-loop and a closed-loop problem.

Open-Loop Problem: This problem solves the deterministic/ nominal optimal control problem:

$$\overline{J} = \min_{\overline{\mathbf{u}}_t} \sum_{t=0}^{T-1} c(\overline{\mathbf{x}}_t, \overline{\mathbf{u}}_t) + c_T(\overline{\mathbf{x}}_T), \tag{4}$$

subject to the nominal dynamics: $\overline{\mathbf{x}}_{t+1} = f(\overline{\mathbf{x}}_t, \overline{\mathbf{u}}_t)$.
Closed-Loop Problem: One may try to optimize the variance of the linear closed-loop system

$$\min_{\mathbf{K}_t} \mathrm{Var}[\delta J_1^{\pi,\ell}] \tag{5}$$

subject to the linear dynamics $\delta\mathbf{x}_{t+1}^{\ell} = \mathbf{A}_t\delta\mathbf{x}_t^{\ell} + \mathbf{B}_t\mathbf{K}_t\delta\mathbf{x}_t^{\ell} + \epsilon\mathbf{B}_t\mathbf{w}_t$. However, the above problem does not have a standard solution but note that we are only interested in a good variance for the cost function and not the optimal one. Thus, this may be accomplished by a surrogate LQR problem that provides a good linear variance as follows.
Surrogate LQR Problem: Here, we optimize the standard LQR cost:

$$\delta J_{\mathrm{LQR}} = \min_{\mathbf{u}_t} \mathop{\mathbb{E}}_{\mathbf{w}_t} \left[\sum_{t=0}^{T-1} \delta\mathbf{x}_t^{\mathsf{T}}\mathbf{Q}\delta\mathbf{x}_t + \delta\mathbf{u}_t^{\mathsf{T}}\mathbf{R}\delta\mathbf{u}_t + \delta\mathbf{x}_T^{\mathsf{T}}\mathbf{Q}_f\delta\mathbf{x}_T \right], \tag{6}$$

subject to the linear dynamics $\delta\mathbf{x}_{t+1}^{\ell} = \mathbf{A}_t\delta\mathbf{x}_t^{\ell} + \mathbf{B}_t\delta\mathbf{u}_t + \epsilon\mathbf{B}_t\mathbf{w}_t$. In this paper, this decoupled design shall henceforth be called the trajectory-optimized LQR (T-LQR) design.

A Historical Context. The above decoupled design might seem like a perturbation feedback design outlined in classical optimal control texts such as (Ch. 6, [6]) and we are certainly not claiming that we are the first to discover it. However, the perturbation design was always thought to be heuristic and its "goodness" for the stochastic optimal control problem was essentially unexplored. Notable as an exception is the reference [8] that considers the problem of how good the deterministic feedback law is for the stochastic system, which is shown to be $O(\epsilon^4)$. However, that paper assumes the availability of the optimal deterministic feedback law which is the solution of the deterministic Hamilton-Jacobi-Bellman (HJB) equation (the DP equation in continuous time problems), which, in itself, is intractable as noted by Fleming as the "practical difficulty" in this work (pgs. 475–476 of [8]). However, MPC, by repeatedly solving the deterministic optimal control problem at every time step, implicitly furnishes the *deterministic feedback law*, and thus, offers the solution to the *practical dilemma* above. The field of MPC, of course, was developed almost two decades after Fleming's work, while stochastic MPC/ MPC-under-uncertainty was explored only starting at the turn of millennium [13]. Thus, this connection was lost and never really explored in the MPC literature. This connection is critical if we want to *tractably* extend MPC to stochastic systems in a *theoretically justifiable fashion*, in the sense that in much of the stochastic MPC literature, these two aspects are at cross purposes to each other thereby preventing a satisfactory

resolution. Thus, the MPC replanning logic is well justified theoretically, even when applied to a stochastic system.

In fact, with a few further developments, and adaptation of Fleming's work to discrete time finite horizon problems, and if the linear feedback gain is modified suitably, the perturbation design also becomes $O(\epsilon^4)$ near-optimal. Due to paucity of space, we postpone this result to a future paper, however, for the sake of completeness and the reader's benefit, the result is included in the supplementary document. *The ultimate takeaway is that the implicit MPC feedback law is an excellent approximation to the optimal stochastic policy, however, a T-LQR type perturbation feedback design is much cheaper computationally, while retaining identical near-optimality guarantees as MPC.*

3.2 Multi-agent setting

Now, we generalize the above result to a class of multi-agent problems. We consider a set of agents that are transition independent, i.e, their dynamics are independent of each other. For simplicity, we also assume that the agents have perfect state measurements. Let the system equations for the agents be given by: $\mathbf{x}_{t+1}^j = f(\mathbf{x}_t^j) + \mathbf{B}_t^j(\mathbf{u}_t^j + \epsilon \mathbf{w}_t^j)$, where $j = 1, 2, \ldots, M$ denotes the j^{th} agent. (We have assumed the control affine dynamics for simplicity). Further, let us assume that we are interested in the minimization of the joint cost of the agents given by $\mathcal{J} = \sum_{t=0}^{T-1} c(\mathbf{X}_t, \mathbf{U}_t) + \Phi(\mathbf{X}_T)$, where $\mathbf{X}_t = [\mathbf{x}_t^1, \ldots, \mathbf{x}_t^M]$, and $\mathbf{U}_t = [\mathbf{u}_t^1, \ldots, \mathbf{u}_t^M]$ are the joint state and control action of the system. The objective of the multi-agent problem is minimize the expected value of the cost $\mathbb{E}[\mathcal{J}]$ over the joint feedback policy $\mathbf{U}_t(\cdot)$. The decoupling result holds here too and thus the multi-agent planning problem can be separated into an open and closed-loop problem. The open-loop problem consists of optimizing the joint nominal cost of the agents subject to the individual dynamics.
Multi-Agent Open-Loop Problem:

$$\overline{\mathcal{J}} = \min_{\overline{\mathbf{U}}_t} \sum_{t=0}^{T-1} c(\overline{\mathbf{X}}_t, \overline{\mathbf{U}}_t) + \Phi(\overline{\mathbf{X}}_T), \tag{7}$$

subject to the nominal agent dynamics $\overline{\mathbf{x}}_{t+1}^j = f(\overline{\mathbf{x}}_t^j) + \mathbf{B}_t^j \overline{\mathbf{u}}_t^j$. The closed-loop, in general, consists of optimizing the variance of the cost $\mathcal{J}$, given by $\text{Var}[\delta\mathcal{J}_1^\ell]$, where $\delta\mathcal{J}_1^\ell = \sum_t \mathbf{C}_t \delta\mathbf{X}_t^l$ for suitably defined $\mathbf{C}_t$, and $\delta\mathbf{X}_t^\ell = [\delta\mathbf{x}_t^1, \ldots, \delta\mathbf{x}_t^M]$, where the perturbations $\delta\mathbf{x}_t^j$ of the j^{th} agent's state is governed by the decoupled linear multi-agent system $\delta\mathbf{x}_t^j = \mathbf{A}_t \delta\mathbf{x}_t^j + \mathbf{B}_t^j \delta\mathbf{u}_t^j + \epsilon \mathbf{B}_t^j \mathbf{w}_t^j$. This design problem does not have a standard solution but recall that we are not really interested in obtaining the optimal closed-loop variance, but rather a good variance. Thus, we can instead solve a surrogate LQR problem given the cost function $\delta\mathcal{J}_{\text{MTLQR}} = \sum_{t=0}^{T-1} \sum_j \delta{\mathbf{x}_t^j}^\mathsf{T} \mathbf{Q}^j \delta\mathbf{x}_t^j + \delta{\mathbf{u}_t^j}^\mathsf{T} \mathbf{R} \delta\mathbf{u}_t^j + \sum_j \delta{\mathbf{x}_T^j}^\mathsf{T} \mathbf{Q}_f^j \delta\mathbf{x}_T^j$. Since the cost function itself is decoupled, the surrogate LQR design degenerates into a decoupled LQR design for each agent.

Surrogate Decoupled LQR Problem:

$$\delta\mathcal{J}^j = \min_{\substack{\mathbf{u}_t^j \\ \mathbf{w}_t^j}} \mathbb{E}\left[\sum_{t=0}^{T-1} \delta\mathbf{x}_t^{j\mathsf{T}}\mathbf{Q}^j\delta\mathbf{x}_t^j + \delta\mathbf{u}_t^{j\mathsf{T}}\mathbf{R}\delta\mathbf{u}_t^j + \delta\mathbf{x}_T^{j\mathsf{T}}\mathbf{Q}_f^j\delta\mathbf{x}_T^j\right], \text{subject to the}$$

linear decoupled agent dynamics $\delta\mathbf{x}_t^j = \mathbf{A}_t\delta\mathbf{x}_t^j + \mathbf{B}_t^j\delta\mathbf{u}_t^j + \epsilon\mathbf{B}_t^j\mathbf{w}_t^j$.

Remark 1. Note that the above decoupled feedback design results in a spatial decoupling between the agents in the sense that, at least in the small noise regime, after their "initial joint plan" is made, the agents never need to communicate with each other in order to complete their missions. However, note that the joint plan requires communication.

3.3 Planning Complexity versus Uncertainty

The decoupling principle outlined above shows that the complexity of planning can be drastically reduced while still retaining near optimal performance for sufficiently small noise (i.e., parameter $\epsilon \ll 1$). Nonetheless, the skeptical reader might argue that this result holds only for low values of ϵ and thus, its applicability for higher noise levels is suspect. Still, because the result is second order, it hints that near optimality might be over a reasonably large ϵ. Naturally, the question is *'will it hold for medium to higher levels of noise?'* We purposely leave the terminology of medium to high noise nebulous but what we mean shall become clear from our experiments.

Preview of the Results. In this paper, we illustrate the degree to which the above result still holds when we allow periodic replanning of the nominal trajectory in T-LQR in an event triggered fashion, dubbed T-LQR2. Here, we shall use MPC as a "gold standard" for comparison since the true stochastic control problem is intractable, and owing to Fleming's result [8], the MPC policy is $O(\epsilon^4)$ near-optimal when compared to the true stochastic policy. In fact, we can make an identical strong $O(\epsilon^4)$ claim for T-LQR as well if the *linear feedback gain is designed carefully*, but owing to the paucity of space, testing with this careful feedback design is left to a future paper. Here, we show that though the number of replanning operations in T-LQR2 increases the planning burden over T-LQR, it is still much reduced when compared to MPC, which replans continually. The ability to trigger replanning means that T-LQR2 can always produce solutions with the same quality as MPC, albeit by demanding the same computational cost as MPC in instances when replanning is triggered. But for moderate levels of noise, T-LQR2 can produce comparable quality output to MPC with substantial computational savings.

In the high noise regime, replanning is more frequent but we shall see that there is another consideration at play. Namely, that the effective planning horizon decreases and there seems no benefit in planning all the way to the end rather than considering only a few steps ahead, and in fact, in some cases, it can be

harmful to consider the distant future. Noting that as the planning horizon decreases, planning complexity decreases, this helps recover tractability even in this regime.

Thus, while lower levels of noise render the planning problem tractable due to the decoupling result requiring no replanning, planning under medium noise remains tractable due to only occasional replanning, while for high levels of noise, tractability ensues because the planning horizon should shrink as the uncertainty increases. When noise inundates the system, long-term predictions become so uncertain that the best-laid plans will very likely run awry, and thus, it would be wasteful to invest significant time thinking very far ahead. To examine this somewhat intuitive truth more quantitatively, the parameter ϵ will be a knob we adjust, exploring these aspects in the subsequent empirical analysis. *We reiterate that the notion of low, medium and high noise regimes may seem somewhat vague, however, we provide precise definitions of these regimes using our empirical results later in this paper.*

4 The Planning Algorithms

The preliminaries and the algorithms are explained below:

4.1 Deterministic Optimal Control Problem:

Given the initial state $\mathbf{x}_0$ of the system, the solution to the deterministic OCP is given by (4), s.t. $\mathbf{x}_{t+1} = f(\mathbf{x}_t) + \mathbf{B}_t\mathbf{u}_t,\ \mathbf{u}_{\min} \leq \mathbf{u}_t \leq \mathbf{u}_{\max},\ |\mathbf{u}_t - \mathbf{u}_{t-1}| \leq \Delta\mathbf{u}_{\max}$. The last two constraint model physical limits that impose upper bounds and lower bounds on control inputs and rate of change of control inputs. The solution to the above problem gives the open-loop control inputs $\overline{\mathbf{u}}_{0:T-1}$ for the system. For our problem, we take a quadratic cost function for state and control as $c_t(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{x}_t^\mathsf{T}\mathbf{W}^x\mathbf{x}_t + \mathbf{u}_t^\mathsf{T}\mathbf{W}^u\mathbf{u}_t$, $c_T(\mathbf{x}_T) = \mathbf{x}_T^\mathsf{T}\mathbf{W}_f^x\mathbf{x}_T$, where $\mathbf{W}^x,\ \mathbf{W}_f^x \succeq \mathbf{0}$ and $\mathbf{W}^u \succ \mathbf{0}$.

4.2 Model Predictive Control (MPC):

We employ the non-linear MPC algorithm due to the non-linearities associated with the motion model. The MPC algorithm implemented here solves the deterministic OCP (4) at every time step, applies the control inputs computed for the first instant and uses the rest of the solution as an initial guess for the subsequent computation. In the next step, the current state of the system is measured and used as the initial state and the process is repeated.

4.3 Short Horizon MPC (MPC-SH):

We also implement a variant of MPC which is typically used in practical applications where it solves the OCP only for a short horizon rather than the entire horizon at every step. At the next step, a new optimization is solved over the shifted horizon. This implementation gives a greedy solution but is computationally easier to solve. It also has certain advantageous properties in high noise

cases which will be discussed in the results section. We denote the short planning horizon as H_c also called as the control horizon, upto which the controls are computed.

4.4 Trajectory Optimised Linear Quadratic Regulator (T-LQR):

As discussed in Section 3, stochastic optimal control problem can be decoupled and solved by designing an optimal open-loop (nominal) trajectory and a decentralized LQR policy to track the nominal.
Design of nominal trajectory: The nominal trajectory is generated by first finding the optimal open-loop control sequence by solving the deterministic OCP (4) for the system. Then, using the computed control inputs and the noise-free dynamics, the sequence of states traversed $\overline{\mathbf{x}}_{0:T}$ can be calculated.
Design of feedback policy: In order to design the LQR controller, the system is first linearised about the nominal trajectory ($\overline{\mathbf{x}}_{0:T}$, $\overline{\mathbf{u}}_{0:T-1}$). Using the linear time-varying system, the feedback policy is determined by minimizing a quadratic cost as shown in (6). The linear quadratic stochastic control problem (6) can be easily solved using the Riccati equation and the resulting policy is $\delta\mathbf{u}_t = -\mathbf{L}_t\delta\mathbf{x}_t^{\ell}$. The feedback gain and the Riccati equations are given by $\mathbf{L}_t = \left(\mathbf{R} + \mathbf{B}_t^T\mathbf{P}_{t+1}\mathbf{B}_t\right)^{-1}\mathbf{B}_t^{\mathsf{T}}\mathbf{P}_{t+1}\mathbf{A}_t$ and $\mathbf{P}_t = \mathbf{A}_t^{\mathsf{T}}\mathbf{P}_{t+1}\mathbf{A}_t - \mathbf{A}_t^{\mathsf{T}}\mathbf{P}_{t+1}\mathbf{B}_t\mathbf{L}_t + \mathbf{Q}$, respectively where $\mathbf{Q}_f, \mathbf{Q} \succeq \mathbf{0}, \mathbf{R} \succ \mathbf{0}$ are the weight matrices for states and control and the terminal condition is $\mathbf{P}_T = \mathbf{Q}_f$.

4.5 T-LQR with Replanning (T-LQR2):

T-LQR performs well at low noise levels, but at medium and high noise levels the system tends to deviate from the nominal. So, we define a threshold $J_{\text{thresh}} = \frac{J_{0:t} - \overline{J}_{0:t}}{\overline{J}_{0:t}}$, where $J_{0:t}$ denotes the actual cost during execution till time t. The factor J_{thresh} measures the percentage deviation of the online trajectory from the nominal, and replanning is triggered for the system from the current state for the remainder of the horizon if the deviation exceeds it. Other replanning criteria such as state deviation can also be considered but we stick to the cost deviation in the following [1]. Note that if we set $J_{\text{thresh}} = 0$, T-LQR2 reduces to MPC. The calculation of the new nominal trajectory and LQR gains are carried out similarly to the explanation in Section 4.4. A generic algorithm for T-LQR and T-LQR2 is shown in Algorithm 1. The implementations of all the algorithms are available at https://github.com/MohamedNaveed/Stochastic_Optimal_Control_algos.

4.6 Multi-Agent versions

The MPC version of the multi-agent planning problem is reasonably straightforward except that the complexity of the planning increased (exponentially) in the number of agents. Also, we note that the agents have to always communicate with each other in order to do the planning. The Multi-agent Trajectory-optimised LQR (MT-LQR) version is also relatively straightforward in that the agents plan

[1] In the absence of a running cost, a criterion such as state deviation could be used. Since we aim to optimize the cost, a criterion based on cost seems more reasonable.

Algorithm 1: T-LQR2 algorithm

Input: initial state $\mathbf{x}_0$, final state $\mathbf{x}_g$, time horizon T, replan threshold J_{thresh}, time step Δt, system and environment parameters $\mathcal{P}$.

Function *Plan*($\boldsymbol{x}_0, \boldsymbol{x}_g, T, \boldsymbol{u}_{init}, \boldsymbol{u}_{guess}, \mathcal{P}$) **is**
  $\overline{\mathbf{u}}_{0:T-1} \leftarrow \text{OCP}(\mathbf{x}_0, \mathbf{x}_g, T, \mathbf{u}_{\text{init}}, \mathbf{u}_{\text{guess}}, \mathcal{P})$
  $\overline{\mathbf{x}}_{t+1} \leftarrow f(\overline{\mathbf{x}}_t) + \mathbf{B}_t \overline{\mathbf{u}}_t; \quad t = 0, 1, \cdots, T-1.$
  $\mathbf{L}_{0:T-1} \leftarrow \textit{Compute\_LQR\_Gain}(\overline{\mathbf{x}}_{0:T-1}, \overline{\mathbf{u}}_{0:T-1})$
  return $\overline{\mathbf{x}}_{0:T}, \overline{\mathbf{u}}_{0:T-1}, \mathbf{L}_{0:T-1}$
end
Function *Main()* **is**
  $\overline{\mathbf{x}}_{0:T}, \overline{\mathbf{u}}_{0:T-1}, \mathbf{L}_{0:T-1} \leftarrow \text{Plan}(\mathbf{x}_0, \mathbf{x}_g, T, \mathbf{0}, \mathbf{u}_{\text{guess}}, \mathcal{P})$
  for $t \leftarrow 0$ **to** $T-1$ **do**
    $\mathbf{u}_t \leftarrow \text{Constrain}(\overline{\mathbf{u}}_t - \mathbf{L}_t(\mathbf{x}_t - \overline{\mathbf{x}}_t))$ `// Enforce limits`
    $\mathbf{x}_{t+1} \leftarrow f(\mathbf{x}_t) + \mathbf{B}_t(\mathbf{u}_t + \epsilon \mathbf{w}_t)$
    if $(J_{0:t} - \overline{J}_{0:t})/\overline{J}_{0:t} > J_{thresh}$ **then** `// Replan?`
      $\overline{\mathbf{x}}_{t+1:T}, \overline{\mathbf{u}}_{t+1:T-1}, \mathbf{L}_{t+1:T-1} \leftarrow \text{Plan}(\mathbf{x}_{t+1}, \mathbf{x}_g, T-t-1, \mathbf{u}_t, \mathbf{u}_{\text{guess}}, \mathcal{P})$
    end
  end
end

the nominal path jointly once, and then the agents each track their individual paths using their decoupled feedback controllers. There is no communication whatsoever between the agents during this operation.

The MT-LQR2 version is a little more subtle. The agents have to periodically replan when the total cost deviates more than J_{thresh} away from the nominal, i.e., the agents do not communicate until the need to replan arises. In general, the system would need to detect this in a distributed fashion, and trigger replanning.

5 Simulation Results:

We test the performance of the algorithms extensively in three different non-linear models namely, the car-like robot model, car with two trailers and a quadrotor. Due to space constraints, only the results for the car-like robot are shown below, however, that the trends are generalizable can be seen from the results on the other models that are shown in the supplementary material. Numerical optimization is carried out using `CasADi` framework [1] with `Ipopt` [24] NLP solver in `Python`. To provide a good estimate of the performance, the results presented were averaged from 100 simulations for every value of noise considered. Simulations were carried out in parallel across 100 cores in a cluster equipped with Intel Xeon 2.5GHz E5-2670 v2 10-core processors.

Car-like robot model:

The car-like robot considered in our work has the motion model described by $x_{t+1} = x_t + v_t \cos(\theta_t)\Delta t$, $y_{t+1} = y_t + v_t \sin(\theta_t)\Delta t$, $\theta_{t+1} = \theta_t + \frac{v_t}{L}\tan(\phi_t)\Delta t$, $\phi_{t+1} =$

$\phi_t + \omega_t \Delta t$, where $(x_t, y_t, \theta_t, \phi_t)^\mathsf{T}$ denote the robot's state vector namely, robot's x and y position, orientation and steering angle at time t. Also, $(v_t, \omega_t)^\mathsf{T}$ is the control vector and denotes the robot's linear velocity and angular velocity (i.e., steering). Here Δt is the discretization of the time step.

Noise characterization:

We add zero mean independent identically distributed (i.i.d), random sequences $(\mathbf{w}_t)$ as actuator noise to test the performance of the control scheme. The standard deviation of the noise is ϵ times the maximum value of the corresponding control input, where ϵ is a scaling factor which is varied during testing, that is: $\mathbf{w}_t = \mathbf{u}_{\max}\boldsymbol{\nu}; \quad \boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the noise is added as $\epsilon\mathbf{w}_t$. Note that, we enforce the constraints in the control inputs before the addition of noise, so the controls can even take a value higher after noise is added. The analyses can be done with process noise as well, but ϵ loses meaning in such a scenario and the plots would just shift depending on the variance of $\mathbf{w}_t$. Since all the algorithms use the same noise model and having been tested in an extensive range of values, the requirement for a process noise model is not really necessary.

5.1 Single agent setting:

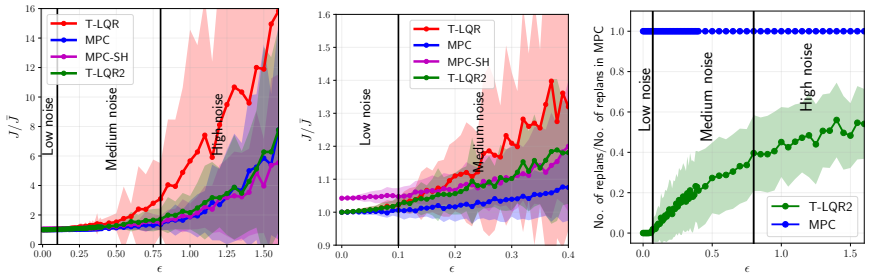
A car-like robot is considered and is tasked to move from a given initial pose to a goal pose. The environment of the robot is shown in Figure 4. The experiment is done for all the control schemes discussed and their performance for different levels of noise are shown in Figure 2.

5.2 Multi-agent setting:

A labelled point-to-point transition problem with 3 car-like robots is considered where each agent is assigned a fixed destination which cannot be exchanged with another agent. The performance of the algorithms is shown in Figure 3. The cost function involves the state and control costs for the entire system similar to the single agent case. One major addition to the cost function is the penalty function to avoid inter-agent collisions which is given by $\Psi^{(i,j)} = \mathrm{M}\exp\left(-(\|\mathbf{p}_t^i - \mathbf{p}_t^j\|_2^2 - r_{\text{thresh}}^2)\right)$ where $\mathrm{M} > 0$ is a scaling factor, $\mathbf{p}_t^i = (x_t^i, y_t^j)$ and r_{thresh} is the desired minimum distance the agents should keep between themselves.

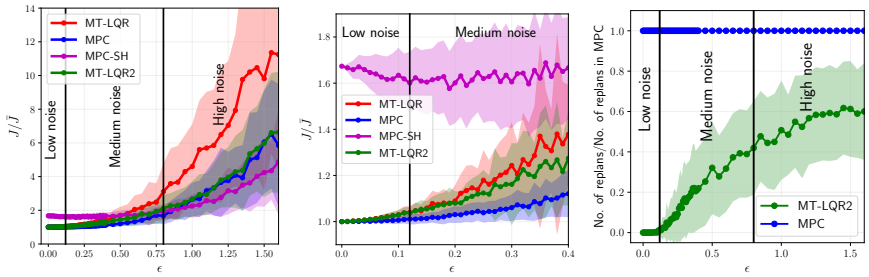
5.3 Definition of noise regimes and discussion of the results:

Here we go on to define what exactly we mean by low, medium and high noise. The low noise regime as labelled in Figure 2b and 3b is the noise level at which the decoupled feedback law (T-LQR and MT-LQR) shows near-optimal performance compared to MPC and does not require any replanning operations. Beyond a limit replanning (T-LQR2 and MT-LQR2) is essential to constrain the cost from deviating away from the optimal and we call this as the medium noise regime.



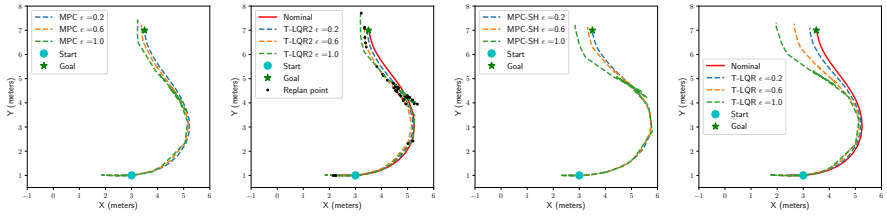
(a) Full noise spectrum (b) Enhanced detail $0 \leq \epsilon \leq 0.4$ (c) Replanning operations

Fig. 2: Cost evolution of the different algorithms for varying noise for a single agent. Control Horizon (H_c) used for MPC-SH was 7. $J_{\text{thresh}} = 2\%$ was the replanning threshold used. $J/\bar{J}$ is the ratio of the cost incurred during execution to the nominal cost and is used as the performance measure throughout the paper. The nominal cost $\bar{J}$ which is calculated by solving the deterministic OCP for the total time horizon, just acts as a normalizing factor here. The solid line in the plots indicates the mean and the shade indicates the standard deviation of the corresponding metric.



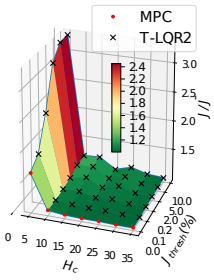
(a) Full noise spectrum (b) Enhanced detail $0 \leq \epsilon \leq 0.4$ (c) Replanning operations

Fig. 3: Cost evolution of the different algorithms for varying noise for 3 agents. Control Horizon (H_c) used for MPC-SH was 7. $J_{\text{thresh}} = 2\%$ was the replanning threshold used.

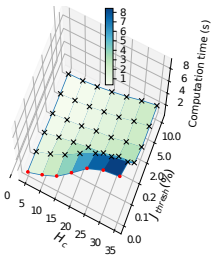


(a) MPC (b) T-LQR2 (c) MPC-SH (d) T-LQR

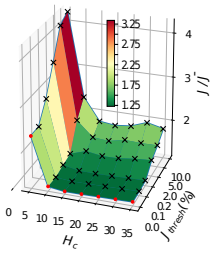
Fig. 4: The figure shows the paths taken by the robot using a particular algorithm and how they change as ϵ varies. The key takeaway is that the paths taken under T-LQR are close to MPC for small values of ϵ and deviate as ϵ increases, while those of T-LQR2 are very close to the paths taken under MPC. It validates our claim about near-optimal performance of the decoupled feedback law under small noises and how performance can be preserved by replanning whenever necessary under medium noises. A equivalent plot for a scenario with obstacles is shown in the supplementary material.



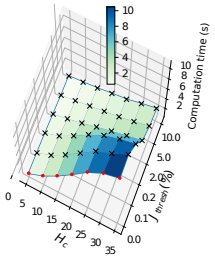
(a) Solution quality for 1 agent with $\epsilon = 0.1$.



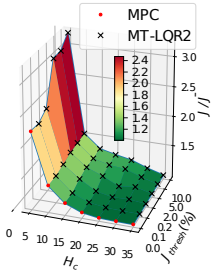
(b) Compute time for 1 agent with $\epsilon = 0.1$.



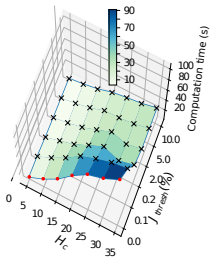
(c) Solution quality for 1 agent with $\epsilon = 0.7$.



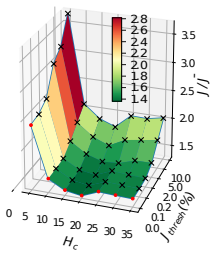
(d) Compute time for 1 agent with $\epsilon = 0.7$.



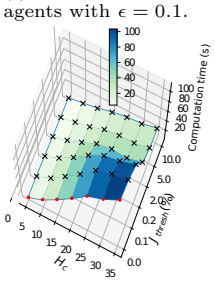
(e) Solution quality for 3 agents with $\epsilon = 0.1$.



(f) Compute time for 3 agents with $\epsilon = 0.1$.



(g) Solution quality for 3 agents with $\epsilon = 0.7$.



(h) Compute time for 3 agents with $\epsilon = 0.7$.

Fig. 5: We study the variation seen in cost incurred and computation time by changing the J_{thresh} and control horizon (H_c) in T-LQR2/MT-LQR2 and MPC for a single agent and a case with 3 agents. Figures 5a, 5b, 5c, 5d show for a single agent and 5e, 5f, 5g, 5h for 3 agents. 5a and 5b show the performance in terms of cost and computation time, respectively, for the same experiment at $\epsilon = 0.1$ (low noise). Similarly, (c) and (d) show for $\epsilon = 0.7$ (medium noise). To put into context, T-LQR2/MT-LQR2 replans less as J_{thresh} is increased which in turn leads to decrease in computation time. The computation time also decreases with decrease in H_c. Though MPC does not have a threshold for replanning, it is plotted at $J_{\text{thresh}} = 0\%$ since it replans at every time step. We desire the performance indices, cost ($J/\bar{J}$) and computation time to be small.

As seen from 5a, for a fixed H_c, the performance of T-LQR2 is same as MPC and does not degrade as J_{thresh} is increased (except when H_c is too small). 5b shows the computational savings, where T-LQR2 is much better than MPC. From 5a and 5b it can be inferred that the decoupled approach provides good performance with substantial savings in computation time. Similarly, 5c shows that T-LQR2 performance is near-optimal to MPC for small J_{thresh} and degrades as it is relaxed, which indicates the necessity of replanning to maintain optimality in the medium noise regime. The corresponding variation in computation time is shown in 5d, where T-LQR2 does slightly better. The computational savings can be increased further but by trading away optimality.

Similar interpretation can be made for the multi-agent case as seen from Figures 5e, 5f, 5g, 5h. It can be seen that the positives of using the decoupled approach are amplified in the multi-agent case.

Now, we analyse how H_c affects the performance. Decreasing H_c leads to greedy sub-optimal solutions. Though not clearly seen in single agent case, the decrease in performance is seen in the multi-agent case (Fig. 5e). But in high noises, decreasing the horizon does not lead to decrease in solution quality or sometimes even produces better solutions as seen in 5g.

Figure 2c and 3c show the significant difference in the number of replanning operations, which determines the computational effort, taken by the decoupled approach compared to MPC. The significant difference in computational time between MPC and T-LQR2 can be seen from Figure 5b. The trend is similar in the multi-agent case which again shows that the decoupling feedback policy is able to give computationally efficient solutions which are near-optimal in low noise cases by avoiding frequent replanning.

In the high noise regime, T-LQR2, MT-LQR2 and even MPC-SH perform on a par with MPC as seen from Figures 2a and 3a meaning, planning too far ahead is not beneficial at high noise levels. It can also be seen in Figure 5g that the performance for MPC as well as MT-LQR2 is best at $H_c = 20$. Planning for a shorter horizon also eases the computation burden as seen in Figure 5h. It can also be seen in Figure 3a where MPC-SH with $H_c = 7$ outperforms MPC with $H_c = 35$ at high noise levels which again show that the effective planning horizon decreases at the high noise regime.

6 Conclusions and Implications

In this paper, we have considered a class of stochastic motion planning problems for robotic systems over a wide range of uncertainty conditions parameterized in terms of a noise parameter ϵ. We have shown extensive empirical evidence that a simple generalization of a recently developed "decoupling principle" can lead to tractable planning without sacrificing performance for a wide range of noise levels. Future work will seek to treat the medium and high noise systems, considered here, analytically, and look to establish the near-optimality of the replanning scheme. Further, we shall consider the question of "when and how to replan" in a distributed fashion in the multi-agent setting, as well as relax the requirement of perfect state observation. It is also conjectured that by designing the linear feedback in a suitable fashion, the decoupling result can be made $O(\epsilon^4)$ near-optimal, thus making the algorithm theoretically as good as MPC owing to Fleming's result [8]. Further, an important limitation of the method is the smoothness of the nominal trajectory such that suitable Taylor expansions are possible, this breaks down when trajectories are non-smooth such as in hybrid systems like legged robots, or maneuvers have kinks for car-like robots such as in a tight parking application. It remains to be seen as to if, and how, one may extend the decoupling to such applications.

References

1. J. A E Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, In Press, 2018.
2. R. Akrour, A. Abdolmaleki, H. Abdulsamad, and G. Neumann. Model free trajectory optimization for reinforcement learning. In *Proc. of the ICML*, 2016.

3. C. Amato, G. Chowdhary, A. Geramifard, N. K. Ure, and M. J. Kochenderfer. Decentralized control of partially observable markov decision processes. In *Proc. IEEE Int. CDC*, pages 2398–2405, 2013.
4. D. P. Bertsekas. *Dynamic Programming and Optimal Control, vols I and II*. Athena Scientific, Cambridge, MA, 2012.
5. C. Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proc. of the 6th conference on Theoretical aspects of rationality and knowledge*, pages 195–210. Morgan Kaufmann Publishers Inc., 1996.
6. A. E. Bryson and Y. C. Ho. *Applied Optimal control*. Allied Publishers, NY, 1967.
7. L. Chisci, J. A. Rossiter, and G. Zappa. Systems with persistent disturbances: predictive control with restricted contraints. *Automatica*, 37:1019–1028, 2001.
8. W. H. Fleming. Stochastic control for small noise intensities. *SIAM Journal on Control*, 9(3):473–517, 1971.
9. W. Heemels, K. Johansson, and P. Tabuada. An introduction to event triggered and self triggered control. In *Proc. IEEE Int. CDC*, 2012.
10. S. Levine and P. Abbeel. Learning neural network policies with guided search under unknown dynamics. In *Advances in NIPS*, 2014.
11. S. Levine and K. Vladlen. Learning complex neural network policies with trajectory optimization. In *Proc. of the ICML*, 2014.
12. H. Li, Y. She, W. Yan, and K. Johansson. Periodic event-triggered distributed receding horizon control of dynamically decoupled linear systems. In *Proc. IFAC World Congress*, 2014.
13. D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50:2967–2986, 2014.
14. D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi. Tube based robust nonlinear model predictive control. *International journal of robust and nonlinear control*, 21:1341–1353, 2011.
15. F. A. Oliehoek. Decentralized pomdps. *Reinforcement Learning*, 2012.
16. F. A. Oliehoek and C. Amato. *A concise introduction to decentralized POMDPs*. Springer, 2016.
17. K. S. Parunandi and S. Chakravorty. T-pfc: A trajectory-optimized perturbation feedback control approach. *IEEE RA-L*, 4(4):3457–3464, Oct 2019.
18. D. V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:389–423, 2002.
19. J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill, Madison, WI, 2015.
20. J. A. Rossiter, B. Kouvaritakis, and M. J. Rice. A numerically stable state space approach to stable predictive control strategies. *Automatica*, 34:65–73, 1998.
21. S. Seuken and S. Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *Int. Conf. on AAMAS*, 17(2):190–250, 2008.
22. E. Theodorou, Y. Tassa, and E. Todorov. Stochastic differential dynamic programming. In *Proc. of the ACC*, 2010.
23. E. Todorov and Y. Tassa. Iterative local dynamic programming. In *Proc. of the IEEE Int. Symposium on ADP and RL.*, 2009.
24. A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 2006.
25. R. Wang, K. S. Parunandi, D. Yu, D. M. Kalathil, and S. Chakravorty. Decoupled data based approach for learning to control nonlinear dynamical systems. *CoRR*, abs/1904.08361, 2019.

SUPPLEMENTARY MATERIAL

In this document, we provide details of the decoupling result, a rudimentary analysis of the high noise regime, and more empirical results on different robotic models from that presented in the main paper.

7 A NEAR OPTIMAL DECOUPLING PRINCIPLE

We discuss in detail the decoupling principle described in Section 3.

We make the following assumptions for the simplicity of illustration. We assume that the dynamics given in (1) can be written in the form

$$x_{t+1} = f(x_t) + B_t u_t + \epsilon w_t, \tag{8}$$

where $\epsilon < 1$ is a small parameter. We also assume that the instantaneous cost $c(\cdot, \cdot)$ has the following simple form,

$$c(x, u) = l(x) + \frac{1}{2} u' R u. \tag{9}$$

We emphasis that these assumptions, quadratic control cost and affine in control dynamics, are purely for the simplicity of treatment. These assumptions can be omitted at the cost of increased notational complexity.
In the following subsections, we first characterize the performance of any feedback policy. Then, we use this characterization to provide $O(\epsilon^2)$ and $O(\epsilon^4)$ near-optimality results in the subsequent subsections.

7.1 Characterizing the Performance of a Feedback Policy

Consider a noiseless version of the system dynamics given by (8). We denote the "nominal" state trajectory as $\bar{x}_t$ and the "nominal" control as $\bar{u}_t$ where $u_t = \pi_t(x_t)$, where $\pi = (\pi_t)_{t=1}^{T-1}$ is a given control policy. The resulting dynamics without noise is given by $\bar{x}_{t+1} = f(\bar{x}_t) + B_t \bar{u}_t$.

Assuming that $f(\cdot)$ and $\pi_t(\cdot)$ are sufficiently smooth, we can linearize the dynamics about the nominal trajectory. Denoting $\delta x_t = x_t - \bar{x}_t, \delta u_t = u_t - \bar{u}_t$, we can express,

$$\delta x_{t+1} = A_t \delta x_t + B_t \delta u_t + S_t(\delta x_t) + \epsilon w_t, \tag{10}$$

$$\delta u_t = K_t \delta x_t + \tilde{S}_t(\delta x_t), \tag{11}$$

where $A_t = \frac{\partial f}{\partial x}|_{\bar{x}_t}$, $K_t = \frac{\partial \pi_t}{\partial x}|_{\bar{x}_t}$, and $S_t(\cdot), \tilde{S}_t(\cdot)$ are second and higher order terms in the respective expansions. Similarly, we can linearize the instantaneous cost $c(x_t, u_t)$ about the nominal values $(\bar{x}_t, \bar{u}_t)$ as,

$$c(x_t, u_t) = l(\bar{x}_t) + L_t \delta x_t + H_t(\delta x_t) + \frac{1}{2} \bar{u}_t' R \bar{u}_t + \delta u_t' R \bar{u}_t + \delta u_t' R \delta u_t, \tag{12}$$

$$c_T(x_T) = c_T(\bar{x}_T) + C_T \delta x_T + H_T(\delta x_T), \tag{13}$$

where $L_t = \frac{\partial l}{\partial x}|_{\bar{x}_t}$, $C_T = \frac{\partial c_T}{\partial x}|_{\bar{x}_t}$, and $H_t(\cdot)$ and $H_T(\cdot)$ are second and higher order terms in the respective expansions.

Using (10) and (11), we can write the closed loop dynamics of the trajectory $(\delta x_t)_{t=1}^T$ as,

$$\delta x_{t+1} = \underbrace{(A_t + B_t K_t)}_{\bar{A}_t} \delta x_t + \underbrace{\{B_t \tilde{S}_t(\delta x_t) + S_t(\delta x_t)\}}_{\bar{S}_t(\delta x_t)} + \epsilon w_t, \tag{14}$$

where $\bar{A}_t$ represents the linear part of the closed loop systems and the term $\bar{S}_t(.)$ represents the second and higher order terms in the closed loop system. Similarly, the closed loop incremental cost given in (12) can be expressed as

$$\begin{aligned} c(x_t, u_t) = &\underbrace{\{l(\bar{x}_t) + \frac{1}{2}\bar{u}_t' R \bar{u}_t\}}_{\bar{c}_t} + \underbrace{[L_t + \bar{u}_t' R K_t]}_{\bar{C}_t} \delta x_t \\ &+ \underbrace{(K_t \delta x_t + \tilde{S}_t(\delta x_t))' R (K_t \delta x_t + \tilde{S}_t(\delta x_t))}_{\bar{H}_t(\delta x_t)}. \end{aligned} \tag{15}$$

Therefore, the cumulative cost of any given closed loop trajectory $(x_t, u_t)_{t=1}^T$ can be expressed as,

$$\begin{aligned} J^\pi &= \sum_{t=1}^{T-1} c(x_t, u_t = \pi_t(x_t)) + c_T(x_T) \\ &= \sum_{t=1}^{T} \bar{c}_t + \sum_{t=1}^{T} \bar{C}_t \delta x_t + \sum_{t=1}^{T} \bar{H}_t(\delta x_t), \end{aligned} \tag{16}$$

where $\bar{c}_T = c_T(\bar{x}_T), \bar{C}_T = C_T$.

We first show the following critical result.

Lemma 1. *Given any sample path, the state perturbation equation*

$$\delta x_{t+1} = \bar{A}_t \delta x_t + \bar{S}_t(\delta x_t) + \epsilon w_t$$

given in (14) *can be equivalently characterized as*

$$\delta x_t = \delta x_t^l + e_t, \ \delta x_{t+1}^l = \bar{A}_t \delta x_t^l + \epsilon w_t \tag{17}$$

where e_t is an $O(\epsilon^2)$ function that depends on the entire noise history $\{w_0, w_1, \cdots w_t\}$ and δx_t^l evolves according to the linear closed loop system. Furthermore, $e_t = e_t^{(2)} + O(\epsilon^3)$, where $e_t^{(2)} = \bar{A}_{t-1} e_{t-1}^{(2)} + \delta x_t^{l'} \bar{S}_{t-1}^{(2)} \delta x_t^l$, $e_0^{(2)} = 0$, and $\bar{S}_t^{(2)}$ represents the Hessian corresponding to the Taylor series expansion of the function $\bar{S}_t(.)$.

Proof. We only consider the case when the state x_t is scalar, the vector case is straightforward to derive and only requires a more complex notation.

We proceed by induction. The first general instance of the recursion occurs at $t = 3$. It can be shown that:

$$\delta x_3 = \underbrace{(\bar{A}_2\bar{A}_1(\epsilon w_0) + \bar{A}_2(\epsilon w_1) + \epsilon w_2)}_{\delta x_3^l} + \underbrace{\{\bar{A}_2\bar{S}_1(\epsilon w_0) + \bar{S}_2(\bar{A}_1(\epsilon w_0) + \epsilon w_1 + \bar{S}_1(\epsilon w_0))\}}_{e_3}. \tag{18}$$

Noting that $\bar{S}_1(.)$ and $\bar{S}_2(.)$ are second and higher order terms, it follows that e_3 is $O(\epsilon^2)$.
Suppose now that $\delta x_t = \delta x_t^l + e_t$ where e_t is $O(\epsilon^2)$. Then:

$$\begin{aligned}\delta x_{t+1} &= \bar{A}_{t+1}(\delta x_t^l + e_t) + \epsilon w_t + \bar{S}_{t+1}(\delta x_t),\\ &= \underbrace{(\bar{A}_{t+1}\delta x_t^l + \epsilon w_t)}_{\delta x_{t+1}^l} + \underbrace{\{\bar{A}_{t+1}e_t + \bar{S}_{t+1}(\delta x_t)\}}_{e_{t+1}}.\end{aligned} \tag{19}$$

Noting that $\bar{S}_t$ is $O(\epsilon^2)$ and that e_t is $O(\epsilon^2)$ by assumption, the result follows that e_{t+1} is $O(\epsilon^2)$.
Now, let us take a closer look at the term e_t and again proceed by induction. It is clear that $e_1 = e_1^{(2)} = 0$. Next, it can be seen that $e_2 = \bar{A}_1 e_1^{(2)} + S_1^{(2)}(\delta x_1^l)^2 + O(\epsilon^3) = \bar{S}_1^{(2)}(\epsilon\omega_0)^2 + O(\epsilon^3)$, which shows the recursion is valid for $t = 2$ given it is so for $t = 1$.
Suppose that it is true for t. Then:

$$\begin{aligned}\delta x_{t+1} &= \bar{A}_t\delta x_t + S_t(\delta x_t) + \epsilon\omega_t,\\ &= \bar{A}_t(\delta x_t^l + e_t) + S_t(\delta x_t^l + e_t) + \epsilon\omega_t,\\ &= \underbrace{(\bar{A}_t\delta x_t^l + \epsilon\omega_t)}_{\delta x_{t+!}^l} + \underbrace{\bar{A}_t e_t^{(2)} + S_t^{(2)}(\delta x_t^l)^2}_{e_{t+1}^{(2)}} + O(\epsilon^3),\end{aligned} \tag{20}$$

where the last line follows because $e_t = e_t^{(2)} + O(\epsilon^3)$, and $\bar{S}_t(.)$ contains second and higher order terms only. This completes the induction and the proof.

Next, we have the following result for the expansion of the cost to go function J^π.

Lemma 2. *Given any sample path, the cost-to-go under a policy can be expanded as:*

$$J^\pi = \underbrace{\sum_t \bar{c}_t}_{\bar{J}^\pi} + \underbrace{\sum_t \bar{C}_t\delta x_t^l}_{\delta J_1^\pi} + \underbrace{\sum_t \delta x_t^{l'}\bar{H}_t^{(2)}\delta x_t^l + \bar{C}_t e_t^{(2)}}_{\delta J_2^\pi} + O(\epsilon^3), \tag{21}$$

where $\bar{H}_t^{(2)}$ denotes the second order coefficient of the Taylor expansion of $\bar{H}_t(.)$.

Proof. We have that:

$$\begin{aligned} J^\pi &= \sum_t \bar{c}_t + \sum_t \bar{C}_t(\delta x_t^l + e_t) + \sum_t \bar{H}_t(\delta x_t^l + e_t), \\ &= \sum_t \bar{c}_t + \sum_t \bar{C}_t \delta x_t^l + \sum_t \delta x_t^{l'} \bar{H}_t^{(2)} \delta x_t^l + \bar{C}_t e_t^{(2)} + O(\epsilon^3), \end{aligned}$$

where the last line of the equation above follows from an application of Lemma 1.

Now, we show the following important result.

Proposition 1.

$$\begin{aligned} \tilde{J}^\pi &= \mathbb{E}[J^\pi] = \bar{J}^\pi + O(\epsilon^2), \\ Var(J^\pi) &= \underbrace{Var(\delta J_1^\pi)}_{O(\epsilon^2)} + O(\epsilon^4). \end{aligned}$$

Proof. It is useful to first write the sample path cost in a slightly different fashion. It can be seen that given sufficient smoothness of the requisite functions, the cost of any sample path can be expanded as follows:

$$J^\pi = \bar{J}^\pi + \epsilon J_1^\pi + \epsilon^2 J_2^\pi + \epsilon^3 J_3^\pi + \epsilon^4 J_4^\pi + \mathcal{R},$$

where:

$$\begin{aligned} J_1^\pi &= \mathcal{J}^1 \bar{\omega}, \\ J_2^\pi &= \bar{\omega}' \mathcal{J}^2 \bar{\omega}, \end{aligned}$$

and so on for J_3^π, J_4^π respectively, where $\mathcal{J}^i$ are constant matrices (tensors) of suitable dimensions, and $\bar{\omega} = [\omega_1, \cdots \omega_N]$. Further, the remainder function $\mathcal{R}$ is an $o(\epsilon^4)$ function in the sense that $\epsilon^{-4}\mathcal{R} \to 0$ as $\epsilon \to 0$.
Further, due to the whiteness of the noise sequences $\bar{\omega}$, it follows that $E[J_1^\pi] = 0$, and $E[J_3^\pi] = 0$, since these terms are made of odd valued products of the noise sequences, while $E[J_2^\pi], E[J_4^\pi]$ are both finite owing to the finiteness of the moments of the noise values and the initial condition. Further $lim_{\epsilon \to 0} \epsilon^{-4} E[\mathcal{R}] = E[\lim_\epsilon \epsilon^{-4}\mathcal{R}] = 0$, i.e., $E[\mathcal{R}]$ is $o(\epsilon^4)$.
Therefore, using Lemma 2, and taking expectations on both sides, we obtain:

$$E[J^\pi] = \bar{J}^\pi + E[\epsilon J_1^\pi] + E[\epsilon^2 J_2^\pi] + O(\epsilon^4) = \bar{J}^\pi + O(\epsilon^2),$$

since $E[J_1^\pi] = 0$, and $E[\epsilon^2 J_2^{\pi,2}]$ is $O(\epsilon^2)$ due to the fact that $E[J_2^\pi] < \infty$.
Next, using Lemma 2, and taking the variances on both sides, and doing some work, we have:

$$\begin{aligned} Var[J^\pi] &= Var[\epsilon J_1^\pi] + E[\epsilon J_1^\pi \epsilon^2 J_2^\pi] + Var[\epsilon^2 J_2^\pi] + o(\epsilon^4) \\ &= Var[\delta J_1^\pi] + O(\epsilon^4), \end{aligned} \tag{22}$$

where the second equality follows from the fact that $E[\epsilon J_1^\pi \epsilon^2 J_2^\pi] = 0$ (proved in the appendix), and $Var[J_2^\pi] < \infty$. This completes the proof of the result.

A further consequence of the result above is the following. Suppose that given a policy $\pi_t(.)$, we only consider the linear part, i.e., the linear approximation $\pi_t^l(x_t) = \bar{u}_t + K_t\delta x_t$. However, according to Lemma 2, the ϵ^2 terms in the expansion of the cost of any sample path solely result from the linear closed loop system. Therefore, it follows that the sample path cost under the full policy $\pi_t(.)$ and the linear policy $\pi_t^l(.)$ agree up to the ϵ^2 term. Therefore, it follows that $E[J^\pi] - E[J^{\pi^l}] = O(\epsilon^4)$! We summarize this result in the following:

Proposition 2. *Let $\pi_t(.)$ be any given feedback policy. Let $\pi_t^l(x_t) = \bar{u}_t + K_t\delta x_t$ be the linear approximation of the policy. Then, the error in the expected cost to go under the two policies, $E[J^\pi] - E[J^{\pi^l}] = O(\epsilon^4)$.*

The above two results in Propositions 1 and 2 will form the basis of an $O(\epsilon^2)$ and an $O(\epsilon^4)$ decoupling result in the following subsections.

7.2 An $O(\epsilon^2)$ Near-Optimal Decoupled Approach for Closed Loop Control

The following observations can now be made from Proposition 1.

Remark 2 (Expected cost-to-go). Recall that $u_t = \pi_t(x_t) = \bar{u}_t + K_t\delta x_t + \tilde{S}_t(\delta x_t)$. However, note that due to Proposition 1, the expected cost-to-go, $\tilde{J}^\pi$, is determined almost solely (within $O(\epsilon^2)$) by the nominal control action sequence $\bar{u}_t$. In other words, the linear and higher order feedback terms have only $O(\epsilon^2)$ effect on the expected cost-to-go function.

Remark 3 (Variance of cost-to-go). Given the nominal control action $\bar{u}_t$, the variance of the cost-to-go, which is $O(\epsilon^2)$, is determined overwhelmingly (within $O(\epsilon^4)$) by the linear feedback term $K_t\delta x_t$, i.e., by the variance of the linear perturbation of the cost-to-go, δJ_1^π, under the linear closed loop system $\delta x_{t+1}^l = (A_t + B_tK_t)\delta x_t^l + \epsilon w_t$.

Proposition 1 and the remarks above suggest that an open loop control super imposed with a closed loop control for the perturbed linear system may be approximately optimal. We delineate this idea below.

Open Loop Design. First, we design an optimal (open loop) control sequence $\bar{u}_t^*$ for the noiseless system. More precisely,

$$\begin{aligned}(\bar{u}_t^*)_{t=1}^{T-1} &= \arg\min_{(\tilde{u}_t)_{t=1}^{T-1}} \sum_{t=1}^{T-1} c(\bar{x}_t, \tilde{u}_t) + c_T(\bar{x}_T), \\ \bar{x}_{t+1} &= f(\bar{x}_t) + B_t\tilde{u}_t.\end{aligned} \tag{23}$$

Closed Loop Design. We find the optimal feedback gain K_t^* such that the variance of the linear closed loop system around the nominal path, $(\bar{x}_t, \bar{u}_t^*)$,

from the open loop design above, is minimized.

$$
\begin{aligned}
(K_t^*)_{t=1}^{T-1} &= \arg \min_{(K_t)_{t=1}^{T-1}} \ \mathrm{Var}(\delta J_1^\pi), \\
\delta J_1^\pi &= \sum_{t=1}^{T} \bar{C}_t x_t^l, \\
\delta x_{t+1}^l &= (A_t + B_t K_t)\delta x_t^l + \epsilon w_t.
\end{aligned} \tag{24}
$$

We now characterize the approximate closed loop policy below.

Proposition 3. *Construct a closed loop policy*

$$
\pi_t^*(x_t) = \bar{u}_t^* + K_t^* \delta x_t, \tag{25}
$$

where $\bar{u}_t^*$ is the solution of the open loop problem (23)*, and K_t^* is the solution of the closed loop problem* (24)*. Let π^o be the optimal closed loop policy. Then,*

$$
|\tilde{J}^{\pi *} - \tilde{J}^{\pi^o}| = O(\epsilon^2).
$$

Furthermore, among all policies with nominal control action $\bar{u}_t^*$, the variance of the cost-to-go under policy π_t^*, is within $O(\epsilon^4)$ of the variance of the policy with the minimum variance.

Proof. We have

$$
\begin{aligned}
\tilde{J}^{\pi^*} - \tilde{J}^{\pi^o} &= \tilde{J}^{\pi^*} - \bar{J}^{\pi^*} + \bar{J}^{\pi^*} - \tilde{J}^{\pi^o} \\
&\leq \tilde{J}^{\pi^*} - \bar{J}^{\pi^*} + \bar{J}^{\pi^o} - \tilde{J}^{\pi^o}
\end{aligned}
$$

The inequality above is due the fact that $\bar{J}^{\pi^*} \leq \bar{J}^{\pi^o}$, by definition of π^*. Now, using Proposition 1, we have that $|\tilde{J}^{\pi^*} - \bar{J}^{\pi^*}| = O(\epsilon^2)$, and $|\tilde{J}^{\pi^o} - \bar{J}^{\pi^o}| = O(\epsilon^2)$. Also, by definition, we have $\tilde{J}^{\pi^o} \leq \tilde{J}^{\pi^*}$. Then, from the above inequality, we get

$$
|\tilde{J}^{\pi^*} - \tilde{J}^{\pi^o}| \leq |\tilde{J}^{\pi^*} - \bar{J}^{\pi^*}| + |\bar{J}^{\pi^o} - \tilde{J}^{\pi^o}| = O(\epsilon^2)
$$

A similar argument holds for the variance as well.

Unfortunately, there is no standard solution to the closed loop problem (24) due to the non additive nature of the cost function $\mathrm{Var}(\delta J_1^\pi)$. Therefore, we solve a standard LQR problem as a surrogate, and the effect is again one of reducing the variance of the cost-to-go by reducing the variance of the closed loop trajectories.

Approximate Closed Loop Problem. We solve the following LQR problem for suitably defined cost function weighting factors Q_t, R_t:

$$
\begin{aligned}
\min_{(\delta u_t)_{t=1}^{T}} \ & \mathbb{E}[\sum_{t=1}^{T-1} \delta x_t' Q_t \delta x_t + \delta u_t' R_t \delta u_t + \delta x_T' Q_T \delta x_t], \\
& \delta x_{t+1} = A_t \delta x_t + B_t \delta u_t + \epsilon w_t.
\end{aligned} \tag{26}
$$

The solution to the above problem furnishes us a feedback gan $\hat{K}_t^*$ which we can use in the place of the true variance minimizing gain K_t^*.

Remark 4. Proposition 1 states that the expected cost-to-go of the problem is dominated by the nominal cost-to-go. Therefore, even an open loop policy consisting of simply the nominal control action is within $O(\epsilon^2)$ of the optimal expected cost-to-go. However, the plan with the optimal feedback gain K_t^* is strictly better than the open loop plan in that it has a lower variance in terms of the cost to go. Furthermore, solving the approximate closed loop problem using the surrogate LQR problem, we can expect a lower variance of the cost-to-go function as well.

7.3 An $O(\epsilon^4)$ Near-Optimal Decoupled Approach for Closed Loop Control

In order to derive the results in this section, we need some additional structure on the dynamics. *In essence, the results in this section require that the time discretization of the dynamics be small enough.* Thus, let the dynamics be given by:

$$x_t = x_{t-1} + \bar{f}(x_{t-1})\Delta t + \bar{g}(x_{t-1})u_t\Delta t + \epsilon\omega_t\sqrt{\Delta t}, \tag{27}$$

where ω_t is a white noise sequence, and the sampling time Δt is small enough that $O(\Delta t^\alpha)$ is negligible for $\alpha > 1$. The noise term above is a Brownian motion, and hence the $\sqrt{\Delta t}$ factor. Further, the incremental cost function $c(x, u)$ is given as: $c(x, u) = \bar{l}(x)\Delta t + \frac{1}{2}u'\bar{R}u\Delta t$. The main reason to use the above assumptions is to simplify the Dynamic Programming (DP) equation governing the optimal cost-to-go function of the system. The DP equation for the above system is given by:

$$J_t(x) = \min_{u_t}\{c(x, u) + E[J_{t+1}(x')]\}, \tag{28}$$

where $x' = x + \bar{f}(x)\Delta t + \bar{g}(x)u_t\Delta t + \epsilon\omega_t\sqrt{\Delta t}$ and $J_t(x)$ denotes the cost-to-go of the system given that it is at state x at time t. The above equation is marched back in time with terminal condition $J_T(x) = c_T(x)$, and $c_T(.)$ is the terminal cost function. Let $u_t(.)$ denote the corresponding optimal policy.Then, it follows that the optimal control u_t satisfies (since the argument to be minimized is quadratic in u_t)

$$u_t = -R^{-1}\bar{g}'J_{t+1}^x, \tag{29}$$

where $J_{t+1}^x = \frac{\partial J_{t+1}}{\partial x}$. Further, let $u_t^d(.)$ be the optimal control policy for the deterministic system, i.e., Eq. 27 with $\epsilon = 0$. The optimal cost-to-go of the deterministic system, $\phi_t(.)$ satisfies the deterministic DP equation:

$$\phi_t(x) = \min_u[c(x, u) + \phi_{t+1}(x')], \tag{30}$$

where $x' = x + \bar{f}(x')\Delta t + \bar{g}(x')u\Delta t$. Then, identical to the stochastic case, $u_t^d = R^{-1}\bar{g}'\phi_t^x$. Next, let $\varphi_t(.)$ denote the cost-to-go of the deterministic policy when applied to the stochastic system, i.e., u_t^d applied to Eq. 27 with $\epsilon > 0$. The cost-to-go $\varphi_t(.)$ satisfies the policy evaluation equation:

$$\varphi_t(x) = c(x, u_t^d(x)) + E[\varphi_{t+1}(x')], \tag{31}$$

where now $x' = x + \bar{f}(x)\Delta t + \bar{g}(x)u_t^d(x)\Delta t + \epsilon\omega_t\sqrt{\Delta t}$. Note the difference between the equations 30 and 31. Then, we have the following important result.

Proposition 4. *The difference between the cost function of the optimal stochastic policy, J_t, and the cost function of the "deterministic policy applied to the stochastic system", φ_t, is $O(\epsilon^4)$, i.e. $|J_t(x) - \varphi_t(x)| = O(\epsilon^4)$ for all (t, x).*

The above result was originally proved in a seminal paper [8] for continuous time, first passage problems. We have provided a simple derivation of the result, in the context of a discrete time finite horizon problem below.

Proof. Using Proposition 1, we know that any cost function, and hence, the optimal cost-to-go function can be expanded as:

$$J_t(x) = J_t^0 + \epsilon^2 J_t^1 + \epsilon^4 J_t^2 + \cdots \tag{32}$$

Thus, substituting the minimizing control in Eq. 29 into the dynamic programming Eq. 46 implies:

$$\begin{aligned} J_t(x) = \bar{l}(x)\Delta t + \frac{1}{2}r(\frac{-\bar{g}}{r})^2(J_{t+1}^x)^2\Delta t + J_{t+1}^x\bar{f}(x)\Delta t \\ +\bar{g}(\frac{-\bar{g}}{r})(J_{t+1}^x)^2\Delta t + \frac{\epsilon^2}{2}J_{t+1}^{xx}\Delta t + J_{t+1}(x), \end{aligned} \tag{33}$$

where J_t^x, and J_t^{xx} denote the first and second derivatives of the cost-to go function. Substituting Eq. 32 into eq. 33 we obtain that:

$$\begin{aligned} (J_t^0 + \epsilon^2 J_t^1 + \epsilon^4 J_t^2 + \cdots) = \bar{l}(x)\Delta t + \\ \frac{1}{2}\frac{\bar{g}^2}{r}(J_{t+1}^{0,x} + \epsilon^2 J_{t+1}^{1,x} + \cdots)^2\Delta t \\ +(J_{t+1}^{0,x} + \epsilon^2 J_{t+1}^{1,x} + \cdots)\bar{f}(x)\Delta t \\ -\frac{\bar{g}^2}{r}(J_{t+1}^{0,x} + \epsilon^2 J_{t+1}^{1,x} + \cdots)^2\Delta t \\ +\frac{\epsilon^2}{2}(J_{t+1}^{0,x} + \epsilon^2 J_{t+1}^{1,x} + \cdots)\Delta t + J_{t+1}(x). \end{aligned} \tag{34}$$

Now, we equate the ϵ^0, ϵ^2 terms on both sides to obtain perturbation equations for the cost functions $J_t^0, J_t^1, J_t^2 \cdots$.
First, let us consider the ϵ^0 term. Utilizing Eq. 34 above, we obtain:

$$J_t^0 = \bar{l}\Delta t + \frac{1}{2}\frac{\bar{g}^2}{r}(J_{t+1}^{0,x})^2\Delta t + \underbrace{(\bar{f} + \bar{g}\frac{-\bar{g}}{r}J_t^{0,x})}_{\bar{f}^0}J_t^{0,x}\Delta t + J_{t+1}^0, \tag{35}$$

with the terminal condition $J_T^0 = c_T$, and where we have dropped the explicit reference to the argument of the functions x for convenience.
Similarly, one obtains by equating the $O(\epsilon^2)$ terms in Eq. 34 that:

$$J_t^1 = \frac{1}{2}\frac{\bar{g}^2}{r}(2J_{t+1}^{0,x}J_{t+1}^{1,x})\Delta t + J_{t+1}^{1,x}\bar{f}\Delta t - \frac{\bar{g}^2}{r}(2J_{t+1}^{0,x}J_{t+1}^{1,x})\Delta t + \frac{1}{2}J_{t+1}^{0,xx}\Delta t + J_{t+1}^1,$$

which after regrouping the terms yields:

$$J_t^1 = \underbrace{(\bar{f} + \bar{g}\frac{-\bar{g}}{r}J_{t+1}^{0,x})J_{t+1}^{1,x}}_{=\bar{f}^0}\Delta t + \frac{1}{2}J_{t+1}^{0,xx}\Delta t + J_{t+1}^1, \tag{36}$$

with terminal boundary condition $J_T^1 = 0$. Note the perturbation structure of Eqs. 35 and 36, J_t^0 can be solved without knowledge of J_t^1, J_t^2 etc, while J_t^1 requires knowledge only of J_t^0, and so on. In other words, the equations can be solved sequentially rather than simultaneously.

Now, let us consider the deterministic policy $u_t^d(.)$ that is a result of solving the deterministic DP equation:

$$\phi_t(x) = \min_u[c(x,u) + \phi_{t+1}(x')], \tag{37}$$

where $x' = x + \bar{f}\Delta t + \bar{g}u\Delta t$, i.e., the deterministic system obtained by setting $\epsilon = 0$ in Eq. 27, and ϕ_t represents the optimal cost-to-go of the deterministic system. Analogous to the stochastic case, $u_t^d = \frac{-\bar{g}}{r}\phi_t^x$. Next, let φ_t denote the cost-to-go of the deterministic policy $u_t^d(.)$ *when applied to the stochastic system, i.e., Eq. 27 with* $\epsilon > 0$. Then, the cost-to-go of the deterministic policy, when applied to the stochastic system, satisfies:

$$\varphi_t = c(x, u_t^d(x)) + E[\varphi_{t+1}(x')], \tag{38}$$

where $x' = \bar{f}\Delta t + \bar{g}u_t^d\Delta t + \epsilon\sqrt{\Delta t}\omega_t$. Substituting $u_t^d(.) = \frac{-\bar{g}}{r}\phi_t^x$ into the equation above implies that:

$$\begin{aligned}
\varphi_t &= \varphi_t^0 + \epsilon^2\varphi_t^1 + \epsilon^4\varphi_t^2 + \cdots \\
&= \bar{l}\Delta t + \frac{1}{2}\frac{\bar{g}^2}{r}(\phi_{t+1}^x)^2\Delta t + (\varphi_{t+1}^{0,x} + \epsilon^2\varphi_{t+1}^{1,x} + \cdots)\bar{f}\Delta t \\
&\quad + \bar{g}\frac{-\bar{g}}{r}\phi_{t+1}^x(\varphi_{t+1}^{0,x} + \epsilon^2\varphi_{t+1}^{1,x} + \cdots)\Delta t \\
&\quad + \frac{\epsilon^2}{2}(\varphi_{t+1}^{0,xx} + \epsilon^2\varphi_{t+1}^{1,xx} + \cdots)\Delta t \\
&\quad + (\varphi_{t+1}^0 + \epsilon^2\varphi_{t+1}^1 + \cdots).
\end{aligned} \tag{39}$$

As before, if we gather the terms for ϵ^0, ϵ^2 etc. on both sides of the above equation, we shall get the equations governing φ_t^0, φ_t^1 etc. First, looking at the ϵ^0 term in Eq. 36, we obtain:

$$\varphi_t^0 = \bar{l}\Delta t + \frac{1}{2}\frac{\bar{g}^2}{r}(\phi_{t+1}^x)^2\Delta t + (\bar{f} + \bar{g}\frac{-\bar{g}}{r}\phi_{t+1}^x)\varphi_{t+1}^{0,x}\Delta t + \varphi_{t+1}^0, \tag{40}$$

with the terminal boundary condition $\varphi_T^0 = c_T$. However, the deterministic cost-to-go function also satisfies:

$$\phi_t = \bar{l}\Delta t + \frac{1}{2}\frac{\bar{g}^2}{r}(\phi_{t+1}^x)^2\Delta t + (\bar{f} + \bar{g}\frac{-\bar{g}}{r}\phi_{t+1}^x)\phi_{t+1}^x\Delta t + \phi_{t+1}, \tag{41}$$

with terminal boundary condition $\phi_T = c_T$. Comparing Eqs. 40 and 41, it follows that $\phi_t = \varphi_t^0$ for all t. Further, comparing them to Eq. 35, it follows that $\varphi_t^0 = J_t^0$, for all t. Also, note that the closed loop system above, $\bar{f} + \bar{g}\frac{-\bar{g}}{r}\phi_{t+1}^x = \bar{f}^0$ (see Eq. 35 and 36).

Next let us consider the ϵ^2 terms in Eq. 39. We obtain:

$$\varphi_t^1 = \bar{f}\varphi_{t+1}^{1,x}\Delta t + \bar{g}\frac{-\bar{g}}{r}\phi_{t+1}^x\varphi_{t+1}^{1,x}\Delta t + \frac{1}{2}\varphi_{t+1}^{0,xx} + \varphi_{t+1}^1.$$

Noting that $\phi_t = \varphi_t^0$, implies that (after collecting terms):

$$\varphi_t^1 = \bar{f}^0\varphi_{t+1}^{1,x}\Delta t + \frac{1}{2}\varphi_{t+1}^{0,xx}\Delta t + \varphi_{t+1}^1, \tag{42}$$

with terminal boundary condition $\varphi_N^1 = 0$. Again, comparing Eq. 42 to Eq. 36, and noting that $\varphi_t^0 = J_t^0$, it follows that $\varphi_t^1 = J_t^1$, for all t. This completes the proof of the result.

Given some initial condition x_0, consider a linear truncation of the optimal deterministic policy, i.e., let $u_t^l(.) = \bar{u}_t + K_t\delta x_t$, where the deterministic policy is given by $u_t^d = \bar{u}_t + K_t\delta x_t + S_t(\delta x_t)$, where $S_t(.)$ denote the second and higher order terms in the optimal deterministic feedback policy. Using Proposition 2, it follows that the cost of the linear policy, say $\varphi_t^l(.)$, is within $O(\epsilon^4)$ of the cost of the deterministic policy $u_t^d(.)$, when applied to the stochastic system in Eq. 27. However, the result in Proposition 4 shows that the cost of the deterministic policy is within $O(\epsilon^4)$ of the optimal stochastic policy. Taken together, this implies that the cost of the linear deterministic policy is within $O(\epsilon^4)$ of the optimal stochastic policy. This may be summarized in the following result.

Proposition 5. *Let the optimal cost function under the true stochastic policy be given $J_t(.)$ Let the optimal deterministic policy be given by $u_t^d(x_t) = \bar{u}_t + K_t\delta x_t + S_t(\delta x_t)$, and the linear approximation to the policy be $u_t^l(x_t) = \bar{u}_t + K_t\delta x_t$, and let the cost of the linear policy be given by $\varphi_t^l(x)$. Then $|J_t(x) - \varphi_t^l(x)| = O(\epsilon^4)$ for all (t, x).*

Now, it remains to be seen how to design the $\bar{u}_t$ and the linear feedback term K_t. The open loop optimal control sequence $\bar{u}_t$ is found identically to the previous section. However, the linear feedback gain K_t is calculated in a slightly different fashion and may be done as shown in the following result. In the following, $\mathcal{F}(x) = x + \bar{f}(x)\Delta t$, $\mathcal{G}(x) = \bar{g}(x)\Delta t$, $A_t = \frac{\partial \mathcal{F}}{\partial x}|_{\bar{x}_t} + \frac{\partial \mathcal{G}\bar{u}_t}{\partial x}|_{\bar{x}_t}$, $B_t = \mathcal{G}(\bar{x}_t)$, $L_t = \frac{\partial l}{\partial x}|'_{\bar{x}_t}$ and $L_{tt} = \nabla^2_{xx} l|_{\bar{x}_t}$. Let $\phi_t(x_t)$ denote the optimal cost-to-go of the detrministic problem, i.e., Eq 27 with $\epsilon = 0$.

Proposition 6. ***Decoupled Design.*** *Given an optimal nominal trajectory $(\bar{x}_t, \bar{u}_t)$, the backward evolutions of the first and second derivatives, $G_t = \frac{\partial \phi_t}{\partial x}|'_{\bar{x}_t}$ and $P_t = \nabla^2_{xx}\phi_t|_{\bar{x}_t}$, of the optimal cost-to-go function $\phi_t(x_t)$, initiated with the terminal boundary conditions $G_N = \frac{\partial c_N(x_N)}{\partial x_N}|'_{\bar{x}_N}$ and $P_N = \nabla^2_x c_N|_{\bar{x}_N}$ respectively,*

are as follows:

$$G_t = L_t + G_{t+1}A_t, \tag{43}$$

$$P_t = L_{tt} + A_t' P_{t+1} A_t - K_t' S_t K_t + G_{t+1} \otimes \tilde{R}_{t,xx}, \tag{44}$$

for $t = \{0, 1, ..., N-1\}$*, where,* $S_t = (R_t + B_t' P_{t+1} B_t), K_t = -S_t^{-1}(B_t' P_{t+1} A_t + (G_{t+1} \otimes \tilde{R}_{t_{xu}})'), \tilde{R}_{t,xx} = \nabla^2_{xx}\mathcal{F}(x_t)|_{\bar{x}_t} + \nabla^2_{xx}\mathcal{G}(x_t)|_{\bar{x}_t,\bar{u}_t}, \tilde{R}_{t,xu} = \nabla^2_{xu}(\mathcal{F}(x_t) + \mathcal{G}(x_t)u_t)|_{\bar{x}_t,\bar{u}_t}$ *where* ∇^2_{xx} *represents the Hessian of a vector-valued function w.r.t* x *and* $\otimes$ *denotes the tensor product.*

Proof. Consider the Dynamic Programming equation for the deterministic cost-to-go function:

$$\phi_t(x_t) = \min_{u_t} Q_t(x_t, u_t) = \min_{u_t}\{c_t(x_t, u_t) + \phi_{t+1}(x_{t+1})\}$$

By Taylor's expansion about the nominal state at time $t+1$,

$$\begin{aligned}\phi_{t+1}(x_{t+1}) =& \phi_{t+1}(\bar{x}_{t+1}) + G_{t+1}\delta x_{t+1} \\ &+ \frac{1}{2}\delta {x_{t+1}}' P_{t+1}\delta x_{t+1} + q_{t+1}(\delta x_{t+1}).\end{aligned}$$

Substituting the linearization of the dynamics, $\delta x_{t+1} = A_t\delta x_t + B_t\delta u_t + r_t(\delta x_t, \delta u_t)$ in the above expansion,

$$\begin{aligned}&\phi_{t+1}(x_{t+1}) = \phi_{t+1}(\bar{x}_{t+1}) + G_{t+1}(A_t\delta x_t + B_t\delta u_t + r_t(\delta x_t \\ &, \delta u_t)) + (A_t\delta x_t + B_t\delta u_t + r_t(\delta x_t, \delta u_t))' P_{t+1}(A_t\delta x_t \\ &+ B_t\delta u_t + r_t(\delta x_t, \delta u_t)) + q_{t+1}(\delta x_{t+1}).\end{aligned}$$

Similarly, expand the incremental cost at time t about the nominal state,

$$\begin{aligned}c_t(x_t, u_t) = \bar{l}_t + L_t\delta x_t + \frac{1}{2}\delta {x_t}' L_{tt}\delta x_t + \frac{1}{2}\delta {u_t}' R_t\bar{u}_t \\ + \frac{1}{2}\bar{u}_t' R_t\delta u_t + \frac{1}{2}\delta {u_t}' R_t\delta u_t + \frac{1}{2}\bar{u}_t' R_t\bar{u}_t + s_t(\delta x_t).\end{aligned}$$

$$\begin{aligned}Q_t(x_t, u_t) =& \overbrace{[\bar{l}_t + \frac{1}{2}\bar{u}_t^\intercal R_t\bar{u}_t + \phi_{t+1}(\bar{x}_{t+1})]}^{\bar{\phi}_t(\bar{x}_t,\bar{u}_t)} \\ &+ \delta {u_t}'(B_t'\frac{P_{t+1}}{2}B_t + \frac{1}{2}R_t)\delta u_t + \delta {u_t}'(B_t'\frac{P_{t+1}}{2}A_t\delta x_t \\ &+ \frac{1}{2}R_t\bar{u}_t + B_t'\frac{P_{t+1}}{2}r_t) + (\delta {x_t}' A_t'\frac{P_{t+1}}{2}B_t + \frac{1}{2}\bar{u}_t R_t \\ &+ r_t'\frac{P_{t+1}}{2}B_t + G_{t+1}B_t)\delta u_t + \delta {x_t}' A_t'\frac{P_{t+1}}{2}A_t\delta x_t \\ &+ \delta {x_t}'\frac{P_{t+1}}{2}A_t' r_t + (r_t'\frac{P_{t+1}}{2}A_t + G_{t+1}A_t)\delta x_t \\ &+ r_t'\frac{P_{t+1}}{2}r_t + G_{t+1}r_t + q_t \equiv \bar{\phi}_t(\bar{x}_t, \bar{u}_t) + H_t(\delta x_t, \delta u_t).\end{aligned}$$

$$\text{Now}, \min_{u_t} Q_t(x_t, u_t) = \min_{\bar{u}_t} \bar{\phi}_t(\bar{x}_t, \bar{u}_t) + \min_{\delta u_t} H_t(\delta x_t, \delta u_t)$$

First order optimality: Along the optimal nominal control sequence $\bar{u}_t$, it follows from the minimum principle that

$$\frac{\partial c_t(x_t, u_t)}{\partial u_t} + \frac{\partial g(x_t)}{\partial u_t}' \frac{\partial \phi_{t+1}(x_{t+1})}{\partial x_{t+1}} = 0$$

$$\Rightarrow R_t \bar{u}_t + B_t' G_{t+1}' = 0 \tag{45}$$

By setting $\frac{\partial H_t(\delta x_t, \delta u_t)}{\partial \delta u_t} = 0$, we get:

$$\begin{aligned}
\delta u_t^* &= -S_t^{-1}(R_t \bar{u}_t + B_t' G_{t+1}') - S_t^{-1}(B_t' P_{t+1} A_t + \\
&\quad (G_t \otimes \tilde{R}_{t,xu})')\delta x_t - S_t^{-1}(B_t' P_{t+1} r_t) \\
&= \underbrace{-S_t^{-1}(B_t' P_{t+1} A_t + (G_{t+1} \otimes \tilde{R}_{t,xu})')}_{K_t} \delta x_t \\
&\quad + \underbrace{S_t^{-1}(-B_t' P_{t+1} r_t)}_{p_t}
\end{aligned}$$

where, $S_t = R_t + B_t' P_{t+1} B_t$.

$$\Rightarrow \delta u_t = K_t \delta x_t + p_t.$$

Substituting it in the expansion of J_t and regrouping the terms based on the order of δx_t (till 2^{nd} order), we obtain:

$$\begin{aligned}
\phi_t(x_t) = \bar{\phi}_t(\bar{x}_t) + (L_t + (R_t \bar{u}_t + B_t' G_{t+1}')K_t + G_{t+1} A_t)\delta x_t \\
+ \frac{1}{2}\delta x_t{}'(L_{tt} + A_t' P_{t+1} A_t - K_t' S_t K_t + G_{t+1} \otimes \tilde{R}_{t,xx})\delta x_t.
\end{aligned}$$

Expanding the LHS about the optimal nominal state result in the recursive equations in Proposition 6.

7.4 Summary of the Decoupling Results and Implications

The previous two subsections showed that the feedback parameterization can be written as: $\pi_t(x_t) = \bar{u}_t + K_t \delta x_t$, where $\delta x_t = x_t - \bar{x}_t$ denotes the state deviation from the nominal. Further, it was shown that the optimal open loop sequence $\bar{u}_t$ is independent of the feedback gain, while the feedback gain K_t can be designed based on the optimal $\bar{u}_t$. *Hence, the term decoupling, in the sense that the search for the optimal parameter* $(\bar{u}_t^*, K_t^*)$ *need not be done jointly.*
Moreover, it was shown that depending on how one designed the gain K_t, we can obtain either $O(\epsilon^2)$ (Proposition 3), or $O(\epsilon^4)$ (Propositions 6), near-optimality to the true stochastic policy.

8 Analysis of the High Noise Regime

In this section, we perform a rudimentary analysis of the high noise regime. The medium noise case is more difficult to analyze and is left for future work, along with a more sophisticated treatment of the high noise regime.
First, recall the Dynamic Programming (DP) equation for the backward pass to determine the optimal time varying feedback policy:

$$J_t(\mathbf{x}_t) = \min_{\mathbf{u}_t} \left\{ c(\mathbf{x}_t, \mathbf{u}_t) + \mathbb{E}\left[J_{t+1}(\mathbf{x}_{t+1})\right] \right\}, \tag{46}$$

where $J_t(\mathbf{x}_t)$ denotes the cost-to-go at time t given the state is $\mathbf{x}_t$, with the terminal condition $J_T(\cdot) = c_T(\cdot)$ where c_T is the terminal cost function, and the next state $\mathbf{x}_{t+1} = f(\mathbf{x}_t) + \mathbf{B}_t(\mathbf{u}_t + \epsilon\mathbf{w}_t)$. Suppose now that the noise is so high that $\mathbf{x}_{t+1} \approx \mathbf{B}_t\epsilon\mathbf{w}_t$, i.e., the dynamics are completely swamped by the noise.
Consider now the expectation $\mathbb{E}\left[c_T(\mathbf{x}_{t+1})\right]$ given some control $\mathbf{u}_t$ was taken at state $\mathbf{x}_t$. Since $\mathbf{x}_{t+1}$ is determined entirely by the noise, $\mathbb{E}\left[c_T(\mathbf{x}_{t+1})\right] = \int c_T(\mathbf{B}_t\epsilon\mathbf{w}_t)\mathbf{p}(\mathbf{w}_t)d\mathbf{w}_t = \overline{c_T}$, where $\overline{c_T}$ is a constant regardless of the previous state and control pair $\mathbf{x}_t, \mathbf{u}_t$. This observation holds regardless of the function $c_T(\cdot)$ and the time t.
Next, consider the DP iteration at time $T-1$. Via the argument above, it follows that $\mathbb{E}\left[J_T(\mathbf{x}_T)\right] = \mathbb{E}\left[c_T(\mathbf{x}_T)\right] = \overline{c_T}$, regardless of the state control pair $\mathbf{x}_{T-1}, \mathbf{u}_{T-1}$ at the $(T-1)^{th}$ step, and thus, the minimization reduces to $J_{T-1}(\mathbf{x}_{T-1}) = \min_{\mathbf{u}} \{c(\mathbf{x}_{T-1}, \mathbf{u}) + \overline{c_T}\}$, and thus, the minimizer is just the greedy action $\mathbf{u}^*_{T-1} = \arg\min_{\mathbf{u}} c(\mathbf{x}_{T-1}, \mathbf{u})$ due to the constant bias $\overline{c_T}$. The same argument holds for any t since, although there might be a different $J_t(\cdot)$ at every time t, the minimizer is still the greedy action that minimizes $c(\mathbf{x}_t, \mathbf{u})$ as the cost-to-go from the next state is averaged out to simply some $\bar{J}_{t+1}$.

9 Additional Simulation Results:

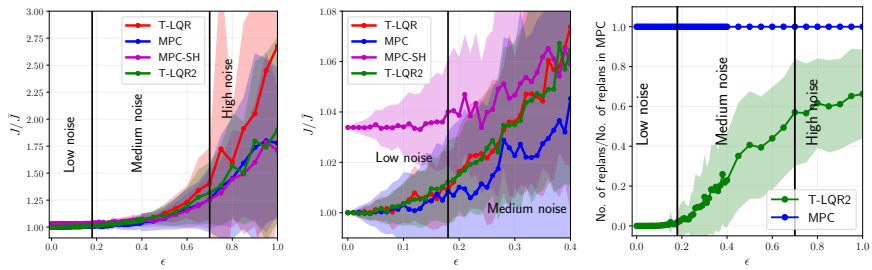
In addition to the simulations performed on the car-like robot shown in the main article, experiments are performed on a car with trailers and a quadrotor whose results are shown in Figure 6 and 7 respectively. We also show a scenario on a car-like robot in an environment with obstacles to illustrate that the decoupling approach can handle such cases. The parameters used in the simulations are given in Table 1 and 2. As seen in car-like robot, the performance of T-LQR2 is close to MPC for a wide range of noise levels. It is also evident from the replanning operations plots in Figure 6c and 7c that T-LQR2 is computationally efficient when compared to MPC. MPC-SH also exhibits the similar trends as shown in the main article.

9.1 Car-like robot with trailers:

Having trailers in a car-like robot makes it more complex by increasing the state dimension of it by the number of trailers attached. Here we consider 2 trailers whose heading angles are given by,

$$\theta_1(t+1) = \theta_1(t) + \frac{v_t}{L} sin(\theta(t) - \theta_1(t))\Delta t,$$
$$\theta_2(t+1) = \theta_2(t) + \frac{v_t}{L} cos(\theta(t) - \theta_1(t)) sin(\theta_1(t) - \theta_2(t))\Delta t.$$

The performance is shown in Figure 6. As seen in the car-like robot, T-LQR is near-optimal in the low noise regime, while T-LQR2 performs similar to MPC in the medium and high noise regime. In the high noise regime, as seen earlier, MPC-SH achieves similar performance to MPC and T-LQR2 despite planning only for a short horizon.



(a) Full noise spectrum (b) Enhanced detail $0 \leq \epsilon \leq 0.4$ (c) Replanning operations

Fig. 6: Cost evolution of the different algorithms for a car with 2 trailer system.

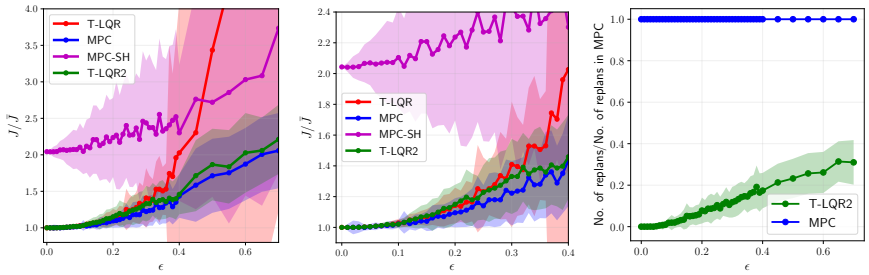
9.2 Quadrotor:

To evaluate in a 3D setting, we consider a quadrotor whose 12D state vector comprises of its position, orientation, linear and angular velocities - $(\mathbf{x}_t, \theta_t, \mathbf{v}_t, \omega_t)$. The model is described by

$$\dot{\mathbf{x}}_t = \mathbf{v}_t, \qquad \dot{\mathbf{v}}_t = \mathbf{g} + \frac{1}{m} R_{\theta_t} \mathbf{F}_t,$$
$$\dot{\theta}_t = \mathbf{W}_\theta^{-1} \omega_t, \qquad \dot{\omega}_t = \mathbf{I}^{-1} \tau_t$$

where, $\mathbf{W}_\theta$ is the transformation from the inertial to body frame and $\mathbf{I}$ is the inertia matrix. The model has thrust $(\mathbf{F}_t)$ and torques $(\tau_\mathbf{t})$ in its body fixed frame as the 4 control inputs. The results are shown in Figure 7. Unlike a mobile robot which is stable even in high noise cases, a quadrotor is susceptible to

failure or reach states from which no form of control can help it recover. So, the performance degrades earlier compared to the other two systems. But it can still be observed that T-LQR2 performs on a par with MPC in spite of the former replanning less than half the number of times compared to the latter.



(a) Full noise spectrum (b) Enhanced detail $0 \leq \epsilon \leq 0.4$ (c) Replanning operations

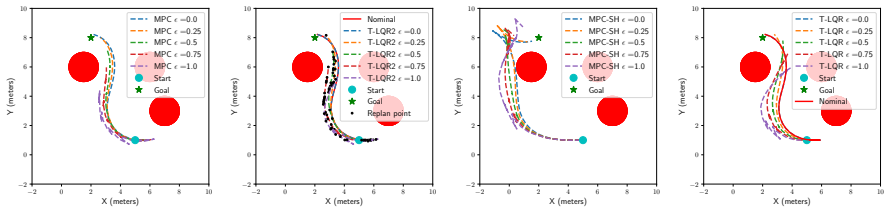
Fig. 7: Cost evolution of the different algorithms for a Quadrotor.

9.3 Car-like robot in the presence of obstacles:

We show a case where the problem involves static obstacles in the environment. We assume the robot knows the map of the environment. The obstacles can be defined as ellipsoids. The ellipsoids can be represented with center $\mathbf{o}^k \in \mathcal{R}^2$ and a positive definite matrix $\mathbf{E}^k \in \mathcal{R}^{2\times 2}$. The obstacle penalty function for an agent whose position is $\mathbf{p}_t$ in an environment with n obstacles is

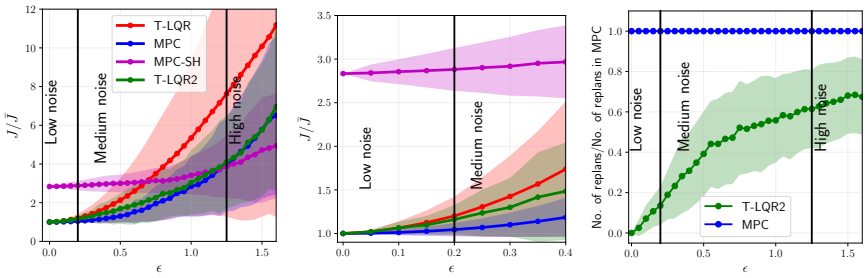
$$\Phi = \mathrm{M} \sum_{k=1}^{n} \exp(-[(\mathbf{p}_t - \mathbf{o}^k)^T \mathbf{E}^k (\mathbf{p}_t - \mathbf{o}^k) - 1]),$$

where M is a scaling factor.



(a) MPC (b) T-LQR2 (c) MPC-SH (d) T-LQR

Fig. 8: The figure shows the paths taken by the robot using a particular algorithm and how they change as ϵ varies. A difference we see here is the paths taken by MPC-SH when compared to the others. Since it plans for a short horizon and hence greedy, it takes a different path unlike the others.



(a) Full noise spectrum (b) Enhanced detail $0 \leq \epsilon \leq 0.4$ (c) Replanning operations

Fig. 9: Cost evolution of the different algorithms for a car-like robot in an environment with obstacles.

	Car-like	Car with trailers	Quadrotor
$\mathbf{x}_0$	$[3, 1, 0, 0]^T$	$[0, 0, \pi/3, 0, 0, 0]^T$	$[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$
$\mathbf{x}_f$	$[3.5, 7, m.pi/2, 0]^T$	$[2, 2, 0, 0, 0, 0]^T$	$[2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$
T, Δt	$35, 0.1$	$40, 0.1$	$30, 0.1$
$\mathbf{W}^x$	$diag(20, 20, 0, 0)$	$diag(10, 10, 1, 1, 1, 1)$	$diag(10, 10, 10, 1, 1, 1, 1, 1, 1, 1, 1, 1)$
$\mathbf{W}^u$	$diag(20, 200)$	$diag(5, 5)$	$diag(5, 10, 10, 10)$
$\mathbf{W}_f^x$	$10^3 diag(7, 7, 10, 1)$	$10^3 diag(1, 1, 1, 0.1, 0.1, 0.1)$	$10^3 diag(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$
Control bounds	$v_t = [-4, 4]$, $\omega_t = [-\pi/12, \pi/12]$	$v_t = [-0.8, 0.8]$, $\omega_t = [-\pi/6, \pi/6]$	$u_t^{(1)} = [0, 1.5]$, $u_t^{(i)} = [-0.05, 0.05]$ $i = 2, 3, 4$

Table 1: Parameters used in the single agent simulations.

	Car-like
$\mathbf{x}_0$	Agent 1: $[3, 1, \pi/2, 0]$; Agent 2: $[5, 1, 0, 0]$; Agent 3: $[6, 8, 0, 0]$
$\mathbf{x}_f$	Agent 1: $[3.5, 7, 0, 0]$; Agent 2: $[2, 8, 0, 0]$; Agent 3: $[8, 1.5, 0, 0]$
T, Δt	35, 0.1
$\mathbf{W}^x$	$diag(20, 20, 0, 0)$
$\mathbf{W}^u$	$diag(20, 200)$
$\mathbf{W}_f^x$	$10^3 diag(7, 7, 10, 1)$
Control bounds	$v_t = [-4, 4]$, $\omega_t = [-\pi/12, \pi/12]$

Table 2: Parameters used in the multi-agent simulations.