



Design Of Real-Time Implementable Distributed Suboptimal Control: An LQR Perspective

Item Type	Article
Authors	Jaleel, Hassan;Shamma, Jeff S.
Citation	Jaleel H, Shamma JS (2017) Design Of Real-Time Implementable Distributed Suboptimal Control: An LQR Perspective. IEEE Transactions on Control of Network Systems: 1-1. Available: http://dx.doi.org/10.1109/TCNS.2017.2754362 .
Eprint version	Post-print
DOI	10.1109/TCNS.2017.2754362
Publisher	Institute of Electrical and Electronics Engineers (IEEE)
Journal	IEEE Transactions on Control of Network Systems
Rights	(c) 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.
Download date	2024-04-21 15:13:18
Link to Item	http://hdl.handle.net/10754/625514

Design Of Real-Time Implementable Distributed Suboptimal Control: An LQR Perspective

Hassan Jaleel and Jeff S. Shamma

Abstract—We propose a framework for multiagent systems in which the agents compute their control actions in real time, based on local information only. The novelty of the proposed framework is that the process of computing a suboptimal control action is divided into two phases: an offline phase and an online phase. In the offline phase, an approximate problem is formulated with a cost function that is close to the optimal cost in some sense and is distributed, i.e., the costs of non-neighboring nodes are not coupled. This phase is centralized and is completed before the deployment of the system. In the online phase, the approximate problem is solved in real time by implementing any efficient distributed optimization algorithm. To quantify the performance loss, we derive upper bounds for the maximum error between the optimal performance and the performance under the proposed framework. Finally, the proposed framework is applied to an example setup in which a team of mobile nodes is assigned the task of establishing a communication link between two base stations with minimum energy consumption. We show through simulations that the performance under the proposed framework is close to the optimal performance and the suboptimal policy can be efficiently implemented online.

I. INTRODUCTION

The goal of this paper is to present a real-time framework for multiagent systems to strike a balance between global performance and the cost of computing an optimal control action in a distributed manner. This cost typically includes the communication overhead of the control policy and the delay in computing an optimal control action. The performance of multiagent systems comprising mobile robots critically depends on these factors. The additional energy consumption because of the communication overhead of the control policy can significantly reduce the lifetime of these battery operated devices. Similarly, the delay in computing a feasible control action can have serious consequences and can even destabilize the system. Thus, for such systems, the utility of computing a feasible control action efficiently can be higher than computing an optimal control action with delay.

In this work, the focus is on designing local interaction laws for individual agents that are implementable in real time, and have limited communication overhead. In the proposed framework, we divide the process of designing a control policy into two phases: an offline phase and an online phase. The offline phase is executed before the deployment of the system. In this phase, an infinite horizon discounted LQR problem is formulated based on prior information of the system like agent's dynamics and information flow topology.

H. Jaleel and J.S. Shamma are with the King Abdullah University of Science and Technology (KAUST), Computer, Electrical and Mathematical Sciences and Engineering Division (CEMSE), Thuwal 23955-6900, Saudi Arabia. Email: hassan.jaleel@kaust.edu.sa, jeff.shamma@kaust.edu.sa. Research supported by funding from KAUST.

The advantage of formulating the problem as an infinite horizon discounted LQR problem is that the optimal cost can be expressed analytically. From the optimal cost structure, we derive an approximate cost function that decouples the cost of non-neighboring agents and formulate an approximate problem (see [1] and [2] for details on approximate dynamic programming). In the online phase, each agent solves its local version of the approximate problem in real time using some efficient distributed optimization algorithm.

To summarize, the main idea presented in this work is to shift the computational burden offline in formulating an approximate problem that can be solved efficiently online, and can provide stability and minimum performance guarantees. To quantify the performance loss, we derive novel bounds for maximum performance gap between the approximate solution and the optimal solution.

Distributed computation of optimal control action has received significant attention in the last decade (see e.g., [3], [4], [5], [29], and [32]). One popular approach is based on model predictive control (MPC) as presented in [12], [14], [15], [16], and [24]. In this approach, convergence to an optimal solution is guaranteed but at the expense of communication of entire trajectories among agents repeatedly. Moreover, strict constraints are imposed on the trajectories of the agents to limit their deviation from nominal trajectories. Consequently, the strategies based on MPC have significant communication overhead and are not always suitable for real-time applications. In [30], suboptimal solutions were proposed but no bounds were provided for the resulting performance. In [17], a suboptimal policy was proposed with limited number of communication iterations. Performance guarantees were provided but under strict constraints on problem structure.

Another approach for distributed optimal control was initially presented in [36] in which it was shown that for spatially invariant systems, LQR design for infinite dimensional system diagonalizes exactly as a parameterized family of finite dimensional LQR problems. In [37], the structure of the controller was fixed and it was shown that if the assumed structure is quadratic invariant, then the optimal stabilizing feedback matrix can be computed efficiently. Spatial and quadratic invariance can guarantee decentralized controllers with optimal performance but impose strict structural requirements on the systems.

In [31] and [32], sparsity inducing regularization terms were added to the cost function to design sparse controller architectures. Then optimal controller matrices with specified sparsity structures were computed by solving optimization problems. In [33], a similar approach was presented but the sparsity inducing regularization functions were restricted to be convex

and the optimal feedback matrix satisfying the desired sparsity structure could be computed by solving convex programs.

In the proposed framework, we do not impose any constraints on the sparsity structure of the controller as in previous works. Instead, we impose the desired sparsity constraints on the cost function to decouple the costs of non-neighboring agents. We start by formulating a problem as an infinite horizon discounted LQR problem. Since the optimal cost of LQR problems have known analytical expressions, the problem can be reformulated as an equivalent one stage lookahead problem with optimal cost to go as terminal cost. This terminal cost is a quadratic function of a positive definite matrix K , which is a solution to an algebraic Riccati equation, that couples the costs of all the agents in the network. For a system with fixed communication network, the matrix K can be computed offline before the system is deployed. We compute it offline and use it to derive an approximate cost to go function that decouples the cost of non-neighboring agents. Using this approximate cost function, we formulate an approximate dynamic program with one step look-ahead. Therefore, the original trajectory optimization problem over an infinite horizon is approximated by a parameter optimization problem. To solve this parameter optimization problem, distributed optimization algorithms exist that guarantee convergence to the centralized solution

Distributed optimization in multiagent system has been an active area of research for a long time (see e.g. [34], [28], [29], [20], [19], [18], and [35]). Subgradient based algorithms like [20] can guarantee convergence to an optimal solution by communication among neighbors only. These algorithms have slow convergence and have significant communication overhead. Decomposition based algorithms ([18] and [19]) or ADMM based algorithms ([28] and [34]) converge faster as compared to subgradient based algorithms. However, these algorithms require communication among the agents with coupled costs. Therefore, for optimal solutions, these algorithms require communication among all the agents in the network. In the proposed framework, the approximate cost of each agent is coupled to its neighbors only, and the decomposition or ADMM based algorithms can be implemented efficiently.

For any approximate or suboptimal solution, it is imperative to assess how much performance can be lost because of the approximation. For the proposed framework, we derive novel bounds on the maximum performance gap between the optimal solution and the approximate solution. Since our main application is mobile robots (ground or aerial vehicles), the state space is entire \mathbb{R}^{2N} or \mathbb{R}^{3N} , where N is the number of the agents in the network. Therefore, we define compact sets that are invariant under the proposed policies to ensure the effectiveness of the performance bounds.

Finally, we consider an example setup in which a collection of relay nodes is deployed to establish a communication link between two base stations for a long period of time. The task is to find an energy efficient distributed mobility strategy to move the relay nodes to optimal locations such that the total energy consumption is minimized. We apply our framework to this system and demonstrate via simulations that the resulting approximate policy can be efficiently implemented in real time, and its performance is almost the same as the optimal

policy.

A. Notation

We denote a graph by $G = (V, E)$ where $V = \{1, 2, \dots, N\}$ is the set of vertices and $E = \{(i, j) \mid i, j \in V\}$ is the set of edges. Graph G is undirected if its links are bidirectional ($(i, j) \in E$ iff $(j, i) \in E$). The neighborhood of node i is

$$\mathcal{N}_i = \{j \in V \mid (i, j) \in E\},$$

and the cardinality of this set is $|\mathcal{N}_i|$. A graph is connected if given any pair of nodes (i, j) , either $(i, j) \in E$ or there exist some intermediate nodes such that $\{(i, i_1), (i_1, i_2), \dots, (i_m, j)\} \in E$. The degree matrix of G is a diagonal matrix $\mathcal{D}(G)$ with the diagonal entries $\mathcal{D}_{ii}(G) = |\mathcal{N}_i|$. The adjacency matrix $\mathcal{A}(G)$ is

$$\mathcal{A}_{ij}(G) = \begin{cases} 1 & : j \in \mathcal{N}_i \\ 0 & : \text{otherwise} \end{cases}$$

The graph Laplacian,

$$\mathcal{L}(G) = \mathcal{D}(G) - \mathcal{A}(G),$$

is a symmetric and positive semidefinite matrix with real and non-negative eigenvalues for undirected graphs. Moreover, for a connected graph, $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_N$.

Let $z_i(k) \in \mathbb{R}^n$ denotes the state of node i at time k . For concise notation, the locations of all the nodes at time k are stacked in vector $\mathbf{z}_k \in \mathbb{R}^{2N}$, i.e., $\mathbf{z}_k = [z_1(k)^T \dots z_N(k)^T]^T$. For $i \in \{1, 2, \dots, N\}$, $\mathbf{z}_{-i,k}$ is the vector of locations of all the neighbors of i , i.e., $\mathbf{z}_{-i,k} = [z_{i_1}(k)^T z_{i_2}(k)^T \dots z_{i_{|\mathcal{N}_i|}}(k)^T]^T$ where $i_1, \dots, i_{|\mathcal{N}_i|} \in \mathcal{N}_i$. For $z \in \mathbb{R}^2$, $\|z\|$ denotes the Euclidean norm. If $A \in \mathbb{R}^{N \times N}$ and $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ are its N eigenvalues in non-increasing order, then $\lambda_i(A) = \lambda_i$. The vector $\mathbf{1}_n$ denotes the vector $[1 \ 1 \dots 1]^T \in \mathbb{R}^n$, $I_n \in \mathbb{R}^{n \times n}$ denotes identity matrix, and $0_{n \times m}$ denotes a matrix of dimension $n \times m$ with all entries equal to 0. Boldface letters like \mathbf{z}_k denote collections of vectors, and their subscript represents time.

II. PROPOSED FRAMEWORK

A. System Setup

We consider a multiagent system in which the agents collaborate to achieve some desired global objective by communicating with each other. The communication network is represented by a graph $G(V, E)$ where V is the set of agents and E is the set of edges. An edge exists between nodes i and j if they can communicate with each other. We assume that the communication network is time invariant and is represented by a fixed graph. The neighborhood of agent i is \mathcal{N}_i , which is the set of all the agents it can communicate with. The communication network topology is algebraically represented by graph Laplacian.

Since the state of each agent belongs to \mathbb{R}^n , we define extended Laplacian as

$$\mathcal{L}' = (\mathcal{L} \otimes I_n).$$

Next, we present the steps involved in the proposed framework followed by a detailed discussion on each step.

B. Framework Description

In the proposed framework, we divide the process of implementing a control policy into two phases, an offline phase and an online phase. The offline phase is a design phase, which is centralized and completed before the deployment of the system. In this phase, the objective is to formulate an approximate problem such that

- 1) the cost of each agent only depends on its local information,
- 2) the resulting system dynamics are stable, and
- 3) the approximate cost function is close to optimal cost function in some sense.

The online phase is an implementation phase in which any efficient distributed optimization algorithm is implemented to solve the approximate problem online. The steps involved in the proposed framework as follows.

Step 1: Formulate the problem as an infinite horizon discounted LQR problem.

$$\lim_{T \rightarrow \infty} \min_{\{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{T-1}\}} \sum_{k=0}^{T-1} \alpha^k (\mathbf{z}_k^T Q \mathbf{z}_k + \mathbf{u}_k^T R \mathbf{u}_k) + \alpha^T \mathbf{z}_T^T Q \mathbf{z}_T$$

$$\text{s.t. } \mathbf{z}_{k+1} = A \mathbf{z}_k + B \mathbf{u}_k, \quad (\mathcal{P})$$

Here, $\mathbf{z}_k \in \mathbb{R}^{nN}$ is the state vector at time k , $A \in \mathbb{R}^{n(N \times N)}$ is the system matrix, and $B \in \mathbb{R}^{n(N \times q)}$ is the input matrix, where q is the number of inputs. In the cost function, Q is some positive semidefinite (PSD) matrix that satisfies the constraints of the communication network, i.e.,

$$Q_{ij} = 0 \text{ if } \mathcal{L}'_{ij} = 0.$$

R is a diagonal and positive definite (PD) matrix of appropriate dimensions. To ensure that the closed loop system is stable we assume that A and B are such that (Q, A) is observable and (A, B) is controllable. The parameter $\alpha \in (0, 1)$ is a discount factor that exponentially reduces the weight of future states in the cost function.

Step 2: Using the principle of optimality, formulate the equivalent one stage look ahead problem with optimal cost to go.

$$\boldsymbol{\mu}^*(\mathbf{z}) = \arg \min_{\mathbf{u}} (g(\mathbf{z}, \mathbf{u}) + \alpha J^*(\mathbf{z}^+)),$$

$$\text{s.t. } \mathbf{z}^+ = A \mathbf{z} + B \mathbf{u}. \quad (\mathcal{P})$$

where

$$g(\mathbf{z}, \mathbf{u}) = \mathbf{z}^T Q \mathbf{z} + \mathbf{u}^T R \mathbf{u},$$

$$J^*(\mathbf{z}) = \mathbf{z}^T K \mathbf{z}.$$

In the above formulation, $J^*(\cdot)$ is the optimal cost to go and is a quadratic function of the matrix K , which is the positive definite solution of the Riccati equation

$$K = \alpha A^T K A - \alpha^2 A^T K B (\alpha B^T K B + R)^{-1} B^T K A + Q.$$

For infinite horizon discounted LQR problem, the optimal policy is

$$\boldsymbol{\mu}^*(\mathbf{z}_k) = -L \mathbf{z}_k,$$

$$L = \alpha (\alpha B^T K B + R)^{-1} B^T K A. \quad (1)$$

Step 3: Project the constraints of the communication network on K and find a symmetric and PSD matrix $K_{\mathcal{L}}$ that satisfies the constraints of the communication network, i.e.,

$$(K_{\mathcal{L}})_{ij} = \begin{cases} 0 & : \mathcal{L}'_{ij} = 0 \\ * & : \text{otherwise} \end{cases} \quad (2)$$

where \mathcal{L}' is the extended Laplacian matrix and “*” represents a free entry that can have any value. We will have a detailed discussion in the next section on selecting the free variables.

Step 4: Formulate an approximate optimization problem

$$\hat{\boldsymbol{\mu}}(\mathbf{z}) = \arg \min_{\mathbf{u}} \left(\mathbf{z}^T Q \mathbf{z} + \mathbf{u}^T R \mathbf{u} + \alpha \tilde{J}(\mathbf{z}^+) \right),$$

$$\text{s.t. } \mathbf{z}^+ = A \mathbf{z} + B \mathbf{u}, \quad (\mathcal{P}1)$$

where $\tilde{J}(\cdot)$ is the approximate cost to go and is defined as

$$\tilde{J}(\mathbf{z}) = \mathbf{z}^T K_{\mathcal{L}} \mathbf{z}. \quad (3)$$

The resulting greedy policy will be

$$\hat{\boldsymbol{\mu}}(\mathbf{z}) = -\hat{L} \mathbf{z},$$

$$\hat{L} = \alpha (\alpha B^T K_{\mathcal{L}} B + R)^{-1} B^T K_{\mathcal{L}} A. \quad (4)$$

Thus, to ensure stability, $(A - B \hat{L})$ should be a stable matrix.

Step 5: Solve the approximate problem $\mathcal{P}1$ online by implementing some efficient distributed optimization algorithm.

C. Discussion

In the proposed framework, the offline phase consists of Steps 1 to 4, in which an approximate problem with one step look-ahead is formulated. The online phase comprises Step 5, in which a distributed optimization algorithm is implemented.

In Step 1, LQR paradigm is selected because of the fact that a variety of problems can either be formulated directly or can be well approximated as LQR problems. The advantage of using infinite horizon is that it yields analytical expression for the optimal cost to go, which assists in the equivalent one step formulation of the problem. The one step formulation of the problem simplifies the approximation process and the comparison of the suboptimal performance with the optimal performance. However, if the optimal cost does not have an analytical expression, the ideas presented in this work can still be applied by finding functions that lower bound the optimal cost function, as presented in [26] and [27].

Finally, the discounted problem formulation aids in approximating finite horizon problems as infinite horizon problems. The discount factor $\alpha \in (0, 1)$ serves two purposes in this problem formulation.

- 1) It ensures that the total cost of the infinite horizon problem remains bounded.
- 2) It enables us to approximate a finite horizon problem as an infinite horizon problem.

This parameter can be adjusted based on system simulations and the structure of the system and the cost function.

In Step 2, after formulating the one step problem with optimal cost to go, the problem can still be solved using distributed optimization algorithms. However, the optimal matrix K does not guarantee to satisfy the communication network constraints. In fact, for most of the applications, all the entries

of K are non-zero implying that the cost of all the agents are coupled. Therefore, in decomposition based algorithms like dual decomposition or primal dual decomposition, or in ADMM based algorithms, all the agents will have to communicate with each other. Consensus based algorithms can still be used because they only require communication among neighbors. However, they have slow convergence, which severely effects the real-time requirements of the system and results in significant energy overhead.

In Steps 3 & 4, necessary conditions are provided for the approximate cost to go matrix. The constraint of positive semi-definiteness is standard for the terminal cost of LQR problems. The imposition of communication network constraints on the optimal matrix K decouples the cost of an agent from non-neighboring nodes and allows us to use fast distributed optimization algorithms. Additional constraints can also be imposed on the approximate cost function depending on an application.

The key question is how to compute the approximate cost function. This is an active area of research in approximate dynamic programming and various techniques for computing suboptimal policies exist in the literature (see [1], [2] and the references therein for details). One approach is presented recently in [27] in which approximate cost function is restricted to a finite-dimensional subspace, i.e.,

$$\tilde{J}(\cdot) = \sum_{l=1}^p \xi^{(l)} K^{(l)},$$

where $\xi^{(l)}$ are the coefficients and $K^{(l)}$ are the basis functions. Then a convex optimization problem is solved to find optimal coefficients, which yields an approximate cost function that tightly lower bounds the optimal cost function.

For the example setup in Section IV, we will use a simple approach in which all the entries of K corresponding to non-neighboring agents are zeroed out and the remaining entries are retained, while maintaining the Laplacian structure of the matrix. We prove that this approximate cost function will result in stable system dynamics for a class of problems.

In Step 5, an efficient distributed optimization algorithm is implemented to solve the approximate problem online. The advantage of the approximate problem is that it allows the use of efficient optimization algorithms with communication among neighboring nodes only.

III. PERFORMANCE ANALYSIS

In this section, we analyze system performance and derive bounds for the maximum gap between the optimal policy and a suboptimal policy derived under the proposed framework. In Section III-B, we consider multiagent systems comprising identical and dynamically decoupled agents. For such systems we present detailed stability analysis and prove that the closed-loop system is stabilized by an approximate policy generated by following the proposed framework.

The performance bounds are derived under the assumption that the system dynamics are stable. The stability of the system dynamics ensures a bounded state space. Therefore, the performance analysis of the system can be restricted to a

bounded set \mathcal{S} that is invariant under the system dynamics. A set \mathcal{S} is invariant under policy μ if $\mathbf{z} \in \mathcal{S}$ implies that $\mathbf{z}^+ = \mathbf{z} + B\mu(\mathbf{z}) \in \mathcal{S}$. The max norm of a function J over a set \mathcal{S} is

$$\|J\|_{\mathcal{S}} = \max_{\mathbf{z} \in \mathcal{S}} |J(\mathbf{z})|.$$

Let $\mathcal{S}_{\hat{\mu}}$ be an invariant set under policy $\hat{\mu}$. Let \mathcal{S}^* be the minimum set such that

- 1) $\mathcal{S}_{\hat{\mu}} \subseteq \mathcal{S}^*$, and
- 2) $\mathbf{z}_0 \in \mathcal{S}_{\hat{\mu}}$ implies that $\mathbf{z}_{k+1} = \mathbf{z}_k + B\hat{\mu}^*(\mathbf{z}_k) \in \mathcal{S}^*$ for all $k > 0$.

For any function $J : \mathcal{S}^* \rightarrow \mathbb{R}$, the mappings T and $T_{\hat{\mu}}$ are such that $(TJ) : \mathcal{S}_{\hat{\mu}} \rightarrow \mathbb{R}$ and $(T_{\hat{\mu}}J) : \mathcal{S}_{\hat{\mu}} \rightarrow \mathbb{R}$, and are defined as

$$\begin{aligned} (TJ)(\mathbf{z}) &= \min_{\mathbf{u}} [g(\mathbf{z}, \mathbf{u}) + \alpha J(A\mathbf{z} + B\mathbf{u})], \\ (T_{\hat{\mu}}J)(\mathbf{z}) &= g(\mathbf{z}, \hat{\mu}(\mathbf{z})) + \alpha J(A\mathbf{z} + B\hat{\mu}(\mathbf{z})). \end{aligned} \quad (5)$$

Let T^k and $T_{\hat{\mu}}^k$ be the composition of the mappings T and $T_{\hat{\mu}}$ with themselves k times respectively, i.e.,

$$\begin{aligned} T^k J &= T(T^{k-1} J), \\ T_{\hat{\mu}}^k J &= T_{\hat{\mu}}(T_{\hat{\mu}}^{k-1} J). \end{aligned}$$

For a stationary policy $\hat{\mu}$, the associated cost to go function $J^{\hat{\mu}}$ is

$$J^{\hat{\mu}} = \lim_{k \rightarrow \infty} T_{\hat{\mu}}^k J.$$

It has been proved in [2] that the optimal cost to go J^* satisfies Bellman equation, i.e.,

$$J^* = TJ^*.$$

Similarly, for a stable policy $\hat{\mu}$, $J^{\hat{\mu}} = T_{\hat{\mu}} J^{\hat{\mu}}$. A mapping T is a contraction mapping if there exists a scalar $\beta < 1$ such that

$$\|TJ - T\bar{J}\|_{\mathcal{S}} \leq \beta \|J - \bar{J}\|_{\mathcal{S}}.$$

The monotonicity lemma (Lem. 2.1 in [2]) states that for any two functions J and \bar{J} defined on \mathcal{S}^* such that

$$J(\mathbf{z}) \leq \bar{J}(\mathbf{z}) \quad \text{for all } \mathbf{z} \in \mathcal{S}^*,$$

the following inequalities hold:

$$\begin{aligned} (TJ)(\mathbf{z}) &\leq (T\bar{J})(\mathbf{z}) \quad \text{for all } \mathbf{z} \in \mathcal{S}_{\hat{\mu}}, \\ (T_{\hat{\mu}}J)(\mathbf{z}) &\leq (T_{\hat{\mu}}\bar{J})(\mathbf{z}) \quad \text{for all } \mathbf{z} \in \mathcal{S}_{\hat{\mu}}. \end{aligned}$$

Next we analyze the performance of any approximate policy by comparing it with the global optimal policy.

Theorem 3.1: *Let $K_{\mathcal{L}}$ be such that*

$$\tilde{J}(\mathbf{z}) = \mathbf{z}^T K_{\mathcal{L}} \mathbf{z}$$

is the approximate cost to go in problem (P1) and the resulting greedy policy $\hat{\mu}$ presented in Eq. (4) is stable. Let

$$\epsilon = \|\tilde{J} - J^*\|_{\mathcal{S}^*},$$

and $J^{\hat{\mu}} = \lim_{k \rightarrow \infty} T_{\hat{\mu}}^k \tilde{J}$. Then the maximum error between the global optimal solution and the approximate solution is

$$\|J^{\hat{\mu}} - J^*\|_{\mathcal{S}_{\hat{\mu}}} \leq \frac{2\alpha\epsilon}{1 - \alpha} \quad (6)$$

To prove Thm. 3.1, we need to prove the following proposition.

Proposition 1: For the sets $\mathcal{S}_{\hat{\mu}}$ and \mathcal{S}^ defined above,*

$$\|T_{\hat{\mu}}\tilde{J} - TJ^*\|_{\mathcal{S}_{\hat{\mu}}} \leq \alpha\|\tilde{J} - J^*\|_{\mathcal{S}^*}. \quad (7)$$

Proof: To prove Eq. (7), the first step is to recognize that $\hat{\mu}$ is the greedy policy with terminal cost \tilde{J} , so $(T_{\hat{\mu}}\tilde{J}) = (T\tilde{J})$. Let $c = \|\tilde{J} - J^*\|_{\mathcal{S}^*}$. Then

$$\begin{aligned} \tilde{J}(\mathbf{z}) - c &\leq J^*(\mathbf{z}) \leq \tilde{J}(\mathbf{z}) + c \quad \forall \mathbf{z} \in \mathcal{S}^* \\ \tilde{J}(\mathbf{z}) - c &\leq J^*(\mathbf{z}) \leq \tilde{J}(\mathbf{z}) + c \quad \forall \mathbf{z} \in \mathcal{S}_{\hat{\mu}} \end{aligned}$$

The second set of inequalities hold because $\mathcal{S}_{\hat{\mu}} \subseteq \mathcal{S}^*$. For

$$\begin{aligned} (T(\tilde{J} - c))(\mathbf{z}) &= \min_{\mathbf{u}} [g(\mathbf{z}, \mathbf{u}) + \alpha(\tilde{J}(\mathbf{z}^+) - c)] \\ &= (T\tilde{J})(\mathbf{z}) - \alpha c. \end{aligned}$$

Similarly $(T(\tilde{J} + c))(\mathbf{z}) = (T\tilde{J})(\mathbf{z}) + \alpha c$. Using the monotonicity property of T ,

$$T\tilde{J}(\mathbf{z}) - \alpha c \leq (TJ)^*(\mathbf{z}) \leq T\tilde{J}(\mathbf{z}) + \alpha c \quad \forall \mathbf{z} \in \mathcal{S}_{\hat{\mu}},$$

which implies that

$$\|T_{\hat{\mu}}\tilde{J} - TJ^*\|_{\mathcal{S}_{\hat{\mu}}} \leq \alpha c = \alpha\|\tilde{J} - J^*\|_{\mathcal{S}^*}$$

Next we present the proof of Thm. 3.1. ■

Proof: The proof of this result is based on the properties of the mappings defined in Eq. (5).

$$\begin{aligned} \|J^{\hat{\mu}} - J^*\|_{\mathcal{S}_{\hat{\mu}}} &= \|T_{\hat{\mu}}J^{\hat{\mu}} - J^*\|_{\mathcal{S}_{\hat{\mu}}} \\ &\stackrel{(i)}{\leq} \|T_{\hat{\mu}}J^{\hat{\mu}} - T_{\hat{\mu}}\tilde{J}\|_{\mathcal{S}_{\hat{\mu}}} + \|T_{\hat{\mu}}\tilde{J} - TJ^*\|_{\mathcal{S}_{\hat{\mu}}} \\ &\stackrel{(ii)}{\leq} \alpha\|J^{\hat{\mu}} - \tilde{J}\|_{\mathcal{S}_{\hat{\mu}}} + \alpha\|\tilde{J} - J^*\|_{\mathcal{S}^*} \\ &\leq \alpha\|J^{\hat{\mu}} - J^*\|_{\mathcal{S}_{\hat{\mu}}} + \alpha\|J^* - \tilde{J}\|_{\mathcal{S}_{\hat{\mu}}} \\ &\quad + \alpha\|\tilde{J} - J^*\|_{\mathcal{S}^*} \\ &\stackrel{(iii)}{\leq} \alpha\|J^{\hat{\mu}} - J^*\|_{\mathcal{S}_{\hat{\mu}}} + 2\alpha\epsilon, \end{aligned}$$

which concludes the proof. ■

In the above proof, (i) is based on triangular inequality, (ii) is based on the result of Prop. 1, and (iii) uses the fact that $\mathcal{S}_{\hat{\mu}} \subseteq \mathcal{S}^*$.

A. Performance Analysis with Restricted Overhead Policy

Thm. 3.1 provides an upper bound on the performance gap between the optimal policy μ^* and the approximate policy $\hat{\mu}$. Although distributed optimization algorithms are guaranteed to converge to the corresponding centralized solution, the convergence results are asymptotic and may require large number of iterations. For real-time applications, computing a control action quickly is more critical than the optimality of that action. Therefore, depending on system bandwidth, the maximum data rates of the sensing and communication modules, and the available energy budget, the number of iterations for the distributed optimization algorithms must be restricted to some suitable value ‘‘iter’’.

If the number of iterations are restricted too much, the system can become unstable even if the approximate policy is proved to be stable. Therefore, in addition to system constraints, it should be ensured via simulations and experimentation that the selected number of iterations maintain system stability. Limiting the number of iterations can introduce additional performance loss that was not included in Thm. 3.1. In the next result, we will explicitly include this performance loss in our analysis.

Let ‘‘iter’’ be the maximum number of iterations for distributed optimization algorithm and $\hat{\mu}_{\text{iter}}$ be the policy with restricted communication iterations. We assume that iter is selected such that $\hat{\mu}_{\text{iter}}$ is a stable policy. For policy $\hat{\mu}_{\text{iter}}$, the associated cost to go function $J^{\hat{\mu}_{\text{iter}}}$ is

$$J^{\hat{\mu}_{\text{iter}}} = \lim_{k \rightarrow \infty} T_{\hat{\mu}_{\text{iter}}}^k J.$$

Following the same procedure as in the previous section, we define three sets to bound the state space. Let $\mathcal{S}_{\hat{\mu}_{\text{iter}}}$ be an invariant set under policy $\hat{\mu}_{\text{iter}}$. Let $\mathcal{S}^{\hat{\mu}}$ be the minimum set such that

- $\mathcal{S}_{\hat{\mu}_{\text{iter}}} \subseteq \mathcal{S}^{\hat{\mu}}$, and
- $\mathbf{z}_0 \in \mathcal{S}_{\hat{\mu}_{\text{iter}}}$ implies that $\mathbf{z}_{k+1} = \mathbf{z}_k + B\hat{\mu}(\mathbf{z}_k) \in \mathcal{S}^{\hat{\mu}}$ for all $k > 0$.

We redefine the set \mathcal{S}^* as the minimum set such that

- $\mathcal{S}^{\hat{\mu}} \subseteq \mathcal{S}^*$, and
- $\mathbf{z}_0 \in \mathcal{S}^{\hat{\mu}}$ implies that $\mathbf{z}_{k+1} = \mathbf{z}_k + B\mu^*(\mathbf{z}_k) \in \mathcal{S}^*$ for all $k > 0$.

Theorem 3.2: Let $K_{\mathcal{L}}$ be such that

$$\tilde{J}(\mathbf{z}) = \mathbf{z}^T K_{\mathcal{L}} \mathbf{z}$$

is the approximate cost to go in problem (P1) and the resulting greedy policies $\hat{\mu}$, presented in Eq. (4), and $\hat{\mu}_{\text{iter}}$ are stable. Let

$$\epsilon = \|\tilde{J} - J^*\|_{\mathcal{S}^*} \text{ and } \epsilon_{\text{iter}} = \|J^{\hat{\mu}_{\text{iter}}} - J^{\hat{\mu}}\|_{\mathcal{S}_{\hat{\mu}}}$$

Then the maximum gap between the optimal performance and the performance under restricted communication policy $\hat{\mu}_{\text{iter}}$ is

$$\|J^{\hat{\mu}_{\text{iter}}} - J^*\|_{\mathcal{S}_{\hat{\mu}_{\text{iter}}}} \leq \epsilon_{\text{iter}} + \frac{2\alpha\epsilon}{1 - \alpha} \quad (8)$$

Proof: The proof is based on the result of Thm. 3.1.

$$\begin{aligned} \|J^{\hat{\mu}_{\text{iter}}} - J^*\|_{\mathcal{S}_{\hat{\mu}_{\text{iter}}}} &\leq \|J^{\hat{\mu}_{\text{iter}}} - J^{\hat{\mu}}\|_{\mathcal{S}_{\hat{\mu}_{\text{iter}}}} + \|J^{\hat{\mu}} - J^*\|_{\mathcal{S}_{\hat{\mu}_{\text{iter}}}} \\ &\leq \|J^{\hat{\mu}_{\text{iter}}} - J^{\hat{\mu}}\|_{\mathcal{S}_{\hat{\mu}}} + \|J^{\hat{\mu}} - J^*\|_{\mathcal{S}_{\hat{\mu}}} \\ &\leq \epsilon_{\text{iter}} + \frac{2\alpha\epsilon}{1 - \alpha} \end{aligned}$$

which concludes the proof. The first step is simply triangular inequality and the next inequality is based on the definition of ϵ_{iter} and Thm. 3.1. ■

One note regarding the limited communication policy $\hat{\mu}_{\text{iter}}$ is that it is not necessary to incur a performance loss in going from $\hat{\mu}$ to $\hat{\mu}_{\text{iter}}$. Although $\hat{\mu}_{\text{iter}}$ is a limited iteration version of $\hat{\mu}$, it can be considered as a different approximate policy and the system performance that results from $\hat{\mu}_{\text{iter}}$ may be better than that from $\hat{\mu}$. However, there cannot be any guarantees, therefore in computing the performance bounds, we have considered the worst case scenario.

The performance bounds in Thms. 3.1 and 3.2 are in terms of the discount factor α and the difference in the optimal and the approximate cost functions ϵ . These bounds can be conservative especially if α is close to one as will be observed in Section V. This is true in general for bounds on worst case performance especially if the state space is large. However, it is important to highlight that these are upper bounds for an entire class of approximate cost functions that have the same ϵ .

B. Stability Analysis

In this section, we present stability analysis for a typical multiagent system in which there are N_m identical and dynamically decoupled mobile agents and N_s stationary agents, as shown in Fig. 1. The agents are indexed such that N_s stationary agents are stacked after N_m mobile agents. This representation results in a block structure that simplifies the analysis. The stationary agents in this architecture can serve multiple purposes. These can be anchor nodes with more storage and higher computational capabilities, which can receive and store information from mobile agents and perform complex operations that cannot be completed by mobile agents particularly in real time. The stationary agents can also correspond to base stations that need to communication with each other and the mobile agents may act as relay nodes to facilitate the communication.

Lemma 3.1: *Consider the following one step look-ahead problem*

$$\begin{aligned} \hat{\mu}(\mathbf{z}) &= \arg \min_{\mathbf{u}} \left(\mathbf{z}^T Q \mathbf{z} + \mathbf{u}^T R \mathbf{u} + \alpha \tilde{J}(\mathbf{z}^+) \right), \\ \text{s.t. } \mathbf{z}^+ &= A \mathbf{z} + B \mathbf{u}. \end{aligned}$$

where Q is as defined in (\mathcal{P}) and approximate cost to go \tilde{J} is

$$\tilde{J}(\mathbf{z}) = \mathbf{z}^T H \mathbf{z}.$$

If H is symmetric and positive semidefinite and

$$A = I_{n(N_m+N_s)}, \quad R = \kappa_M I_{nN_m}, \quad \text{and } B = \begin{bmatrix} I_{nN_m} \\ 0_{n(N_s \times N_m)} \end{bmatrix}$$

with $\kappa_M > 0$, then the closed loop system matrix

$$(I_{n(N_m+N_s)} - B\hat{L})$$

under the greedy policy $\hat{\mu}(\mathbf{z}) = -\hat{L}\mathbf{z}$ is marginally stable, where

$$\hat{L} = \alpha(\alpha B^T H B + R)^{-1} B^T H.$$

Proof: Let $\tilde{H} = B\hat{L}$ and λ_i be the i^{th} eigenvalue of $(I_{n(N_m+N_s)} - \tilde{H})$ such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n(N_m+N_s)}$. To prove that the system is marginally stable, we need to show that $|\lambda_i| \leq 1$ for all $i = \{1, 2, \dots, n(N_m + N_s)\}$ and $(I_{n(N_m+N_s)} - \tilde{H})$ has $n(N_m + N_s)$ independent eigenvectors. For analysis purposes, we partition H as follows:

$$H = \begin{bmatrix} H_f & H_{fl} \\ H_{lf} & H_{ll} \end{bmatrix}, \quad (9)$$

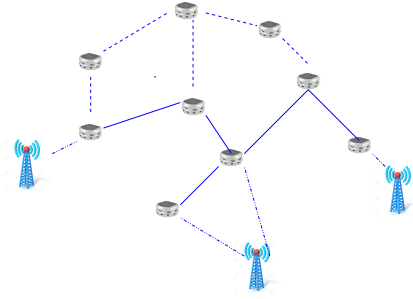


Fig. 1. A typical architecture for a multiagent system.

where

$$\begin{aligned} H &\in \mathbb{R}^{n(N_m+N_s) \times n(N_m+N_s)} & H_f &\in \mathbb{R}^{n(N_m \times N_m)} \\ H_{lf} &\in \mathbb{R}^{n(N_s \times N_m)} & H_{ll} &\in \mathbb{R}^{n(N_s \times N_s)}. \end{aligned}$$

Since H is symmetric, $H_{lf} = H_{fl}^T$. Thus,

$$\tilde{H} = \begin{bmatrix} \alpha(\alpha H_f + R)^{-1} H_f & \alpha(\alpha H_f + R)^{-1} H_{fl} \\ 0_{n(N_s \times N_m)} & 0_{n(N_s \times N_s)} \end{bmatrix}.$$

where $H_f = B^T H B$. Using the properties of block matrices, the eigenvalues of \tilde{H} are the eigenvalues of $\alpha(\alpha H_f + R)^{-1} H_f$ and $0_{n(N_s \times N_s)}$. Therefore, $0_{n(N_s \times N_s)}$ contributes nN_s zero eigenvalues and $\alpha(\alpha H_f + R)^{-1} H_f$ contributes nN_m eigenvalues. Next we will show that these nN_m eigenvalues are real, positive and less than one.

Since H is symmetric and positive semidefinite, $H_f = B^T H B$ is also symmetric and positive semidefinite. Therefore, H_f has real and non-negative eigenvalues and nN_m independent eigenvectors. To show that the eigenvalues are also less than one, let λ_i be the i^{th} eigenvalue of H_f . Then $\frac{1}{\alpha\lambda_i + \kappa_M}$ and $\frac{\alpha\lambda_i}{\alpha\lambda_i + \kappa_M}$ are the corresponding eigenvalues of $(\alpha H_f + R)^{-1}$ and $(\alpha H_f + R)^{-1} \alpha H_f$ respectively. Here we have used the fact that if two matrices P and Q have the same set of eigenvectors, then they commute and $\lambda_i(PQ) = \lambda_i(P)\lambda_i(Q)$. Therefore, the eigenvalues of $(\alpha H_f + R)^{-1} \alpha H_f$ are always less than or equal to one for $\kappa_M > 0$. This ensures that the eigenvalues of $I_{nN_m} - \alpha(H_f + R)^{-1} \alpha H_f$ are also less than or equal to one. Since the last nN_s rows of \tilde{H} are zero, $I_{nN_s} - \tilde{H}_{ll} = I_{nN_s}$ and there will be nN_s more independent eigenvectors. This implies that the eigenvalues of $I_{n(N_m+N_s)} - \tilde{H}$ are always less than or equal to one with independent eigenvectors, which concludes the proof. ■

The marginal stability is because of the existence of multiple zero eigenvalues of the closed loop system matrix. This implies that the system will converge to the null space of the matrix $(I_{n(N_m+N_s)} - B\hat{L})$. Thus, in Lem. 3.1, we provided sufficient conditions in terms of the structure of the matrices A , B , and R , for the stability of the closed loop system under the proposed framework.

IV. COOPTIMIZATION OF MOBILITY AND COMMUNICATION

Consider two base stations \mathcal{B}_1 and \mathcal{B}_2 separated by a distance d . Without loss of generality we can assume that the base station \mathcal{B}_1 is located at the origin. The objective is to

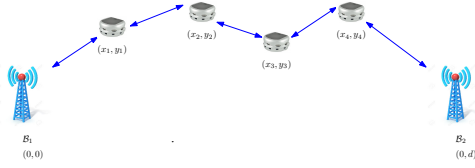


Fig. 2. Illustration of the communication network. B_1 and B_2 are the base stations and N relay nodes are located at (x_i, y_i) for all $i \in \{1, 2, \dots, N\}$.

establish an uninterrupted communication link between these base stations for a time interval of length T with minimum energy consumption. One possible solution to this problem is to place multiple relay nodes at some optimal locations to minimize the overall communication energy consumption.

The problem of optimal placement of relay nodes to cooperate mobility and communication has been actively studied over the recent years (see e.g., [6], [7], [8], [9], and [13] etc). In [6], synchronous and asynchronous distributed algorithms were presented for steering relay nodes to the optimal locations for establishing a communication link between two base stations. In [7], a similar problem was considered, but the proposed solutions were based on heuristics and no optimality guarantees were provided. In [8] and [9] this problem was addressed for a single agent under realistic communication and channel models. In [10] and [11], both mobility and communication costs were included but a centralized setup was considered.

From [6], the power required to successfully transmit q bits of data over a distance d is

$$P_{\text{comm}} = \kappa_C d^2 + b,$$

where $\kappa_C = \eta q$, η is the proportionality constant and b reflects the additional power consumed by transmitter/receiver circuitry. In this work we are interested in communication power only, so assume $b = 0$.

The energy required to establish the communication link between the two base stations will be high for large values of d and T . One solution is to use N relay nodes as shown in Fig. 2. Depending on the value of N and the deployment locations of the nodes, the overall energy consumption can be significantly reduced. For minimum energy consumption, the relay nodes must be evenly spaced on the straight line between the two base stations [6]. However, we are interested in a scenario in which the initial deployment of the nodes is random, and the relay nodes need to estimate and move to the optimal locations in a decentralized manner while maintaining the communication link. This redeployment towards the optimal locations will have its own cost that must be accounted for.

Let $z_i(k)$ denotes the location of relay node i at time k for $i \in \{1, 2, \dots, N\}$, and $z_{N+1}(k)$ and $z_{N+2}(k)$ denote the locations of the base stations B_1 and B_2 respectively. Because the base stations remain stationary, $z_{N+1}(k)$ and $z_{N+2}(k)$ are constant values. This network of the relay nodes and the base stations is represented by a graph $G_c = (V, E)$ in which the vertex set V consists of N relay nodes $\{1, 2, \dots, N\}$ and two base stations $\{N+1, N+2\}$.

To find the optimal locations of the relay nodes, we assume that the system satisfies the following properties.

Assumptions:

- 1) The relay nodes are mobile nodes with single integrator dynamics

$$z_i(k+1) = z_i(k) + u_i(k),$$

where $u_i(k) \in \mathbb{R}^2$ is the input at time k .

- 2) The communication network is a line graph, and it remains fixed, i.e., $\mathcal{L} \in \mathbb{R}^{(N+2) \times (N+2)}$ is a constant matrix.

Although the communication link is established initially, the overall energy consumption can be minimized if the relay nodes can somehow move to the optimal locations in an efficient manner. However, there is no leader or centralized authority that has the knowledge of the optimal locations, so the nodes need to figure it out locally based on communication with their neighbors. To ensure that the overall energy consumption is minimized, it is imperative to incorporate the cost of mobility in the system model because mobility is orders of magnitude expensive than communication. From [22], mobility power consumption can be approximated to be a linear function of speed, i.e.,

$$P_{\text{mob}} = \psi(\|u(k)\|),$$

where $u(k)$ is the robot's velocity at time k . We assume that mobility cost of a node is proportional to the square of its speed, i.e., $\psi(\|u(k)\|) = \kappa_M \|u(k)\|^2$, where κ_M is a mobility constant. This choice of mobility model is valid for ground vehicles that use DC motors and operate at low speeds. The advantage of this model is that it will help in obtaining an analytical solution to the optimization problem.

To solve this problem, we will follow the five steps in the proposed framework.

A. Application of the Proposed Framework

The first step is to formulate an infinite horizon discounted LQR problem. The total cost that is to be minimized is the sum of the mobility cost J_{mob} and the communication cost J_{comm} of the system. Thus, the stage cost in problem \mathcal{P} is

$$g(\mathbf{z}_k, \mathbf{u}_k) = J_{\text{comm}}(\mathbf{z}(k)) + J_{\text{mob}}(\mathbf{u}_k)$$

$$J_{\text{comm}}(\mathbf{z}_k) = \frac{1}{2} \left(\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \kappa_C \|z_i(k) - z_j(k)\|^2 \right),$$

$$J_{\text{mob}}(\mathbf{u}_k) = \sum_{i=1}^N \kappa_M \|u_i(k)\|^2,$$

and the terminal cost is

$$J_T(\mathbf{z}_T) = \frac{1}{2} \left(\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \kappa_C \|z_i(T) - z_j(T)\|^2 \right).$$

In this example, the dimension of state space, the number of mobile agents, and the number of stationary agents is $n = 2$, $N_m = N$, and $N_s = 2$ respectively.

The second step is to formulate the equivalent one stage lookahead problem with stationary optimal cost to go as its

terminal cost. Next we will show that for this problem and a class of similar problems, the matrix K , which is the positive definite solution of the Riccati equation, is a non-singular M matrix with all the entries equal to non-zero.

Definition 4.1: A matrix D is a positive matrix ($D > 0$) if all of its entries are positive. It is a non-negative matrix ($D \geq 0$) if all of its entries are non-negative.

Definition 4.2: An $N \times N$ matrix Z is an M matrix if it can be written as

$$Z = sI_N - D, \quad s > 0, \quad D \geq 0, \quad \rho(D) \leq s,$$

Proposition 2: If $Q \in \mathbb{R}^{2(N+2) \times 2(N+2)}$ is a symmetric M matrix, $A = I_{2(N+2)}$, $B = \begin{bmatrix} I_{2N} \\ 0_{2(N_s \times N)} \end{bmatrix}$, and $R = \kappa_M I_{2N}$, then the positive definite solution of the Riccati equation

$$K = \alpha K - \alpha^2 K B (\alpha B^T K B + R)^{-1} B^T K + Q$$

is also an M matrix. Moreover, if Q is a Laplacian matrix of a connected graph then all the entries of K are non-zero.

Proof: The proof is presented in the appendix. ■

For this example $Q = \mathcal{L}'$, which implies that it is an M matrix and Prop. 2 is applicable. Thus, the optimal policy is centralized and each agent requires state information of all the other agents to compute the optimal cost to go.

The next step is to compute an approximate cost to go matrix $K_{\mathcal{L}}$. As discussed in the previous section, this is an active area of research and various criteria can be used. For this example, we define $K_{\mathcal{L}}$ as follows:

$$(K_{\mathcal{L}})_{ij} = \begin{cases} K_{ij} & : i \neq j \text{ and } \mathcal{L}'_{ij} \neq 0 \\ K_{ii} - \sum_{p \notin \mathcal{N}_i} K_{ip} & : i = j \\ 0 & : \text{otherwise} \end{cases} \quad (10)$$

Here $K_{\mathcal{L}}$ is a projection of K on the sparsity structure of \mathcal{L}' and the diagonal entries are adjusted to ensure that the resulting matrix is graph Laplacian. Since K is an M matrix as shown in Prop. 2, the matrix $K_{\mathcal{L}}$ is guaranteed to be a symmetric and PSD matrix and it satisfies the communication constraints of the network by construction. Therefore, $K_{\mathcal{L}}$ is a feasible candidate for an approximate cost to go matrix. The proposed approximate cost to go function is

$$\tilde{J}(\mathbf{z}) = \mathbf{z}^T K_{\mathcal{L}} \mathbf{z},$$

and the approximate problem is

$$\hat{\boldsymbol{\mu}}(\mathbf{z}_k) = \arg \min_{\mathbf{u}} g(\mathbf{z}_k, \mathbf{u}) + \alpha \tilde{J}(\mathbf{z}_k + B\mathbf{u}),$$

with dynamics and constraints specified in \mathcal{P} . Based on Lem. 3.1, the proposed policy will result in stable dynamics because $K_{\mathcal{L}}$ is symmetric and PSD and the matrices A , B , and R have the right structure.

We want to reemphasize that this is one choice of approximate cost function. We can always rely on sophisticated methods for finding approximate cost functions that are optimal in some sense. However, this paper is not about such methods. Therefore, we are using a simple approach to demonstrate the steps involved in the proposed framework.

The final step is efficient online implementation of the distributed problem. Since the cost of each agent in Eq. 4.2 only depends on the states of its immediate neighbors, we can use dual decomposition algorithm. In dual decomposition, the coupling between the costs of neighboring agents is removed by introducing local estimates of coupling variables. Let u_j^i be agent i 's estimate of the input of agent j and $u^i = [u_{i_1}^i \quad u_{i_2}^i \quad \dots \quad u_{i_{|\mathcal{N}_i|}}^i]$ where $i_1, \dots, i_{|\mathcal{N}_i|} \in \mathcal{N}_i$. Then the lagrange dual function of each agent is

$$L_i(\lambda_i) = \min_{u^i} \frac{\kappa_C}{2} \sum_{j \in \mathcal{N}_i} \|z_i - z_j\|^2 + \kappa_M \|u^i\|^2 + \frac{(K_{\mathcal{L}})_{ij}}{2} \sum_{j \in \mathcal{N}_i} \|((z_i + u^i) - (z_j + u_j^i))\|^2 + \sum_{j \in \mathcal{N}_i} \lambda_{ij}^T u_j^i - \sum_{j \in \mathcal{N}_i} \lambda_{ji}^T u_j^i,$$

where λ_{ij} is the dual variable associated with the consistency constraint $u_j^i = u_j^j$ and

$$\lambda_i = [\lambda_{i,i_1}^T \quad \lambda_{i,i_2}^T \quad \dots \quad \lambda_{i,i_{|\mathcal{N}_i|}}^T \quad \lambda_{i_1,i}^T \quad \lambda_{i_2,i}^T \quad \dots \quad \lambda_{i_{|\mathcal{N}_i|},i}^T]^T.$$

Each agent can compute its local dual function L_i as a function of its dual variables, independent of all the other agents. The dual function of the problem is the sum of all the local dual functions, i.e.,

$$L(\lambda) = \sum_{i=1}^N L_i(\lambda_i),$$

and the dual problem is the maximization of the dual function.

$$L^* := \max_{\lambda} L(\lambda).$$

The dual problem can be solved via gradient method in which each dual variable is updated as follows.

$$\lambda_{ij}(l+1) = \lambda_{ij}(l) + \gamma(u_j^i(l) - u_i^j(l)).$$

At each time k , agent i starts by assuming some initial values of the dual variables in vector λ_i and computes L_i . Then, it communicates with all agents $j \in \mathcal{N}_i$ to receive their estimates of its control input u_j^i and executes the update step. At the same time, it transmits its estimates u_i^j to all $j \in \mathcal{N}_i$. These two steps are repeated until the gradient method does not change the dual variables significantly. In Section V, we will present the results of dual decomposition algorithm with limited number of update steps for the dual ascent problem. There are some results in the literature like [17] that relate the number of iterations of the dual decomposition algorithm to the resulting performance. These results can be used to estimate the suitable number of iterations for the algorithm and the performance loss that can be expected.

V. SIMULATION

To verify the performance of the proposed framework, we simulated the above system with two base stations and six relay nodes. The details of the simulated system are as follows. The two base stations are separated by a distance $d = 100$ and are located at $[0 \ 0]^T$ and $[0 \ 100]^T$. The number of relay nodes is $N = 6$ and their initial deployment locations are

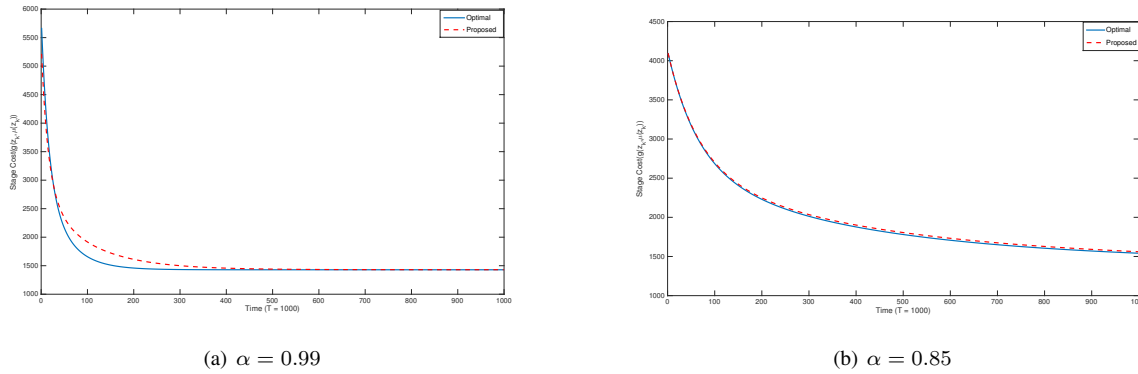


Fig. 3. Comparison of the stage costs of the proposed policies and the optimal policy.

$\mathbf{z}_0 = [\mathbf{x}^T \ \mathbf{y}^T]^T$, where $\mathbf{x} = [5 \ 12 \ 26 \ 30 \ 33 \ 50 \ 0 \ 100]^T$ and $\mathbf{y} = [20 \ 22 \ 15 \ 17 \ 25 \ 28 \ 0 \ 0]^T$. The last two entries in \mathbf{x} and \mathbf{y} are the locations of the base stations. The values of the system parameters used in the simulations are $T = 1000$, $\kappa_C = 1$, $\kappa_M = 1000$. We simulated the system for $\alpha = 0.99$ and $\alpha = 0.85$ to demonstrate that we can model systems with finite time horizon by varying the value of α . The dual decomposition algorithm was implemented with number of iterations $\text{iter} = 2$. The step size for the dual problem was $\gamma = 24$ for $\alpha = 0.99$ and $\gamma = 6$ for $\alpha = 0.85$ and the dual variables were always initialized with zeros.

The simulation results are presented in Figs. 3 and 4. In Figs. 3(a) and 3(b), a comparison of the costs incurred by the system under the optimal and the proposed policies is presented for $\alpha = 0.99$ and $\alpha = 0.85$ respectively. In both the figures, the solid line is the stage cost under the optimal policy $g(\mathbf{z}_k, \boldsymbol{\mu}^*(\mathbf{z}_k))$ and the dashed line is the stage cost under the suboptimal policy $g(\mathbf{z}_k, \hat{\boldsymbol{\mu}}_{\text{iter}}(\mathbf{z}_k))$ for all $k \in \{1, 2, \dots, T\}$. From these figures, it is evident that under the suboptimal policy the system performance is almost the same as the optimal one for both the cases even with only two iterations of the distributed optimization algorithm. Thus, the proposed policy is suitable for online distributed implementation.

In Figs. 4(a)–4(d), the trajectories of the relay nodes for the optimal and the proposed policy are presented. When $\alpha = 0.99$, the system behavior resembles that of infinite horizon problem and the final locations of the relay nodes lie on the straight line between the two base stations equidistant from each other as shown in Figs. 4(a) and 4(b). However for $\alpha = 0.85$, the system behaves like a finite horizon problem since the relay nodes stop before reaching the straight line as shown in Figs. 4(c) and 4(d). Thus, by varying the value of α we can formulate finite horizon problems as infinite horizon problems. To find a suitable value of α , we can simulate the system with different values and use the one that best describes the system behavior.

Finally, we compute the performance bounds for the simulated system based on Thm. 3.2. Given the initial locations of the relay nodes \mathbf{z}_0 , we define the set $\mathcal{S}_{\hat{\boldsymbol{\mu}}_{\text{iter}}}$ as the trajectory of the nodes under the policy $\hat{\boldsymbol{\mu}}_{\text{iter}}$ starting from \mathbf{z}_0

$$\mathcal{S}_{\hat{\boldsymbol{\mu}}_{\text{iter}}} = \{\mathbf{z} \in \mathbb{R}^{2(N+2)} : \exists k \geq 0 \text{ s.t. } \mathbf{z} = (I_{2(N+2)} - B\hat{L})^k \mathbf{z}_0\}.$$

where \hat{L} is defined in Eq. (4) for policy $\hat{\boldsymbol{\mu}}_{\text{iter}}$. The set $\mathcal{S}^{\hat{\boldsymbol{\mu}}}$ is the union of trajectories that result under the policy $\hat{\boldsymbol{\mu}}$ with each point in $\mathcal{S}_{\hat{\boldsymbol{\mu}}_{\text{iter}}}$ as an initial condition. Similarly, \mathcal{S}^* is the union of trajectories under the optimal policy starting from each point in $\mathcal{S}^{\hat{\boldsymbol{\mu}}}$. Let

$$\eta^{\hat{\boldsymbol{\mu}}_{\text{iter}}} = \|J^{\hat{\boldsymbol{\mu}}_{\text{iter}}} - J^*\|_{\mathcal{S}_{\hat{\boldsymbol{\mu}}_{\text{iter}}}}, \quad \eta_{\text{max}}^{\hat{\boldsymbol{\mu}}_{\text{iter}}} = \epsilon_{\text{iter}} + \frac{2\alpha\epsilon}{1-\alpha},$$

i.e., $\eta^{\hat{\boldsymbol{\mu}}_{\text{iter}}}$ is the maximum error between the optimal and the suboptimal policies computed based on system simulation, and $\eta_{\text{max}}^{\hat{\boldsymbol{\mu}}_{\text{iter}}}$ is the analytically computed upper bound. Then for the simulated system, the performance bounds and the actual performance results are as follows:

	$\alpha = 0.99$	$\alpha = 0.85$
ϵ	1.4841×10^5	1.2488×10^3
ϵ_{iter}	1.225×10^4	7.8×10^{-3}
$\eta_{\text{max}}^{\hat{\boldsymbol{\mu}}_{\text{iter}}}$	2.93×10^7	1.4153×10^4
$\eta^{\hat{\boldsymbol{\mu}}_{\text{iter}}}$	8.167×10^3	0.8566

As pointed out in Section III, the upper bounds are conservative but these bounds still help the designers to prepare for the worst case scenario.

VI. CONCLUSION

In this work, we presented a real-time framework for computing suboptimal control actions for distributed multiagent systems. In the proposed framework, the emphasis was on intelligently utilizing the prior information of the system available to formulate an infinite horizon discounted LQR problem, which provided an analytical expression for the optimal cost function. Using this function as a standard, we computed approximate cost function that was close to the optimal function, resulted in stable dynamics, and satisfied the communication network constraints, i.e., costs of non-neighboring nodes were decoupled. Finally we used efficient distributed optimization algorithms to solve the problem online. We also derived novel upper bounds for performance gap between the optimal and the suboptimal policies with restrictive communication among neighboring nodes.

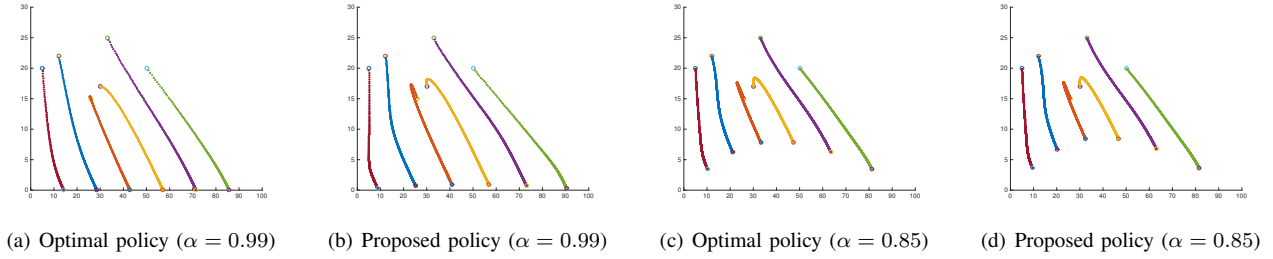


Fig. 4. Trajectories of the mobile relay nodes under the optimal policy and the proposed policy for $\alpha = 0.99$ and $\alpha = 0.85$.

APPENDIX

Proposition 1: If $Q \in \mathbb{R}^{2(N+2) \times 2(N+2)}$ is a symmetric M matrix, $B = \begin{bmatrix} I_{2N} \\ 0_{4 \times 2N} \end{bmatrix}$ and $R = \kappa_M I_{2N}$, then the positive definite solution of the Riccati equation

$$K = \alpha K - \alpha^2 K B (\alpha B^T K B + R)^{-1} B^T K + Q.$$

is also an M matrix. Moreover, if Q is a Laplacian matrix of a connected graph then all the entries of K are non-zero.

Proof: If $B = I_{2(N+2)}$ and $\alpha = 1$, then this result has been proved in Thm. 5.1 of [23]. We will extend this result for B and α as defined above. We start the proof by representing the matrices Q and K as block matrices following the convention introduced in (9).

$$Q = \begin{bmatrix} Q_f & Q_{fl} \\ Q_{fl}^T & Q_l \end{bmatrix} \text{ and } K = \begin{bmatrix} K_f & K_{fl} \\ K_{fl}^T & K_l \end{bmatrix},$$

Then,

$$\begin{aligned} Q &= (1 - \alpha)K + \alpha^2 K B (\alpha B^T K B + R)^{-1} B^T K. \\ B^T Q B &= (1 - \alpha)B^T K B + \alpha^2 B^T K B (I + \alpha B^T K B R^{-1})^{-1} \\ &\quad \dots R^{-1} B^T K B \\ R^{-1} Q_f &= (1 - \alpha)R^{-1} K_f + \alpha^2 R^{-1} K_f (I + \alpha K_f R^{-1})^{-1} \\ &\quad \dots R^{-1} K_f \end{aligned} \quad (11)$$

The last equality is based on matrix decomposition, $B^T K B = K_f$ and $B^T Q B = Q_f$ and multiplying both sides by R^{-1} . Because $R = \kappa_M I_{2N}$, it commutes with any square matrix. Using matrix inversion lemma

$$\begin{aligned} R^{-1} Q_f &= R^{-1} K_f - \alpha (I + \alpha K_f R^{-1})^{-1} R^{-1} K_f \\ (R^{-1} K_f) (R^{-1} Q_f) &= (R^{-1} K_f)^2 - \alpha R^{-1} K_f (I + \alpha K_f R^{-1})^{-1} \\ &\quad R^{-1} K_f \end{aligned}$$

From Eq. (11), $\alpha^2 R^{-1} K_f (I + \alpha K_f R^{-1})^{-1} R^{-1} K_f = R^{-1} Q_f - (1 - \alpha) R^{-1} K_f$. Using this equality in the above equation yields the following quadratic equation.

$$(R^{-1} K_f)^2 - R^{-1} K_f \left(R^{-1} Q_f - \frac{1 - \alpha}{\alpha} I \right) - \frac{1}{\alpha} R^{-1} Q_f = 0$$

After performing a series of simple algebraic manipulations

$$R^{-1} K_f = G + \sqrt{G + \frac{4}{\alpha} \sqrt{G} - \frac{2(1 - \sqrt{\alpha})}{\alpha} I_{2N}}, \quad (12)$$

where $G = R^{-1} Q_f + \frac{(1 - \sqrt{\alpha})^2}{\alpha}$. Since Q_f is a symmetric positive semidefinite M matrix, G is a positive definite M matrix and $G_s = \sqrt{G}$ is also a positive definite M matrix

[25], and all of its entries are non-zero [23]. Using Lemma 5.5 of [23], $G + \sqrt{G + \frac{4}{\alpha} \sqrt{G}}$ is an M matrix with negative off-diagonal entries. Now, $R^{-1} K$ is a positive definite matrix, so its principal submatrix $R^{-1} K_f$ is also positive definite. Thus, all the diagonal entries of $R^{-1} K_f$ are positive proving that K_f is a positive definite M matrix.

Next we will show that all the entries of K_{fl} are negative. Using the definition of B ,

$$\begin{aligned} Q_{fl} &= (1 - \alpha)K_{fl} + \alpha^2 K_f (\alpha K_f + R)^{-1} K_{fl} \\ K_{fl} &= [(1 - \alpha)I + \alpha^2 K_f (\alpha K_f + R)^{-1}]^{-1} Q_{fl} \\ &= (1 - \alpha) (sI - D)^{-1} Q_{fl} \end{aligned}$$

where $s = \left(1 + \frac{\alpha^2}{1 - \alpha}\right)$ and $D = \frac{\alpha^2}{1 - \alpha} (I + \alpha R^{-1} K_f)^{-1}$. Because $\alpha R^{-1} K_f$ is an M matrix, D is a positive matrix. Furthermore, $\rho((I + \alpha R^{-1} K_f)^{-1}) < 1$, which implies that $\rho(D) < s$ and $(1 - \alpha) (sI - D)^{-1}$ is a positive matrix. Since Q_{fl} has all the entries non-positive, K_{fl} has all the entries strictly negative unless an entire column of Q_{fl} is zero which is not possible for a connected graph.

Finally, to show that K_l ,

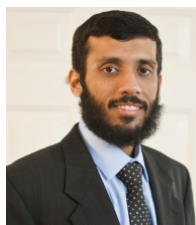
$$Q_l = (1 - \alpha)K_l + \alpha^2 K_{fl}^T (\alpha K_f + R)^{-1} K_{fl}$$

Since $(\alpha K_f + R)^{-1}$ is a positive matrix and all the entries of K_{fl} are negative, $\alpha^2 K_{fl}^T (\alpha K_f + R)^{-1} K_{fl}$ is a positive matrix. By definition Q_l has positive diagonal entries and non-positive off-diagonal entries. To satisfy the above equation, K_l must have positive diagonal entries and negative off-diagonal entries. This concludes the proof. ■

REFERENCES

- [1] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. II, 2nd Edition, Athena Scientific, Belmont, MA, 2007
- [2] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996.
- [3] J. R. Marden and J. S. Shamma, "Game Theory and Distributed Control," in *Handbook of Game Theory*, (H. P. Young and S. Zamir, Eds.), vol. 4, Elsevier Science, 2014.
- [4] R. Scattolini, "Architectures for distributed and hierarchical model predictive control-A review," in *J. Process Control*, no. 19, pp. 723–731, 2009.
- [5] L. Bakule, "Decentralized control: An overview," in *Annu. Rev. Control*, vol. 32, no. 1, pp. 87–98, 2008.
- [6] D.K. Goldenberg, J. Lin, and A.S. Morse, "Towards Mobility as a Network Control Primitive," in *Proc. ACM MobiHoc*, pp. 163–174, 2004.
- [7] F. El-Moukaddem, E. Tornig, G. Xing, and S. Kulkarni, "Mobile relay configuration in data-intensive wireless sensor networks," in *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 261–273, Feb. 2013.

- [8] Y. Yan and Y. Mostofi, "Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments," in *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1562-1572, Apr. 2013.
- [9] Y. Yan and Y. Mostofi, "To go or not to go: On energy-aware and communication-aware robotic operation," in *IEEE Transactions on Control of Network Systems*, vol. 1, iss. 3, pp. 218-231, July 2014.
- [10] H. Jaleel, Y. Wardi, and M. Egerstedt, "Minimizing mobility and communication energy in robotic networks: An optimal control approach," in *Proc. ACC*, Portland, OR, June 2014.
- [11] U. Ali, Y. Yan, Y. Mostofi, and Y. Wardi, "An optimal control approach for communication and motion co-optimization in realistic fading environments," in *Proc. ACC*, 2015, pp. 2930-2935.
- [12] G. Ferrari-Trecate, L. Galbusera, M. P. E. Marciandi, and R. Scattolini, "Model predictive control schemes for consensus in multiagent systems with single and double integrator dynamics," in *IEEE Trans. Autom. Control*, vol. 54, iss. 11, pp. 2560-2572, Nov. 2009.
- [13] C. Dixon and E. Frew, "Maintaining optimal communication chains in robotic sensor networks using mobility control," in *Mobile Networks and Applications*, vol. 14, iss. 3, pp. 281-291, 2009.
- [14] M. A. Müller, M. Reble, and F. Allgöwer, "Cooperative control of dynamically decoupled systems via distributed model predictive control," in *Int. J. of Robust & Nonlinear Control*, vol. 22, pp. 1376-1397, 2012.
- [15] H. Li, W. Yan, "Receding horizon control based consensus scheme in general linear multiagent systems," in *Automatica*, vol. 56, pp. 12-18, 2015.
- [16] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," in *Automatica*, vol. 42, no. 4, pp. 549-558, Apr. 2006.
- [17] P. Giselsson and A. Rantzer, "Distributed model predictive control with suboptimality and stability guarantees," in *Proc. 49th IEEE Conf. Dec. Control*, Atlanta, GA, USA, 2010.
- [18] R.L. Raffard, C.J. Tomlin, and S.P. Boyd, "Distributed optimization for cooperative agents: Application to formation flight," in *Proc. 43rd IEEE Conf. Decision Control*, 2004, pp. 2453-2459.
- [19] H. Terelius, U. Topcu, and R. M. Murray, "Decentralized multi-agent optimization via dual decomposition," in *Proc. 18th World Congr. of Int. Fed. Autom. Control (IFAC)*, Milano, Italy, Aug. 2011, pp. 7391-7397.
- [20] A. Nedic, A. Ozdaglar, and P. Parrilo, "Constrained consensus and optimization in multiagent networks," in *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922-938, 2009.
- [21] C. R. Johnson, "Inverse M-matrices," in *Linear Algebra Appl.* vol. 47, pp. 195-216, 1982.
- [22] Y. Mei, Y-H. Lu, Y. C. Hu, and C.S. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques," in *Proc. 12th IEEE ICAR*, pp. 492-497, 2005.
- [23] Y. Cao and W. Ren, "Optimal linear consensus algorithms: An LQR perspective," in *IEEE Trans. System, Man, Cybernetics*, vol. 40, no. 3, pp. 819-830, Mar. 2010.
- [24] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," in *Automatica*, vol. 42, no. 12, pp. 2105-2115, Dec. 2006.
- [25] G. Alefeld and N. Schneider, "On square roots of M-matrices," in *Linear Algebra Appl.*, vol. 42, pp. 119-132, Feb. 1982.
- [26] Y. Wang and S. Boyd, "Performance bounds and suboptimal policies for linear stochastic control via LMIs," 2009. [Online]. Available: www.stanford.edu/~boyd/papers/gen_ctrl_bnds.html
- [27] Y. Wang and S. Boyd, "Approximate dynamic programming via iterated Bellman inequalities," in *Int. J. Robust Nonlinear Control*, vol. 25, no. 10, pp. 1472-1496, Jul. 2015.
- [28] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method and multipliers," in *Foundations and Trends in Machine Learning*, vol. 3, pp. 1-122, 2011
- [29] A. Nedic and A. Ozdaglar, "Cooperative distributed multiagent optimization," in *Convex Optimization in Signal Processing and Communications*, Y. Eldar and D. Palomar, Eds. New York: Cambridge Univ. Press, 2009, pp. 340-386.
- [30] V. Gupta, B. Hassibi, and R. M. Murray, "A suboptimal algorithm to synthesize control laws for a network of dynamic agents," in *Int. J. Control*, vol. 78, no. 16, pp. 1302-1313, Nov. 2005.
- [31] M. R. Jovanovic and N. K. Dhingra, "Controller architectures: Tradeoffs between performance and structure," in *Eur. J. Control*, vol. 30, pp. 76-91, July 2016.
- [32] A. Mahajan, N. C. Martins, M. C. Rotkowitz, and S. Yuksel, "Information structures in optimal decentralized control," in *Proc. IEEE Conf. Decision and Control*, Maui, HI, USA, pp. 1291-1306, Dec. 2012.
- [33] N. Matni and V. Chandrasekaran, "Regularization for design," in *IEEE Trans. on Aut. Control*, vol. 61, no. 12, pp. 3991-4006, Dec. 2016.
- [34] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," in *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 2718-2723, May 2013.
- [35] C. Conte, T. Summers, M. Zeilinger, M. Morari, and C. Jones, "Computational aspects of distributed optimization in model predictive control," in *Proc. IEEE Conf. Decision and Control*, 2012.
- [36] B. Bamieh, F. Paganini, and M. Dahleh, "Distributed control of spatially invariant systems," in *IEEE Trans. Autom. Control*, vol. 47, pp. 1091-1118, July 2002.
- [37] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," in *IEEE Trans. Autom. Control*, vol. 51, no. 2, pp. 274-286, Feb. 2006.



Hassan Jaleel Hassan Jaleel is a Postdoctoral Research Fellow at Robotics, Intelligent Systems & Control (RISC) lab, KAUST, Saudi Arabia. He received his M.S. and PhD degrees in Electrical and Computer Engineering from the Georgia Institute of Technology in Atlanta, Georgia, in 2010 and 2013 respectively, and B.S. degree in Electrical Engineering from UET, Lahore, Pakistan, in 2007. His research interests lie in the areas of real-time distributed optimization, game-theoretic methods for multiagent systems, and robotics. In particular, he is

interested in designing distributed control strategies that can be implemented in real time on multiagent systems operating under uncertainties with limited onboard processing capabilities. He is also working on developing new solution concepts for stochastic learning dynamics in games, and a cooperative game theoretic framework for online distributed matching algorithms with complementarities. Jaleel was a Fulbright scholar from 2009 to 2013.



Jeff Shamma Jeff S. Shamma is a Professor and Chair of Electrical Engineering at the King Abdullah University of Science and Technology (KAUST) in Thuwal, Saudi Arabia, where he is also the director of the Robotics, Intelligent Systems & Control laboratory (RISC). He is the former Julian T. Hightower Chair in Systems & Control in the School of Electrical and Computer Engineering at Georgia Tech. He also has held faculty positions at the University of Minnesota, The University of Texas at Austin, and the University of California, Los Angeles. Shamma

received a Ph.D. in systems science and engineering from MIT in 1988. He is the recipient of an NSF Young Investigator Award, the American Automatic Control Council Donald P. Eckman Award, and the Mohammed Dahleh Award. Shamma is a Fellow of the IEEE and the International Federation of Automatic Control (IFAC) and a current distinguished lecturer of the IEEE Control Systems Society. He is the deputy editor-in-chief for the *IEEE Transactions on Control of Network Systems*. Shamma's research is in the areas of feedback control, systems theory, and robotics. His most recent research has been in decision and control for distributed multiagent systems and the related topics of game theory and network science, with applications to cyberphysical and societal network systems.