

A Theory of Dynamics, Control and Optimization in Layered Architectures

Submitted to the 2016 IEEE American Control Conference (ACC 2016)

Nikolai Matni and John C. Doyle *

Abstract

The controller of a large-scale distributed system (e.g., the internet, the power-grid and automated highway systems) is often faced with two complementary tasks: (i) that of finding an optimal trajectory with respect to a functional or economic utility, and (ii) that of efficiently making the state of the system follow this trajectory despite model uncertainty, process and sensor noise and distributed information sharing constraints. While each of these tasks has been addressed individually, there exists as of yet no controller synthesis framework that treats these two problems in a holistic manner. This paper proposes a unifying optimization based methodology that jointly addresses these two tasks by leveraging the strengths of well established frameworks for distributed control: the Layering as Optimization (LAO) framework and the distributed optimal control framework. We show that our proposed control scheme has a natural layered architecture composed of a low-level tracking layer and top-level planning layer. The tracking layer consists of a distributed optimal controller that takes as an input a reference trajectory generated by the top-level layer, where this top-level layer consists of a trajectory planning problem that optimizes a weighted sum of a utility function and a “tracking penalty” regularizer. We further provide an exact solution to the tracking layer problem under a broad range of information sharing constraints, discuss extensions to the proposed problem formulation, and demonstrate the effectiveness of our approach on a numerical example.

1 Introduction

Distributed systems such as software defined networks, power-grids and the human sensorimotor control system present a unique challenge to the control engineer. When controlling or reverse engineering these often large-scale systems, we are faced with two complementary tasks: that of finding an optimal trajectory (or set-point) with respect to a functional or economic utility, and that of efficiently making the state of the system follow this optimal trajectory (or set-point) despite model uncertainty, process and sensor noise and information sharing constraints.

Layering has proven to be a powerful architectural approach [1] to designing such large-scale control systems – indeed, layered architectures are ubiquitous in the internet [2, 3], modern approaches to power-grid control [4, 5], and biological systems [6, 7]. In the context of engineering systems, the layering as optimization (LAO) (cf. [2, 3] and references therein) and the reverse/forward engineering (cf. [4, 5] and references therein) paradigm have been particularly fruitful in tackling internet

*N. Matni & J. C. Doyle are with the Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA. {nmatni,doyle}@caltech.edu. This research was in part supported by NSF NetSE, AFOSR, the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office, and from MURIs “Scalable, Data-Driven, and Provably-Correct Analysis of Networks” (ONR) and “Tools for the Analysis and Design of Complex Multi-Scale Networks” (ARO). The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

and power-grid control problems, respectively. Both of these frameworks can be loosely viewed as using the dynamics of the system to implement a distributed optimization algorithm, ensuring that the state of the system converges to a set-point that optimizes a utility function. These approaches can scale to large systems by taking advantage of the structure underlying the utility optimization problem, and can simultaneously identify and guarantee stability around an optimal equilibrium point.

A complementary and somewhat orthogonal approach to the control of large scale systems is that considered in the distributed optimal control literature [8,9]. In such problems, the assumption is that the optimal set-point has already been specified, and the objective is rather to bring the state of the system to this set-point and keep it there as efficiently as possible, despite disturbances and information sharing constraints between actuators, sensors and controllers. Such approaches are not necessarily scalable, although recent developments [10, 11] are aimed at addressing this limitation, but do fully utilize knowledge of the system dynamics and information sharing constraints to guarantee optimal transient behavior around a pre-specified equilibrium point. This area has seen an explosion of results in the past decade, making it impossible to provide a proper review of the literature in the space available. We instead point the reader to the recent survey paper [9] for an overview of the challenges related to distributed optimal controller synthesis, and later introduce necessary concepts and specific results as they are needed.

From this brief discussion, it is apparent that these two frameworks are complementary in their strengths and weaknesses. For instance, although the LAO framework is extremely flexible and scales seamlessly to large systems, it is not naturally able to accommodate process or sensor noise, nor does it penalize undesirable transient behaviors. On the other hand, the distributed optimal control framework naturally handles process and sensor noise and penalizes undesirable transient behaviors – however, it is very limited in the types of cost-functions that it can accommodate, and an optimal set-point must be pre-specified.

This complementarity in strengths and weaknesses is the motivation behind this paper. Our main contribution is a unifying optimization based methodology that naturally combines the advantages of the LAO and distributed optimal control frameworks, allowing for each of their strengths to be leveraged together in a principled way. In particular, we define a dynamic version of the utility optimization problems typically considered in the LAO setting, and show that via a suitable relaxation of the problem, a layered architecture naturally emerges. This architecture has two layers: a tracking layer and a planning layer (cf. Fig. 1). The low-level tracking layer consists of a distributed optimal controller that tracks a reference trajectory generated by the top-level planning layer, where this top layer consists of a trajectory planning problem that optimizes a weighted sum of a utility function and a “tracking penalty” regularizer. We then show how the tracking layer problem can be solved exactly under interesting information sharing constraints by leveraging existing results from the distributed optimal control literature.

This paper is organized as follows: we formulate a dynamic version of a utility maximization problem in §2 and show how various problems considered in the literature can be obtained as a special case. In §3, we show that through a suitable relaxation, a natural layered architecture arises, in which the bottom layer consists of a distributed optimal control problem, and the top layer consists of a “tracking regularized” trajectory planning problem. In §4, we give an explicit solution to our relaxed problem when the distributed optimal control problem satisfies certain information sharing constraints and is formulated with respect to an LQG-like cost function. In §5, we discuss several extensions to the basic problem formulated in §2 and §3 – due to space limitations we only briefly outline these extensions, but make sure to highlight what technical challenges, if any, remain to be solved. We end with a numerical example in §6.

Notation

For a sequence of vectors $x_t \in \mathbb{R}^n$, $t = 0, \dots, N$, we use $x_{0:N}$ to denote the signal $(x_t)_{t=0}^N$. Similarly, given a sequence of matrices M_t , $t = 0, \dots, N$, and a sequence of vectors x_t of compatible dimension, we use $M_{0:N} \cdot x_{0:N}$ to denote the signal $(M_t x_t)_{t=0}^N$. We use Z to denote the block upshift matrix, that is to say a block matrix with identity matrices along the first block super-diagonal, and zero elsewhere. Similarly, we use E_i to denote a block column matrix with the i th block set to identity, and all others set to 0. Finally, I_n denotes the $n \times n$ identity matrix. Unless required for the discussion, we do not explicitly denote dimensions and we assume that all vectors, operators and spaces are of compatible dimension throughout.

2 Problem Formulation

Consider linear dynamics described by

$$x_{t+1} = A_t x_t + B_t u_t + H_t w_t, \quad z_t = C_t x_t, \quad x_0 \text{ given} \quad (1)$$

where x_t is the state of the system at time t , z_t is the controlled output, w_t is the process noise driving the system, u_t is the control input, and x_0 is a known initial condition. As of yet, we only restrict any realization of the disturbance signal $w_{0:N}$ to be in ℓ_∞ .

We assume that the plant described by the dynamics (1) is distributed and composed of p subsystems. To each subsystem we assign a corresponding subset x_t^i , u_t^i , w_t^i and z_t^i of the signals described in (1), corresponding to the state, control action, process noise and controlled output at subsystem i . We assume that the matrices B_t , H_t and C_t are block-diagonal,¹ and that each A_t admits a compatible block-wise partition (A_t^{ij}) . We then have that the dynamics at subsystem i are described by

$$\begin{aligned} x_{t+1}^i &= A_t^{ii} x_t^i + \sum_{j: A^{ij} \neq 0} A_t^{ij} x_t^j + B_t^{ii} u_t^i + H_t^{ii} w_t, \\ z_t^i &= C_t^{ii} x_t^i. \end{aligned} \quad (2)$$

In this paper we consider control tasks with respect to non-traditional cost functions, as specified in optimization problem (5), subject to information sharing constraints. In order to impose such information sharing constraints, we define the information set \mathcal{I}_t^i available to a controller at node i at time t as

$$\mathcal{I}_t^i := \left\{ x_{0:t-\tau_{i1}}^1, x_{0:t-\tau_{i2}}^2, \dots, x_{0:t}^i, \dots, x_{0:t-\tau_{ip}}^p \right\}, \quad (3)$$

where τ_{ij} is the communication delay from subsystem j to subsystem i .² We then say that a control law u_t respects the information sharing constraints of the system if the control action u_t^i taken at time t by subsystem i is a function of the information set \mathcal{I}_t^i , i.e., if there exists some Borel measurable map γ_t^i such that

$$u_t^i = \gamma_t^i(\mathcal{I}_t^i) \quad (4)$$

for each subsystem i . We focus in particular on *partially nested* [12] information structures, which satisfy the important property that for every admissible control policy γ , whenever u_s^i affects \mathcal{I}_t^j , we then have that $\mathcal{I}_s^i \subseteq \mathcal{I}_t^j$. This property implies that a class of distributed optimal control problems admits a unique optimal policy that is linear in its information set: we build on this to prove our main technical result in §4.

¹These assumptions can be relaxed for some of the distributed controller synthesis methods referenced in Table 1.

²We define the information sets with a state-feedback setting in mind: the definition extends in a natural way to the output feedback setting by replacing the state x with the measured output y .

The control task that we are interested in solving is specified by the following optimization problem

$$\begin{aligned} & \underset{x_{0:N}, u_{0:N-1}}{\text{minimize}} \quad \mathcal{C}(z_{0:N}) + \|D_{0:N-1} \cdot u_{0:N-1}\|_w \\ & \text{s.t.} \quad \text{dynamics (1), distributed constraints (4), } z_{0:N} \in \mathcal{R} \end{aligned} \tag{5}$$

where $\mathcal{C}(\cdot)$ is a convex cost function, $D_{0:N}$ is a collection of matrices D_t satisfying $D_t^\top D_t \succ 0$, $\|\cdot\|_w$ is a suitable signal-to-signal penalty such as $\mathbb{E}_w \|\cdot\|_2^2$ (corresponding to an LQG like penalty if the disturbances w_t are taken to be jointly Gaussian), $\max_{\|w\|_{\ell_2} \leq 1} \|\cdot\|_2^2$ (corresponding to an \mathcal{H}_∞ like penalty) or $\max_{\|w\|_{\ell_\infty} \leq 1} \|\cdot\|_\infty$ (corresponding to an \mathcal{L}_1 like penalty), and \mathcal{R} defines a convex constraint set on the controlled output $z_{0:N}$.

The interpretation of this problem is straightforward: select a state trajectory $x_{0:N}$ that optimizes the cost function $\mathcal{C}(\cdot)$ and that respects the constraint $z_{0:N} \in \mathcal{R}$, that can be achieved with a reasonable control effort despite the dynamics and information sharing constraints of the system. As we describe in more detail in what follows, existing tools from distributed (layering as) optimization and distributed optimal control are able to address different special cases of this problem, but as of yet, no framework exists that satisfactorily combines the tools developed in each of these approaches. The goal of this paper is to propose such a framework, allowing for the flexibility and scalability of LAO to be combined with the desirable transient behavior achieved by distributed optimal controllers, thus the broadening collective scope and applicability of these approaches.

We begin by highlighting special cases of problem (5) that can be solved using existing controller synthesis frameworks.

No dynamics or control cost

If the dynamic constraint (1) and the control cost $\|D_{0:N} \cdot u_{0:N-1}\|_w$ are removed from optimization problem (5), it reduces to a static optimization problem in the variable $x_{0:N}$. The layering as optimization (LAO) framework (cf. [2,3] and references therein) and the forward/reverse engineering framework (cf. [4,5] and references therein) have proven to be powerful tools in controlling large scale systems by using system dynamics to solve such a static optimization problem in a scalable way. In particular, under suitable and somewhat idealized conditions (namely in the absence of noise), these methods provably converge to an optimum of the original optimization problem. The resulting control schemes should thus be viewed as *stabilizing* controllers around such an optimal equilibrium point. It is worth noting however that since the dynamics of the system are not explicitly considered in the optimization problem, this framework does not optimize the trajectory taken by the system to reach this optimal equilibrium point.

No state constraints and control theoretic cost function

If the output constraints $z_{0:N} \in \mathcal{R}$ are dropped from optimization problem (5), and the cost function $\mathcal{C}(\cdot)$ is taken to be a suitable control theoretic cost such as the LQG or \mathcal{H}_∞ cost function, then this reduces to a distributed optimal control problem [8,9,13]. If the information constraints are such that the resulting control problem is partially nested (alternatively quadratically invariant) [8,9,12,14], then one may solve for the resulting linear distributed optimal controller in many cases of interest (we highlight such cases in §4) by leveraging recent results from the distributed optimal control community [15–18]. A limitation of this approach is that it only applies to a few cost functions, all of which can only be applied to systems that have a pre-specified equilibrium point.

No information sharing constraints and no driving noise

If the information sharing constraints $u_t^i = \gamma_t^i(\mathcal{I}_t^i)$ are removed from optimization problem (5) and the driving noise $w_{0:N-1}$ is set to 0, the resulting optimization problem is identical to subproblems solved in the context of model predictive control (MPC) (cf. [19] and references therein). Although extensions of model predictive control to distributed settings do exist (cf. [20] and references therein), they are not as flexible in dealing with information sharing constraints as analogous results in the distributed optimal control literature, especially in the output feedback setting. However, a major advantage of MPC is its ability to naturally accommodate nonlinearities, and in particular actuator saturation – extending the applicability of our framework to nonlinear systems by incorporating it into the MPC scheme is thus an important avenue for future work, but will not be discussed here.

In light of the previous discussion, the conceptual challenge that needs to be overcome in solving optimization problem (5) is that we need to simultaneously identify an optimal trajectory and ensure that the system can effectively track this trajectory despite the system dynamics (1), the control cost $\|D_{0:N-1} \cdot u_{0:N-1}\|_w$, the information sharing constraints (4) of the controller and the driving noise of the system. In what follows, we propose a relaxation based approach inspired by the notion of vertical layering in the LAO framework to functionally separate the tasks of planning an optimal trajectory and efficiently tracking it. This separation allows for the tracking problem to be solved as a traditional distributed optimal control problem, independent of the planning problem: when the tracking problem admits an analytic solution, it follows that optimization problem (5) then reduces to a suitably modified planning problem in which the tracking cost acts as a *regularizer*. We make this intuition precise in the next section.

3 Vertical Layering with Dynamics

Our goal is to create a separation between the tasks of planning an optimal trajectory and synthesizing a controller that ensures that the state of the system efficiently tracks said trajectory. To that end, we propose the following relaxation to optimization problem (5): we introduce a redundant “reference” variable $r_{0:N}$ constrained to satisfy $C_{0:N} \cdot r_{0:N} = z_{0:N}$, and we then relax this equality constraint to a soft constraint in the objective function of the problem. The resulting relaxed optimization problem is then given by

$$\begin{aligned} \underset{r_{0:N}}{\text{minimize}} \quad & \mathcal{E}(C_{0:N} \cdot r_{0:N}) \\ \text{s.t.} \quad & C_{0:N} \cdot r_{0:N} \in \mathcal{R} \end{aligned} \quad + \overbrace{\underset{x_{0:N}, u_{0:N-1}}{\text{minimize}} \left\| \begin{array}{c} \rho C_{0:N} \cdot (x_{0:N} - r_{0:N}) \\ D_{0:N-1} \cdot u_{0:N-1} \end{array} \right\|_w}_{\text{Tracking Problem}} \quad (6)$$

s.t. dynamics (1)
distributed constraints (4)

where $\rho > 0$ is the tracking weight. This relaxation achieves our desired goal of separating the planning problem from the tracking problem: in particular, for a fixed $r_{0:N}$ the right most optimization problem, labeled as “Tracking Problem” in optimization problem (6), is completely independent of the cost function $\mathcal{E}(\cdot)$ and the constraint set \mathcal{R} .

Remark 1 *Note that with this relaxation, we can no longer guarantee that the state trajectory satisfies $C_{0:N} \cdot x_{0:N} \in \mathcal{R}$. Although this may seem unsatisfying, it is worth noting (i) that relaxing state constraints in this way is a standard approach in the LAO framework; (ii) that the optimal reference trajectory satisfies $C_{0:N} \cdot r_{0:N} \in \mathcal{R}$; and (iii) that under nominal operating conditions (i.e., for the noise $w_{0:N}$ set to 0), we have that $C_{0:N} \cdot x_{0:N} \approx C_{0:N} \cdot r_{0:N}$ for sufficiently large ρ .*

3.1 A Layered Architecture

Assume that for a fixed reference trajectory $r_{0:N}$ that the tracking problem specified in optimization problem (6) admits an analytic solution such that its objective at optimality is given by a function $f_\rho^{\text{track}}(r_{0:N})$ that is convex in $r_{0:N}$, and further assume that we can compute the optimal feedback policies γ_t^i such that the controller achieving the tracking cost $f_\rho^{\text{track}}(r_{0:N})$ can be implemented as a function of the reference trajectory $r_{0:N}$. We may then rewrite optimization problem (6) as

$$\underset{r_{0:N}}{\text{minimize}} \mathcal{C}(C_{0:N} \cdot r_{0:N}) + f_\rho^{\text{track}}(r_{0:N}) \text{ s.t. } C_{0:N} \cdot r_{0:N} \in \mathcal{R}. \quad (7)$$

Optimization problem (7) has an appealing architectural interpretation, as illustrated in Figure 1. The low-level tracking layer consists of the distributed optimal controller in feedback with the distributed plant, and drives the evolution of the system to match that specified by the reference trajectory received from the planning layer above. In order to determine this reference trajectory, the planning layer optimizes the weighted sum of a utility cost function \mathcal{C} and a tracking penalty f_ρ^{track} . The tracking penalty f_ρ^{track} should be interpreted as a *model* or *simulation* of the tracking layer's response to a given reference trajectory $r_{0:N}$. By replacing the explicit dynamic constraints (1) with the function f_ρ^{track} , which captures the behavior of the optimal closed loop response of the system, optimization problem (7) is now *static*. The static nature of the problem implies that much as in the LAO framework, well established distributed optimization techniques can be applied to solve it. Furthermore, as the reference trajectory $r_{0:N}$ is now a virtual quantity, we are not limited to using the system dynamics to implement an optimization algorithm, potentially allowing for more sophisticated methods to be used.

In the next section, we present a class of problems for which the prerequisite analytic expressions for f_ρ^{track} and γ_t^i can be obtained using existing methods from the traditional and distributed optimal control literature.

4 Tracking Problems with Analytic Solutions

In this section we focus on a tracking problem with a LQG like cost-function that is subject to centralized reference information constraints and distributed state information constraints. We make precise what these information sharing constraints mean and show that in this setting, the optimal tracking policy is unique and linear in its information, and can be constructed using existing results from the distributed optimal control literature. We defer a discussion of alternative noise assumptions, performance metrics, and more complex reference information constraints to §5.

We take the disturbances w_t to be identically and independently drawn from a zero mean normal distribution with identity covariance, i.e., $w_t \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$, and use the mean square error signal-to-signal metric, i.e., $\|z_{0:N}\|_w = \sum_{t=0}^N \mathbb{E}_w \|z_t\|_2^2$. Let $e_t := x_t - r_t$ denote the tracking error between the state x_t and the reference r_t , and let $\mathbf{r}_t^\top := [r_t^\top \ r_{t+1}^\top \ \cdots \ r_N^\top \ 0^\top \ \cdots \ 0^\top] \in \mathbb{R}^{1 \times Nn}$ be a suitably zero-padded stacked vector of references $r_{t:N}$. Letting

$$\tilde{A}_t := \begin{bmatrix} A_t & (A_t E_1^\top - E_2^\top) \\ 0 & Z \end{bmatrix}, \quad \tilde{B}_t := \begin{bmatrix} B_t \\ 0 \end{bmatrix}, \quad \tilde{H}_t := \begin{bmatrix} H_t \\ 0 \end{bmatrix},$$

we can then rewrite the Tracking Problem specified in optimization problem (6) in terms of e_t and

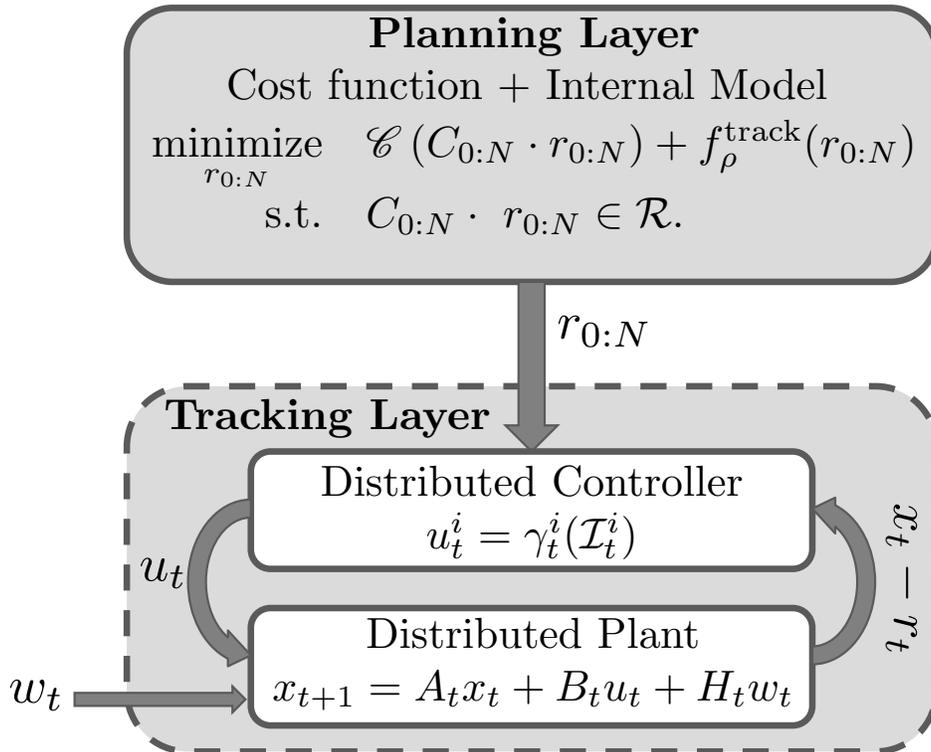


Figure 1: Layered architecture obtained through our proposed relaxation based solution to optimization problem (5). The architecture has two layers, a low-level tracking, or reflex, layer, and a high-level planning layer. Whereas the tracking layer is cost-function-agnostic, the planning layer possesses an *internal model* of the tracking layer’s ability to follow a given reference trajectory.

\mathbf{r}_t as

$$\begin{aligned}
& \underset{e_{0:N}, u_{0:N}}{\text{minimize}} && \sum_{t=0}^{N-1} \mathbb{E}_w [e_t^\top Q_t e_t + u_t^\top R_t u_t] + \\
& && \mathbb{E}_w [e_N^\top Q_N e_N] \\
& \text{s.t.} && \begin{bmatrix} e_{t+1} \\ \mathbf{r}_{t+1} \end{bmatrix} = \tilde{A}_t \begin{bmatrix} e_t \\ \mathbf{r}_t \end{bmatrix} + \tilde{B}_t u_t + \tilde{H}_t w_t \\
& && e_0 \text{ and } \mathbf{r}_0 \text{ given, } u_t^i = \gamma_t^i(\mathcal{I}_t^i \cup \mathcal{J}_t^i) \text{ for all } i,
\end{aligned} \tag{8}$$

where $Q_t := \rho C_t^\top C_t$ and $R_t := D_t^\top D_t$.

Here the state information set \mathcal{I}_t^i available to node i at time t is specified by (3), and the corresponding reference information set \mathcal{J}_t^i is specified by

$$\mathcal{J}_t^i := \left\{ \mathbf{r}_{0:t-\kappa_{i1}}^1, \mathbf{r}_{0:t-\kappa_{i2}}^2, \dots, \mathbf{r}_{0:t}^i, \dots, \mathbf{r}_{0:t-\kappa_{ip}}^p \right\} \tag{9}$$

where κ_{ij} is the reference communication delay from node j to node i . We allow for $\kappa_{ij} \leq \tau_{ij}$, i.e., for the stacked reference trajectory \mathbf{r}_t to possibly be communicated faster than states x_t of the system, so that we may accommodate cases in which reference following is performed in a centralized manner but disturbance rejection is performed in a distributed manner.

In this section, we assume that information available to the planning layer is globally available to the tracking layer, i.e., we assume that $\mathcal{J}_t^i = \{x_0, \mathbf{r}_{0:t}\}$ for all sub-controllers i , but that information pertaining to the state, i.e., \mathcal{I}_t^i , is constrained as in (3) but *partially nested* [12]. As stated in the following theorem, these assumptions on the information sharing constraints coupled with the choice of a quadratic objective function imply that the optimal control policies γ_t^i exist, are unique, and are linear in their information sets. We then use this result to show that a separation principle exists, allowing us to decompose the control task into two independent subproblems: a reference following problem and a disturbance rejection problem.

Theorem 1 *Consider optimization problem (8), and suppose that $\mathcal{J}_t^i = \{x_0, \mathbf{r}_{0:t}\}$ and that \mathcal{I}_t^i defines a partially nested information sharing constraint. Then the optimal control policies γ_t^i exist, are unique, and are linear in their information set.*

Proof: See Appendix. ■

An analogous result is proved in [12] for distributed control problems with a quadratic objective, and initial conditions and disturbances drawn from zero mean Gaussian distributions. In the tracking problem (8), the initial conditions e_0 and \mathbf{r}_0 are nonzero and deterministic, and hence the proof needs to be suitably modified to accommodate this fact. Intuitively, the argument of [12] applies here because we assume that the nonzero initial conditions e_0 and \mathbf{r}_0 are globally known, and hence the effect of the initial conditions can be computed by each sub-controller, returning us to the zero mean setting.

Exploiting the linearity of the optimal policies, we show in the Appendix that we can then decompose the error state e_t into a deterministic component d_t and a centered stochastic component s_t that obey the following dynamics:

$$d_{t+1} = \tilde{A}_t d_t + \tilde{B}_t \mu_t, \quad d_0 = \begin{bmatrix} e_0 \\ \mathbf{r}_0 \end{bmatrix} \tag{10a}$$

$$s_{t+1} = \tilde{A}_t s_t + \tilde{B}_t \nu_t + \tilde{H}_t w_t, \quad s_0 = 0 \tag{10b}$$

$$\begin{bmatrix} e_t \\ \mathbf{r}_t \end{bmatrix} = s_t + d_t. \tag{10c}$$

where μ_t is a linear map acting on the globally available deterministic information set $\mathcal{D}_t = \{d_{0:t}\}$ and ν_t is a linear map with components ν_t^i satisfying $\nu_t^i = \nu_t^i(\mathcal{S}_t^i)$, with the stochastic information set \mathcal{S}_t^i available to node i at time t specified by

$$\mathcal{S}_t^i := \left\{ s_{0:t-\tau_{i1}}^1, s_{0:t-\tau_{i2}}^2, \dots, s_{0:t}^i, \dots, s_{0:t-\tau_{ip}}^p \right\}. \quad (11)$$

Notice in particular that the delays specifying the information sharing constraints in \mathcal{S}_t^i are inherited from the state information sharing constraints as specified in equation (3). Further, for all $t \geq 0$, it holds that $\mathbb{E}[s_t] = 0$ and consequently that $\mathbb{E}[d_t s_t^\top] = 0$. The following Corollary is then immediate.

Corollary 1 *Suppose that the assumptions of Theorem 1 hold. Then the tracking problem (8) can be decomposed into a reference following problem (RFP)*

$$\begin{aligned} & \underset{d_{0:N}, \mu_{0:N}}{\text{minimize}} && \sum_{t=0}^{N-1} d_t^\top Q_t d_t + \mu_t^\top R_t \mu_t + d_N^\top Q_N d_N \\ & \text{s.t.} && \text{dynamics (10a), } \mu_t = \mu_t(\mathcal{D}_t), \end{aligned} \quad (\text{RFP})$$

and a disturbance rejection problem (DRP)

$$\begin{aligned} & \underset{s_{0:N}, \nu_{0:N}}{\text{minimize}} && \sum_{t=0}^{N-1} \mathbb{E} [s_t^\top Q_t s_t + \nu_t^\top R_t \nu_t] + \mathbb{E} [s_N^\top Q_N s_N] \\ & \text{s.t.} && \text{dynamics (10b), } \nu_t^i = \nu_t^i(\mathcal{S}_t^i) \text{ for all } i. \end{aligned} \quad (\text{DRP})$$

The consequences of this corollary are profound – in particular, notice that the disturbance reference problem (DRP) is independent of the reference trajectory \mathbf{r}_t . This implies that the tracking penalty f_ρ^{track} to be used in the planning layer optimization problem (7) is completely specified by the optimal cost of the reference following problem (RFP), which is easily seen to be a standard centralized linear quadratic regulator (LQR) problem. We therefore have that the optimal tracking cost is given by

$$f_\rho^{\text{track}}(r_{0:N}) = \begin{bmatrix} x_0 - r_0 \\ \mathbf{r}_0 \end{bmatrix}^\top P_0 \begin{bmatrix} x_0 - r_0 \\ \mathbf{r}_0 \end{bmatrix}, \quad (12)$$

and achieved by the policy $\mu_t = -K_t d_t$, for P_t and K_t specified by the Riccati recursions

$$\begin{aligned} P_t &= \tilde{A}_t^\top \begin{bmatrix} Q_t & 0 \\ 0 & 0 \end{bmatrix} \tilde{A}_t - \tilde{A}_t^\top P_{t+1} \tilde{B}_t K_t, \quad P_N = \begin{bmatrix} Q_N & 0 \\ 0 & 0 \end{bmatrix}, \\ K_t &= \left(R_t + \tilde{B}_t^\top P_{t+1} \tilde{B}_t \right)^{-1} \tilde{B}_t^\top P_{t+1} \tilde{A}_t. \end{aligned} \quad (13)$$

Finally, the optimal control policies ν_t^i for the disturbance rejection problem (DRP) can be computed for many interesting information patterns using existing results from the classical and distributed optimal control literatures – a selection of relevant results are listed in Table 1. The resulting optimal costs change as a function of the information constraints of the system, but they are independent of the reference trajectory $r_{0:N}$ and do not affect the planning optimization problem (7).

5 Discussion

This section provides a brief discussion of useful extensions of the results presented thus far. It is impossible to cover these extensions in detail due to length constraints: instead, we motivate and pose the relevant problems, sketch possible solutions and highlight the technical challenges, if any, that need to be overcome. We defer a detailed treatment of each of these areas to our forthcoming paper [26].

Information Pattern	Reference
Centralized SF + OF	[15]
Delay and/or Sparsity Constrained SF	[18, 21, 22]
Delay and/or Sparsity Constrained OF	[23–25]

Table 1: A selection of relevant results from distributed \mathcal{H}_2 optimal control that can be applied to the disturbance rejection problem (DRP). This list is far from exhaustive and we recommend the interested reader consult the references found within the cited papers. We note that some of the results are presented for infinite horizon problems, but are easily modified to accommodate finite horizon problems with zero mean initial conditions. Here SF and OF denote state and output feedback, respectively.

5.1 Virtual Dynamics & Recursive Layered Architectures

The layered architecture presented in §3 is quite extreme, consisting of a static planning problem and a dynamic tracking layer – in practice, it may be desirable for a cyber-physical system to have more than two layers, with each layer operating on different simplified models of the underlying system. The vertical layering approach that we describe in §3 can be modified to include virtual (and possibly simpler) dynamics in the planning layer: in such a way the relaxation approach can be applied recursively to yield a multi-layered system.

Before explaining how this can be done, consider a familiar “live demo” that illustrates the need for the inclusion of several layers, with higher layers using simpler dynamic models. In particular, consider the task of filling a glass of water and taking a drink. Any conscious planning and execution is typically done with simple arm and glass positions and velocities. These positions and velocities are then converted to the muscle torques and forces needed to move the glass and compensate for the changing water weight. Similarly, these actions at the muscular level are effected by changes at the cellular level, and in turn actions at the cellular level are effected by changes at the macromolecular level. In this example the use of layering allows for functional tasks to be rapidly planned and executed in a simple virtual space, without explicitly accounting for the bewildering complexity of our underlying physiology and biochemistry.

To see how such a recursive layered architecture can be synthesized using our suggested relaxation approach, consider the following modified planning problem obtained by adding virtual dynamics (described by state matrices M_t and control matrices N_t) to (7):

$$\begin{aligned}
& \underset{r_{0:N}}{\text{minimize}} && \mathcal{E}(r_{0:N}^{\text{state}}) + f_{\rho}^{\text{track}}(r_{0:N}) \\
& \text{s.t.} && r_{0:N}^{\text{state}} \in \mathcal{R} \\
& && r_{t+1}^{\text{state}} = M_t r_t^{\text{state}} + N_t r_t^{\text{input}}.
\end{aligned} \tag{14}$$

Here we have partitioned the reference r_t into a virtual state component r_t^{state} and a virtual control input component r_t^{input} , and assumed that $C_{0:N} \cdot r_{0:N} = r_{0:N}^{\text{state}}$ – this specific decomposition of the reference state is not necessary, but makes the exposition clearer.

Notice that problem (14) is of a very similar form to that of the optimization problem (5) that we originally considered, and we can apply a similar vertical decomposition to that used to obtain (6). In particular, by introducing a redundant higher-level virtual quantity h_t constrained to satisfy $h_t = r_t^{\text{state}}$ and suitably relaxing this to a soft constraint in the objective, we obtain

$$\begin{aligned}
& \underset{h_{0:N}}{\text{minimize}} && \mathcal{E}(h_{0:N}) \\
& \text{s.t.} && h_{0:N} \in \mathcal{R} \\
& && + \overbrace{\underset{r_{0:N}}{\text{minimize}} \lambda \|h_{0:N} - r_{0:N}^{\text{state}}\| + f_{\rho}^{\text{track}}(r_{0:N})}^{\text{Intermediate Tracking Problem}} \\
& && \text{s.t.} \quad r_{t+1}^{\text{state}} = M_t r_t^{\text{state}} + N_t r_t^{\text{input}},
\end{aligned} \tag{15}$$

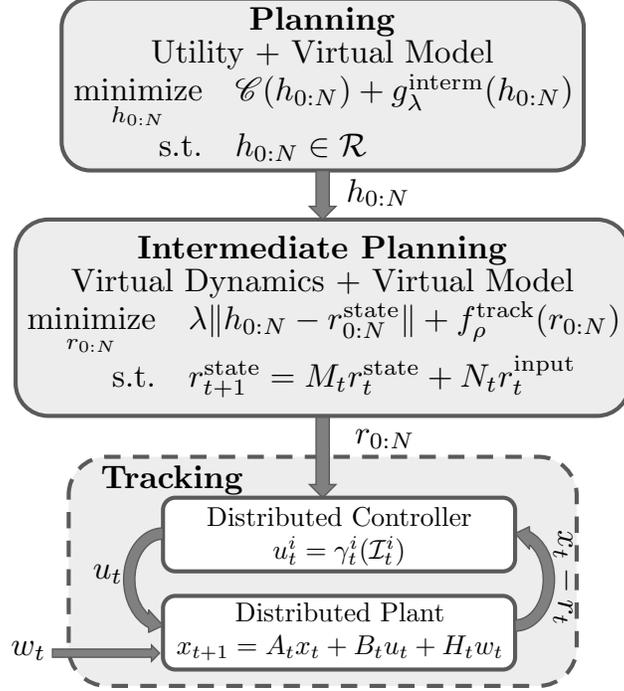


Figure 2: Three-tiered layered architecture obtained by recursively applying our proposed relaxation based solution to optimization problem (5). The architecture has three layers: a low-level tracking, or reflex, layer, an intermediate tracking layer acting on virtualized dynamics, and a high-level static planning layer.

where $\lambda > 0$ denotes the “intermediate tracking problem” weight. As in §3, assume that an analytic solution to the intermediate tracking problem is available such that its cost can be expressed via a function $g_\lambda^{\text{interm}}(v_{0:N})$ convex in $v_{0:N}$, and that this cost can be achieved using a feedback policy $r_t^{\text{input}}(h_{0:t})$ (this assumption holds if LQG like penalties are used throughout). Optimization problem (15) then reduces to a planning optimization problem of exactly the same form as (7) save for the use of $g_\lambda^{\text{interm}}$ in lieu of f_ρ^{track} . We illustrate the resulting three-tiered layered architecture in Fig. 2.

We end this section with a final observation and important direction for future work. We note that the higher-level virtual state $h_t \approx r_t^{\text{state}}$ is of a smaller dimension than that of the full reference quantity r_t . This suggests an approach to performing model reduction in the context of a layered architecture of optimal controllers. In particular, the tracking penalty f_ρ^{track} restricted to the range of the virtual dynamics can be used to quantify the ability of the tracking-layer to emulate the simple and possibly lower-order virtual dynamics used in the higher layers, thus providing a measure of how well the desired virtualization can be implemented. Formalizing this intuition and exploring its consequences is the subject of current research.

5.2 Real-Time Planning

As of yet, an implicit assumption has been that the planner is able to solve optimization problem (7) before the system moves. In general, planning will have to occur in real time. A straightforward way to incorporate real-time planning into our framework is to have the planning layer implement an optimization algorithm, such as gradient descent, and to allow the planning layer to run K iterations of this algorithm per time step t of the dynamics. The updates to the reference trajectory are simply modeled as noise entering the reference trajectory state \mathbf{r}_t in the augmented state dynamics specified

in (8). In particular, the error dynamics are now given by

$$\begin{bmatrix} e_{t+1} \\ \mathbf{r}_{t+1} \end{bmatrix} = \tilde{A}_t \begin{bmatrix} e_t \\ \mathbf{r}_t \end{bmatrix} + \tilde{B}_t u_t + \tilde{H}_t w_t + \begin{bmatrix} 0 \\ I \end{bmatrix} \Delta \mathbf{r}_t, \quad (16)$$

where $\Delta \mathbf{r}_t$ contains the reference trajectory update computed after K iterations of the optimization algorithm, i.e., $\Delta \mathbf{r}_t = \mathbf{r}_t^{(tK)} - \mathbf{r}_t^{((t-1)K)}$ for $\mathbf{r}_t^{(k)}$ the k th iterate of the algorithm.

5.3 Alternative Tracking Metrics

Using a LQG like tracking metric allows us to decompose the tracking problem (8) into two independent subproblems, each of which can be solved using existing techniques from the literature. This separation principle is not likely to extend to induced norms such as the \mathcal{H}_∞ or \mathcal{L}_1 norms, thus motivating the need to holistically consider the tracking problem. For instance, if both the state information set \mathcal{I}_t and reference information set \mathcal{J}_t are centralized and the tracking metric is taken to be a finite horizon \mathcal{H}_∞ cost function, then the methods from [27] can be used to solve the resulting tracking problem. Interestingly, the tracking penalty f_ρ^{track} is also a quadratic form, just as the tracking penalty (12) of the LQG tracking problem, but is specified by the solution to a modified Riccati recursion. As far as we are aware, no exact solutions exist to the finite horizon \mathcal{L}_1 optimal control problem with known initial conditions – recent work [28] indicates that such performance metrics are particularly relevant to sensorimotor control, thus motivating further study of such problems.

5.4 Distributed Reference Following

Although the centralized reference following and distributed disturbance rejection control architecture studied in §4 has direct applications to several areas (e.g., sensorimotor control and datacenter control), there is nonetheless an obvious motivation for distributing the reference following tasks in large-scale systems (e.g., wide area networks and the power-grid). Unfortunately, under such information sharing constraints, it is not clear if the arguments used to prove Theorem 1 can be modified to show that linear control laws are optimal – in fact, it is not even clear if the optimal policy is independent of the reference trajectory, or if it is a feedback policy. These questions arise because of the lack of a so-called central coordinator that has access to the global plant model and noise statistics – the existence of such a coordinator is either implicitly [12] or explicitly assumed [29] in the distributed optimal control literature.

However, if we are satisfied with restricting ourselves to linear feedback policies, quadratic costs and jointly Gaussian disturbances, then the decomposition described in the previous section still holds (subject to suitable constraints on the reference sharing constraints κ_{ij}). Unfortunately, as far as we are aware, no results in the distributed optimal control literature are able to accommodate *known nonzero initial conditions* – recall that in our framework, the initial conditions of the tracking problem are specified by a known initial error e_0 and a reference trajectory $r_{0:N}$. In the case of \mathcal{H}_2 distributed optimal control, statistical independence between subsets e_0^i and \mathbf{r}_0^i of the initial conditions and disturbances w_t^i is the main tool used to solve the controller synthesis task (e.g., [18, 21]): however, known nonzero initial conditions breaks this statistical independence, thus preventing us from applying these methods. We were able to circumvent this issue in §4 by assuming that the deterministic nonzero initial conditions are globally known, but if we assume information sharing constraints on the reference trajectory \mathbf{r}_0 , then this approach no longer works.

A possible approach to overcoming this limitation in the theory is to further relax the tracking problem by embedding the initial condition x_0 and reference trajectory $r_{0:N}$ into the disturbance

signal during the controller synthesis procedure. This allows for a much wider class of tracking problems to be solved using existing methods from the literature: in addition to the methods summarized in Table 1, methods for distributed \mathcal{H}_∞ optimal control [30–32] and \mathcal{L}_1 optimal control [33] can be leveraged. If this approach is taken, then computing the resulting tracking penalty f_ρ^{track} is more involved – in particular, we must view this as a *simulation* of the closed loop response of the system to a given x_0 and $r_{0:N}$. Thus the proposed relaxation can be viewed as one of synthesizing the tracking layer controller for average-case (if an LQG like tracking metric is used) or worst-case (if an induced norm tracking metric is used) reference trajectories and initial conditions. However, the planning layer simulates, via the internal model of the tracking layer captured in the tracking penalty f_ρ^{track} , the response of the system to the *specific* reference trajectory being considered when solving the planning problem (7). We defer a general statement and analysis of this approach to [26], but informally consider an example here. In the case of LQG optimal control we can exploit the independence of the noise terms to decompose the the closed loop cost into a term measuring the effect of the disturbances w_t and a term measuring the effect of the initial error e_0 and the reference trajectory \mathbf{r}_0 . Letting $G_{\mathbf{r}}(e_0, \mathbf{r}_0)$ denote the closed loop response of the system to initial conditions specified by e_0 and \mathbf{r}_0 , the corresponding tracking penalty is then given by $f_\rho^{\text{track}}(r_{0:N}) = \|G_{\mathbf{r}}(e_0, \mathbf{r}_0)\|_{\mathcal{H}_2}^2$.

5.5 Distributed Planning (Horizontal Decompositions)

A strength of the LAO framework is that if the utility function $\mathcal{C}(\cdot)$ and state constraints \mathcal{R} are suitably structured, then the resulting optimization problem can be *horizontally* decomposed and solved in a distributed manner, allowing the approach to scale to very large systems. Although such real-time distributed optimization schemes can easily be incorporated into the proposed framework (as described in §5.2) other challenges must be overcome. In particular, assuming that the utility function $\mathcal{C}(\cdot)$ and state constraints \mathcal{R} are suitably structured is not sufficient to ensure that the planning problem (7) admits a similar horizontal decomposition. This is because, in general, the closed loop response of a system will be dense, even if the dynamics and controller are distributed – this results in a dense penalty f_ρ^{track} .

This motivates the need for synthesizing distributed controllers that lead to (approximately) sparse closed loop responses such that the resulting tracking penalty f_ρ^{track} is also (approximately) sparse, and hence amenable to horizontal decomposition. The recently developed localized optimal control framework [10, 11] yields such sparse closed loop responses – however, the localized optimal control method is only applicable to problems for which the initial condition is embedded into the disturbance signal, and thus the additional relaxation to the tracking problem described in §5.4 must be applied. Alternatively, it has been observed that structured systems often have approximately sparse optimal closed loop responses, e.g., [34, 35]. In the case of an LQG tracking metric, this implies that the matrix P_0 specifying the tracking penalty (12) is approximately sparse. Thus one may find a structured matrix Q_0 satisfying $Q_0 \succeq P_0$, and use the corresponding quadratic form in the tracking problem (8). This can be interpreted as a conservative localized approximation of the closed loop response by the planning layer – if the closed loop response is already approximately sparse, then this approximation introduces little to no conservatism into reference trajectory $r_{0:N}$ generated by the planning layer.

6 Numerical Example

Consider linear time-invariant (LTI) dynamics (1) describing a discrete-time single integrator with sampling time τ :

$$A_{\text{di}} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \quad B_{\text{di}} = \begin{bmatrix} 0 \\ \tau \end{bmatrix}, \quad H_{\text{di}} = .1\tau I_2$$

and suppose that two such double integrators are dynamically coupled via randomly generated matrices A_{12} and A_{21} . The dynamics (1) of the coupled system are then specified by

$$A_{2\text{di}} = \begin{bmatrix} A_{\text{di}} & A_{12} \\ A_{21} & A_{\text{di}} \end{bmatrix}, \quad B_{2\text{di}} = \begin{bmatrix} B_{\text{di}} & \\ & B_{\text{di}} \end{bmatrix}, \quad H_{2\text{di}} = \begin{bmatrix} H_{\text{di}} & \\ & H_{\text{di}} \end{bmatrix}$$

In this example, we impose the following distributed constraints on the control laws

$$u_t^1 = \gamma_t(x_{0:t}^1, x_{0:t-1}^2, x_0, \mathbf{r}_{0:t}), \quad u_t^2 = \gamma_t(x_{0:t-1}^1, x_{0:t}^2, x_0, \mathbf{r}_{0:t}),$$

i.e., each subsystem can communicate its state to its neighbor with a delay of one time-step, and that the reference trajectory is globally available. We follow the approach described in §4 and use an LQG-like tracking metric: the resulting tracking problem (8) satisfies the assumptions of Theorem 1 and Corollary 1. Thus we can decompose the tracking problem into a centralized reference following program (RFP) and a distributed disturbance rejection problem (DRP), where this latter problem can be solved via the methods of [21].

We take the control penalty $R_t = .01I_2$, let $v_t = .1 \sin(\frac{2\pi}{10}t)$ and define the cost function as $\mathcal{C}(r_{0:N}) = \sum_{t=0}^N |r_t^1(1) - v_t| + |r_t^2(1) - v_t|$, where $r_t^i(1)$ denotes the reference position coordinate of double-integrator i . We assume for now that there are no state-constraints \mathcal{R} . Thus the planning goal is to minimize the total variation between the position of each double integrator and a sinusoidal reference trajectory. For our numerical examples, we set $\tau = .1$ and $x_0 = [.2, 0, -.2, 0]^\top$.

Shown in Fig. 3 are the state trajectories for varying levels of ρ – as the system is being driven by noise, the reference is not tracked exactly, even as ρ grows. We also solve the problem with no driving noise (i.e., $H_{\text{di}} = 0$): the result is shown in the last subfigure of Fig. 3. In this case, we can solve the original optimization problem (5), and numerically verify that the solution computed via relaxation (6) is exact for sufficiently large ρ .

In the previous examples, we assumed that the planner was able to solve optimization problem (7) before the system moved or the dynamics had any effect on the state, but as discussed in §5.2 this assumption can be relaxed. Illustrated in Figure 4 is the distributed control problem described above with real-time planning, with $\rho = 1000$ and $H_{\text{di}} = 0$ (we set the process noise w_t to zero so as to make clear the effect of real-time planning) and various values of optimization algorithm iterations K per time step t of the dynamics. For this example, we incorporated a state constraint that imposes that the position of both double integrators have magnitude no larger than .05, and we used a projected subgradient descent method with fixed step-size to solve the problem. As can be seen, the state tracks the current iterate of the optimal trajectory generated by the optimization algorithm, and these iterates converge to the true optimal trajectory.

References

- [1] J. C. Doyle and M. Csete, “Architecture, constraints, and behavior,” *PNAS*, vol. 108, no. Supplement 3, pp. 15 624–15 630, 2011.
- [2] M. Chiang, S. Low, A. Calderbank, and J. Doyle, “Layering as optimization decomposition: A mathematical theory of network architectures,” *Proc. of the IEEE*, vol. 95, no. 1, pp. 255–312, Jan 2007.

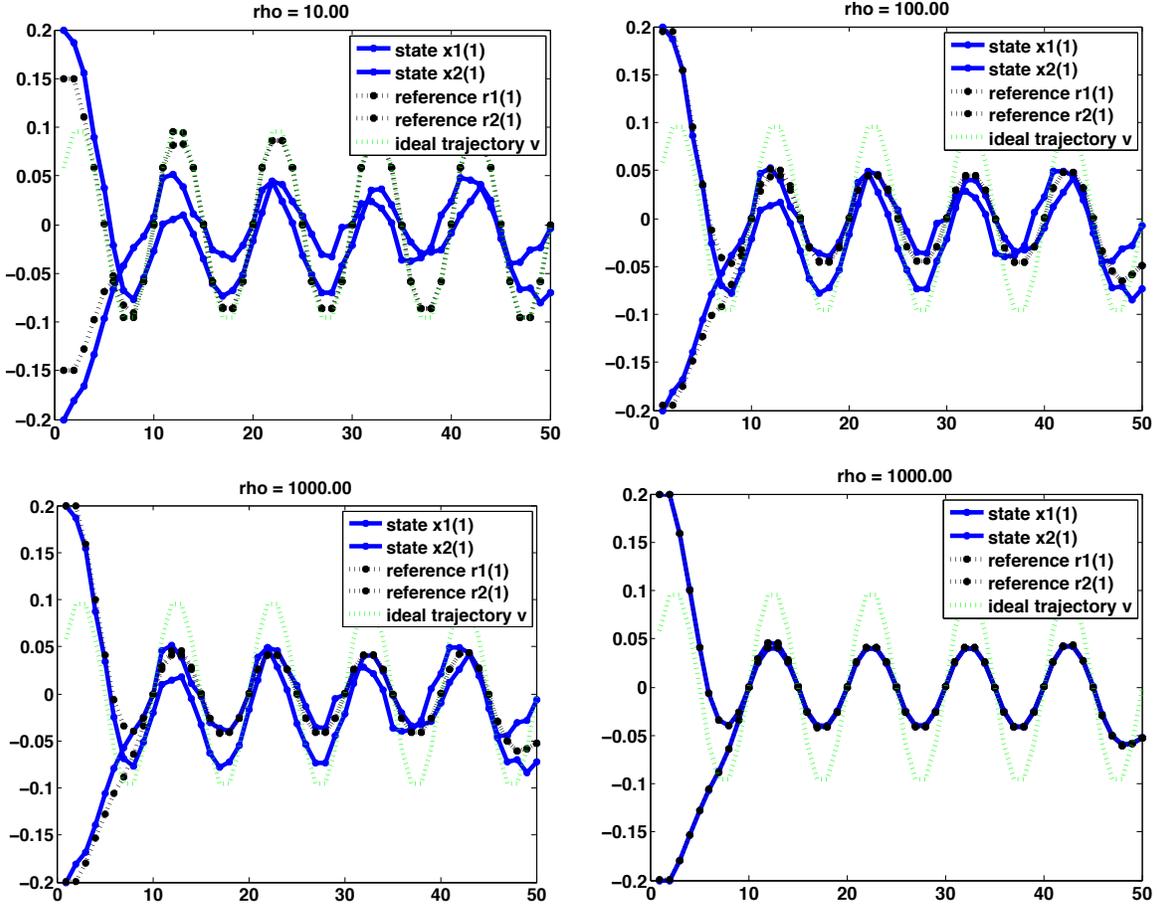


Figure 3: As ρ increases, the relaxation becomes increasingly tight – note that as the system is being driven by noise, the reference is not tracked exactly even as ρ grows. For reference, the last figure illustrates the state-trajectory when there is no driving noise. Notice that the optimal trajectory is different from v_t because of the nonzero control cost $R_t = .01I_2$.

- [3] D. Palomar and M. Chiang, “A tutorial on decomposition methods for network utility maximization,” *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 8, pp. 1439–1451, Aug 2006.
- [4] C. Zhao, U. Topcu, N. Li, and S. Low, “Design and stability of load-side primary frequency control in power systems,” *Automatic Control, IEEE Transactions on*, vol. 59, no. 5, pp. 1177–1189, May 2014.
- [5] D. Cai, E. Mallada, and A. Wierman, “Distributed optimization decomposition for joint economic dispatch and frequency regulation,” in *IEEE Conference on Decision and Control (CDC)*. To appear., 2015.
- [6] T. Prescott and A. Papachristodoulou, “Signal propagation across layered biochemical networks,” in *American Control Conference (ACC), 2014*, June 2014, pp. 3399–3404.
- [7] H. El-Samad, H. Kurata, J. C. Doyle, C. A. Gross, and M. Khammash, “Surviving heat shock: Control strategies for robustness and performance,” *PNAS*, vol. 102, no. 8, pp. 2736–2741, 2005.

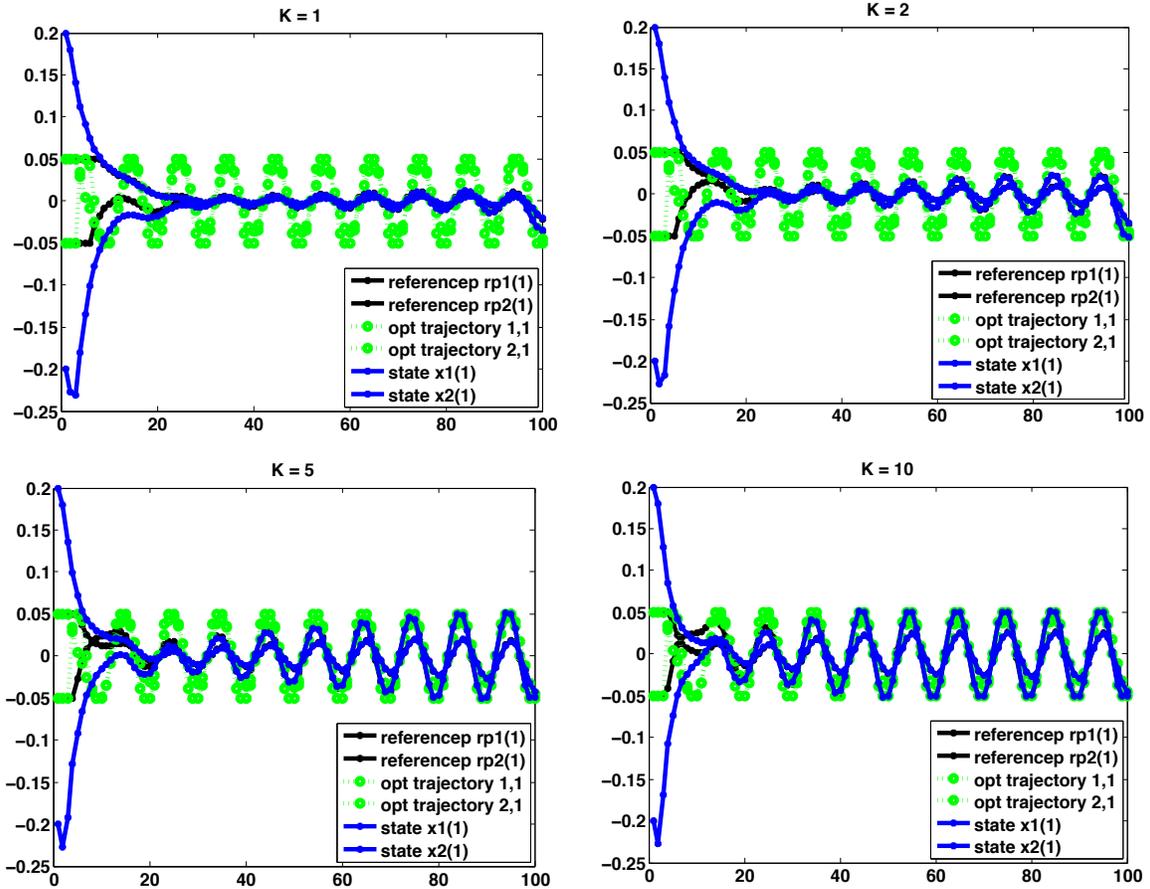


Figure 4: As the number of optimization algorithm iterations K per time step t increases, the reference trajectory (shown in black) converges to the optimal one (shown in green) increasingly quickly. As can be seen, the changes in reference trajectory act as noise in the error system, and thus despite a high value of ρ , the state does not follow the reference trajectory exactly – this is especially true at earlier times when there are larger changes in the iterates, corresponding to larger $\Delta \mathbf{r}_t$ affecting the error dynamics (16).

- [8] M. Rotkowitz and S. Lall, “A characterization of convex problems in decentralized control,” *Automatic Control, IEEE Transactions on*, vol. 51, no. 2, pp. 274–286, 2006.
- [9] A. Mahajan, N. Martins, M. Rotkowitz, and S. Yuksel, “Information structures in optimal decentralized control,” in *Decision and Control (CDC), IEEE 51st Annual Conference on*, Dec 2012, pp. 1291–1306.
- [10] Y.-S. Wang, N. Matni, and J. Doyle, “Localized LQR optimal control,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 1661–1668.
- [11] Y.-S. Wang and N. Matni, “Localized LQG optimal control for large-scale systems,” in *ACC 2016. Submitted.*, 2016.
- [12] Y.-C. Ho and K.-C. Chu, “Team decision theory and information structures in optimal control problems—part i,” *Automatic Control, IEEE Transactions on*, vol. 17, no. 1, pp. 15–22, 1972.

- [13] H. S. Witsenhausen, “A counterexample in stochastic optimum control,” *SIAM Journal on Control*, vol. 6, no. 1, pp. 131–147, 1968.
- [14] M. Rotkowitz, R. Cogill, and S. Lall, “Convexity of optimal control over networks with delays and arbitrary topology,” *International Journal of Systems, Control and Communications*, vol. 2, no. 1-3, pp. 30–54, 2010.
- [15] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.
- [16] J. Swigart and S. Lall, “An explicit state-space solution for a decentralized two-player optimal linear-quadratic regulator,” in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 6385–6390.
- [17] P. Shah and P. A. Parrilo, “ \mathcal{H}_2 -optimal decentralized control over posets: A state space solution for state-feedback,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, 2010, pp. 6722–6727.
- [18] A. Lamperski and L. Lessard, “Optimal state-feedback control under sparsity and delay constraints,” in *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2012, pp. 204–209.
- [19] M. Morari and J. H. Lee, “Model predictive control: past, present and future,” *Comp. & Chem. Eng.*, vol. 23, no. 4–5, pp. 667 – 682, 1999.
- [20] P. D. Christofides, R. Scattolini, D. M. de la Peña, and J. Liu, “Distributed model predictive control: A tutorial review and future research directions,” *Comp. & Chem. Eng.*, vol. 51, pp. 21 – 41, 2013.
- [21] A. Lamperski and J. Doyle, “Dynamic programming solutions for decentralized state-feedback lqg problems with communication delays,” in *American Control Conference (ACC)*, June 2012, pp. 6322–6327.
- [22] P. Shah and P. Parrilo, “ \mathcal{H}_2 -optimal decentralized control over posets: A state-space solution for state-feedback,” *Automatic Control, IEEE Transactions on*, vol. 58, no. 12, pp. 3084–3096, Dec 2013.
- [23] A. Lamperski and J. C. Doyle, “The \mathcal{H}_2 control problem for quadratically invariant systems with delays,” *Automatic Control, IEEE Transactions on*. *To appear.*, 2015.
- [24] A. Nayyar and L. Lessard, “Optimal control for lqg systems on graphs—part i: Structural results,” *arXiv preprint arXiv:1408.2551*, 2014.
- [25] T. Tanaka and P. A. Parrilo, “Optimal output feedback architecture for triangular LQG problems,” *arXiv preprint arXiv:1403.4330*, 2014.
- [26] N. Matni and J. C. Doyle, “Incorporating dynamics into layering as optimization,” *In preparation.*, 2015.
- [27] T. Başar and P. Bernhard, *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media, 2008.

- [28] Y. Nakahira, N. Matni, and J. C. Doyle, “Hard limits on robust control over delayed and quantized communication channels with applications to sensorimotor control,” in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on. To appear.*, 2015.
- [29] A. Nayyar, A. Mahajan, and D. Teneketzis, “Decentralized stochastic control with partial history sharing: A common information approach,” *Automatic Control, IEEE Transactions on*, vol. 58, no. 7, pp. 1644–1658, 2013.
- [30] C. W. Scherer, “Structured \mathcal{H}_∞ -optimal control for nested interconnections: A state-space solution,” *arXiv:1305.1746*, 2013.
- [31] L. Lessard, “State-space solution to a minimum-entropy \mathcal{H}_∞ -optimal control problem with a nested information constraint,” in *Dec. and Control, IEEE 53rd Annual Conf. on*, Dec 2014, pp. 4026–4031.
- [32] N. Matni, “Distributed control subject to delays satisfying an \mathcal{H}_∞ norm bound,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 4006–4013.
- [33] M. A. Dahleh and I. J. Diaz-Bobillo, *Control of uncertain systems: a linear programming approach*. Prentice-Hall, Inc., 1994.
- [34] B. Bamieh, F. Paganini, and M. Dahleh, “Distributed control of spatially invariant systems,” *Automatic Control, IEEE Transactions on*, vol. 47, no. 7, pp. 1091–1107, Jul 2002.
- [35] N. Motee and Q. Sun, “Sparsity measures for spatially decaying systems,” in *American Control Conference (ACC)*, June 2014, pp. 5459–5464.

Proof: [Proof of Theorem 1 (sketch)] The proposed information sharing pattern is easily seen to be partially nested as \mathcal{J}_t^i is globally shared, and \mathcal{I}_t^i is assumed to be partially nested. It is shown in Theorems 1 & 2 of [12] that in this setting, it suffices to consider static team decision problems, as any finite horizon LQG problem can be reduced to such a static problem. Let $\xi^\top = [e_0^\top, \mathbf{r}_0^\top, w_0^\top, \dots, w_{N-1}^\top]^\top$ be the stacked vector of initial conditions and disturbances. Whereas [12] considers vectors ξ distributed as zero mean Gaussian random variables, we must consider nonzero mean Gaussian random variables because of the deterministic nonzero initial conditions specified by e_0 and \mathbf{r}_0 . Under the information sharing pattern $\mathcal{J}_t^i = \{x_0, \mathbf{r}_{0:t}\}$ this nonzero mean is globally known, allowing us to extend the argument surrounding equations (20)-(23) of [12]. The necessary modification occurs in equations (21) and (22) by replacing the expression for the conditional expectation $\mathbb{E}[\xi | \mathcal{I}_t^i \cup \mathcal{J}_t^i]$ with a suitably modified version to account for the nonzero mean. The resulting optimal policy is then a linear function of both the available information sets $\mathcal{I}_t^i \cup \mathcal{J}_t^i$ and the globally known nonzero mean. ■

Proof: [Proof of decomposition (10)] Our proof strategy is to show that there exists an invertible linear map from the information set $\mathcal{I}_t^i \cup \mathcal{J}_t^i$ to the deterministic information set \mathcal{D}_t and the stochastic information set \mathcal{S}_t^i . That this map is invertible is key to ensure that the proposed decomposition does not introduce conservatism into the solution to the tracking problem (8). We proceed by induction. At time $t = 0$, we set $\mathcal{S}_0^i = \{0\}$ and $\mathcal{D}_0 = \{d_0\} = \{e_0, \mathbf{r}_0\}$ – clearly there exists an invertible linear map from $\mathcal{I}_0^i \cup \mathcal{J}_0^i = \{0\} \cup \{x_0, \mathbf{r}_0\}$ to $\mathcal{D}_0 \cup \mathcal{S}_0^i$. By linearity, we can then decompose the control input at time zero as $u_0 = \mu_0(\mathcal{D}_0) + \nu_0$, where here $\nu_0 = 0$. It follows that equation (10c) holds for $t = 0$. Now assume that equation (10c) holds at time t , and that there exists an invertible linear map from $\mathcal{I}_t^i \cup \mathcal{J}_t^i$ to $\mathcal{D}_t \cup \mathcal{S}_t^i$. It follows that

$$\begin{bmatrix} e_{t+1} \\ \mathbf{r}_{t+1} \end{bmatrix} = \tilde{A}_t d_t + \tilde{B}_t \mu_t + \tilde{A}_t s_t + \tilde{B}_t \nu_t + \tilde{H}_t w_t = d_{t+1} + s_{t+1},$$

where d_{t+1} and s_{t+1} are as defined in equations (10a) and (10b). The first equality follows from the induction hypothesis and by exploiting the linearity of the optimal control law to write $u_t = \mu_t + \nu_t$, where μ_t is a linear function of \mathcal{D}_t and ν_t satisfies $\nu_t^i = \nu_t^i(\mathcal{S}_t^i)$. It is clear that d_{t+1} can be computed given the policy μ_t and d_t , and thus s_{t+1}^i may be computed locally as $s_{t+1}^i = z_{t+1}^i - d_{t+1}^i$, where here we write $z_t = [e_t^\top, \mathbf{r}_t^\top]$, thus defining a linear map from $\mathcal{I}_{t+1}^i \cup \mathcal{J}_{t+1}^i$ to $\mathcal{D}_{t+1} \cup \mathcal{S}_{t+1}^i$. To see that this map is invertible, it suffices to note that (i) $\{e_0, \mathbf{r}_0\} \subset \mathcal{D}_{t+1}$, and that (ii) $z_{t+1-\tau_{ij}}^j$ can be computed at node i given $d_{t+1-\tau_{ij}}^j \in \mathcal{D}_{t+1}$ and $s_{t+1-\tau_{ij}}^j \in \mathcal{S}_{t+1}^i$. Finally, it is clear that d_t is a deterministic state component, and that s_t is a zero mean Gaussian random variable (by the linearity of ν_t), from which we get $\mathbb{E}[d_t s_t^\top] = 0$. ■