SIAM J. CONTROL OPTIM.
Vol. 51, No. 1, pp. 172–194

A FINITE-TIME DUAL METHOD FOR NEGOTIATION BETWEEN DYNAMICAL SYSTEMS[*]

DANIEL ZELAZO[†], MATHIAS BÜRGER[‡], AND FRANK ALLGÖWER[‡]

Abstract. This work presents a distributed algorithm for online negotiations of an optimal control policy between dynamical systems. We consider a network of self-interested agents that must agree upon a common state within a specified finite time. The proposed algorithm exploits the distributed structure of the corresponding dual problem and uses a "shrinking horizon" property to enforce the finite-time constraint. It is shown that this algorithm evolves like a time-varying and linear dynamical system, parameterized by a scalar variable analogous to the step-size rule in subgradient methods. The convergence and performance properties of the system are studied in the context of error systems between the algorithm trajectories and a sequence of centralized optimal control trajectories. This analysis provides a simple linear matrix inequality condition for choosing a proper step-size rule and also gives conditions for when no step-size rule can guarantee uniform convergence of the error systems. These conditions are shown to be functions of communication graph Laplacian eigenvalues and the state and control weights of each agent. We also provide a lower bound on the horizon time that guarantees that the terminal state generated by the algorithm is δ-close to an agreement state. The results are then demonstrated via a few numerical simulations.

Key words. distributed optimization, multiagent systems, finite-time consensus, subgradient methods

AMS subject classifications. 93C99, 90C20, 90C90, 90C35

DOI. 10.1137/100819175

1. Introduction.

Distributed algorithms for large-scale optimization problems are becoming increasingly important for a broad range of applications. These algorithms are motivated by scenarios where access to global information is either unavailable or unattainable due to the constraints of the system. These include limited computational resources, communication bandwidth, and power restrictions. While the study of distributed algorithms is not new, it has gained recent attention in the systems and controls community for its relevance to multiagent systems.

In multiagent systems, it is often the goal of a team of agents to achieve through cooperation and coordination some global objective. Due to the same constraints listed earlier, this team objective must be reached using distributed protocols for the control and decision making of each agent. One of the most well-studied problems related to this is known as the consensus, or agreement protocol [9, 11, 14]. In agreement problems, each agent must agree upon a common value of interest (e.g., the heading or velocity of a team of autonomous vehicles). Within the controls community, a primary focus is on the application of these distributed protocols to physical systems.

The elegance of the agreement protocol lies in its simplicity. It comes as no surprise that this algorithm also applies to other classes of problems beyond the control of physical systems. In fact, the origins of the agreement protocol can be traced to distributed computation and optimization problems [2, 17]. More recently, sub-

[*] Received by the editors December 22, 2010; accepted for publication (in revised form) November 16, 2012; published electronically January 8, 2013.
http://www.siam.org/journals/sicon/51-1/81917.html

[†] Faculty of Aerospace Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel (dzelazo@technion.ac.il).

[‡] Institute for Systems Theory and Automatic Control, University of Stuttgart, 70550 Stuttgart, Germany (buerger@ist.uni-stuttgart.de, allgower@ist.uni-stuttgart.de).

gradient algorithms have become a focal point for research in distributed optimization [10, 6, 20]. Although the consensus problem and distributed optimization problems are strongly related, a major difference is that in the latter the agents are not physical entities but processing nodes.

There has been some recent work lying at the intersection of these two fields focusing simultaneously on the control of physical systems and distributed solutions to global optimization problems. Such a scenario has been considered in [7], where each agent negotiates a consensus value based on some cost function using a distributed optimization algorithm before controlling the physical system to that value. Dual decomposition is used in [13] for an optimal distributed controller design. In [4], a dual decomposition method is used to dynamically determine an optimal velocity for a team of self-interested agents.

This paper studies a distributed optimization problem which is coupled to a physical control system. A team of self-interested agents is considered that should achieve consensus at a specified time. In particular, we consider an ensemble of single integrator agents each equipped with a quadratic cost function penalizing its distance to a desired state, referred to as its "preference," and its control energy. The agents are only coupled by the consensus constraint at the end of a finite-time horizon. While this problem can be formulated as a centralized optimal control problem (OCP), we study a distributed solution that negotiates the optimal control policy in real time based on a dual decomposition subgradient algorithm. Under the premise that communication and computation are not instantaneous, we assume that between communication rounds, the agents are already moving in the direction they consider to be optimal at that time instance. The dynamic element of this problem effectively changes the parameters of the optimization problem, and we consider how this change deviates from the solution of the centralized static case. The distributed dual subgradient algorithm we develop relies on the notion of a *shrinking horizon* to account for the dynamic changes in the system as time progresses. We term this the *shrinking horizon preference agreement* (SHPA) algorithm.

The description and analysis of the SHPA algorithm constitute the main contributions of this paper. In particular, we first show that the SHPA algorithm can be described as a linear time-varying (LTV) dynamical system containing the physical state of each agent and an augmented state associated with each edge in the communication graph. The augmented state can be interpreted as the Lagrange multiplier associated with the terminal coupling constraint. The performance of the SHPA algorithm, therefore, depends strongly on the trajectory of this augmented state. In this direction, we also define an LTV error system between the augmented state and the optimal multiplier value corresponding to the centralized OCP. The convergence properties and performance of the SHPA and error dynamics LTV description depend on the choice of a parameter analogous to the step-size rule for subgradient algorithms. A surprising result highlights that under certain communication topologies and state and control weights for each agent, no step-size rule exists that can guarantee uniform convergence of the LTV systems. For the general SHPA setup, we present a feasibility linear matrix inequality (LMI) to determine the existence of a step-size rule. In the special case where every agent has an identical cost function, we derive explicit lower and upper bounds for the step-size that depend on the cost function weights and the ratio of the largest and smallest nontrivial graph Laplacian eigenvalues. We also provide a lower bound on the horizon time that guarantees that the terminal state generated by the SHPA algorithm is δ-close to an agreement state.

The organization of the paper is as follows. In the next subsection we introduce our notation. The general problem setup, including the formulation of the centralized OCP, is given in section 2, which also reviews the static distributed dual subgradient algorithm. In section 3 the SHPA algorithm is presented. A convergence and performance analysis of the algorithm is given in section 4. Finally, in section 5 a simulation example is given, and section 6 provides some concluding remarks.

Notation. The notation we employ is standard. The set of real numbers is denoted $\mathbb{R}$, and $\mathbb{R}_{>}$ ($\mathbb{R}_{\geq}$) is the set of positive (nonnegative) numbers. For a vector $x \in \mathbb{R}^m$, we denote its transpose by x' and its ith component by $x(i)$; the ijth element of the matrix A is given as $[A]_{ij}$. The all-ones vector of length m is denoted $\mathbb{1}_m$ and I_m is the $m \times m$ identity matrix. The inner product of two vectors is denoted $\langle x, y\rangle = x'y$; the Euclidean norm of a vector x is denoted $\|x\| = \langle x, x\rangle^{1/2}$. Similarly, the spectral norm of a matrix A is denoted $\|A\|$. The ith canonical basis vector for the Euclidean vector space $\mathbb{R}^m$ is denoted $e_{i,m}$; that is, $e_{i,m} \in \mathbb{R}^m$ and $[e_{i,m}]_j = 1$ if $j = i$ and 0 otherwise.

The communication structure between agents is captured by a graph $\mathcal{G}$ with node set $\mathcal{V} = \{v_1, \dots, v_n\}$ and edge set $\mathcal{E}$. A *spanning tree* is a connected graph with $|\mathcal{V}|-1$ edges and does not contain cycles. The *incidence matrix* of the graph $\mathcal{G}$, $E \in \mathbb{R}^{n\times|\mathcal{E}|}$, is a $\{0, \pm 1\}$-matrix with rows and columns indexed by the vertices and edges of $\mathcal{G}$ such that $[E]_{ik}$ has the value $+1$ if node i is the initial node of edge k, -1 if it is the terminal node, and 0 otherwise [5]. In this work we consider only fixed topologies, and for notational convenience we write only E for the incidence matrix.

2. The preference-based agreement problem.

We study the problem of a group of self-interested dynamical agents that must agree upon a common state at the end of a given time horizon. The agents are modeled as a group of n single integrator systems,

$$x_i(t+1) = x_i(t) + u_i(t),\ x_i(0) = x_{i0}, \tag{2.1}$$

with $i = 1, \dots, n$ and $x_i(t) \in \mathbb{R}$. The state and control vector for all n agents are denoted as $x(t) = [x_1(t), \dots, x_n(t)]'$ and $u(t) = [u_1(t), \dots, u_n(t)]'$. Agents can communicate with each other according to a fixed communication graph $\mathcal{G}$, assumed to be a spanning tree. Furthermore, we only consider *synchronous communication*, where all agents communicate at the same time instant.

The self-interest of each agent is modeled as a quadratic objective, attaining its minimum at a specific individual preference value ξ_i. Each agent aims to minimize the objective

$$J_i(t_0, T, x_i, u_i) = \frac{1}{2}\left(\sum_{t=t_0}^{T-1} q_i(x_i(t+1) - \xi_i)^2 + r_i u_i(t)^2\right), \tag{2.2}$$

where $q_i, r_i \in \mathbb{R}_{>}$ are the state and control weights. The individual agents are coupled by a requirement to achieve agreement at the end of the time horizon T; that is, there is a terminal time constraint,

$$x_1(T) = x_2(T) = \cdots = x_n(T) \Leftrightarrow E'x(T) = 0. \tag{2.3}$$

From a centralized perspective, the preference-based agreement problem can be stated as the OCP with terminal constraint

$$\begin{aligned} OCP(t_0,T,x_0):\ & \min_{x,u} \sum_{i=1}^{n} J_i(t_0,T,x_i,u_i) \\ \text{subject to (s.t.) } & x(t+1)=x(t)+u(t),\ x(t_0)=x_0, \\ & E'x(T)=0. \end{aligned} \tag{2.4}$$

We collect the entire state and control trajectories of each agent into the row vectors $\mathbf{x}_i = \left[\begin{array}{ccc} x_i(t_0+1) & \cdots & x_i(T)\end{array}\right]$ and $\mathbf{u}_i = \left[\begin{array}{ccc} u_i(t_0) & \cdots & u_i(T-1)\end{array}\right]$. As we are considering a team of n agents, we introduce further notation to streamline the presentation. The boldface vectors $\mathbf{x} = \left[\begin{array}{ccc} (\mathbf{x}_1)' & \cdots & (\mathbf{x}_n)'\end{array}\right]' \in \mathbb{R}^{n\times(T-t_0)}$ and $\mathbf{u} = \left[\begin{array}{ccc} (\mathbf{u}_1)' & \cdots & (\mathbf{u}_n)'\end{array}\right]' \in \mathbb{R}^{n\times(T-t_0)}$ denote the complete trajectories for the state and control of the entire ensemble of agents, and $(\overline{\mathbf{x}},\overline{\mathbf{u}})$ denotes the *optimal* trajectory generated by the solution of $OCP(t_0,T,x_0)$. At times, we will be interested in the state or control trajectory value for all agents at a particular time τ; we will denote this by $\mathbf{x}(\tau)\in\mathbb{R}^{n\times 1}$ and $\mathbf{u}(\tau)\in\mathbb{R}^{n\times 1}$.

Note that problem $OCP(t_0,T,x_0)$ can be reformulated as a static quadratic program. Using the new notation, the objective for each agent can be stated as $J_i(t_0,T,\mathbf{x}_i,\,\mathbf{u}_i) = \frac{1}{2}(q_i\|\mathbf{x}_i - \mathbb{1}'_{T-t_0}\xi_i\|^2 + r_i\|\mathbf{u}_i\|^2)$ and the dynamic constraint as the linear equation

$$\mathbf{x}_i = \mathbb{1}'_{T-t_0}x_{i0} + \mathbf{u}_i B'_{T-t_0}. \tag{2.5}$$

Here, $B_{T-t_0}\in\mathbb{R}^{T-t_0\times T-t_0}$ is defined such that $[B_{T-t_0}]_{kl}=1$ for $k\geq l$ and zero otherwise.

Throughout this paper, we will not only rely on the primal problem formulation (2.4), but we will often consider the dual problem. The dual problem is obtained by relaxing the coupling constraint with a multiplier μ into the objective to obtain the Lagrangian,

$$\mathcal{L}(\mathbf{x},\mathbf{u},\mu) = \sum_{i=1}^{n} J_i(t_0,T,\mathbf{x}_i,\,\mathbf{u}_i) + \mu' E'\mathbf{x}(T). \tag{2.6}$$

The dual function is obtained by minimizing (2.6) subject to the dynamic constraint (2.5), $g(\mu)=\min_{\mathbf{x},\mathbf{u}}\mathcal{L}(\mathbf{x},\mathbf{u},\mu)$. The *dual problem* can thus be stated as

$$\max_{\mu\in\mathbb{R}^{|\mathcal{E}|}} g(\mu).$$

We denote the optimal solution of the primal and dual problems as $(\overline{\mathbf{x}},\overline{\mathbf{u}},\overline{\mu})$. As $OCP(t_0,T,x_0)$ is a strictly convex problem (a quadratic program with linear constraints), we have strong duality which implies that $g(\overline{\mu}) = J(t_0,T,\overline{\mathbf{x}},\overline{\mathbf{u}})$ [15].

2.1. Shrinking horizon OCP. As a precursor to the main results of this work, we briefly discuss the solution of the OCP as the problem parameters vary. In particular, we will be interested in how the OCP evolves as the time horizon shrinks and the initial conditions change. For a given initial time t, we denote the time horizon as $\tilde{T}=T-t$. We denote the optimal solution of the primal and dual problems associated with $OCP(t,T,x(t))$ by the triple

$$\left(\overline{\mathbf{x}}^{(t,x(t))},\overline{\mathbf{u}}^{(t,x(t))},\overline{\mu}^{(t,x(t))}\right)\in\mathbb{R}^{n\times\tilde{T}}\times\mathbb{R}^{n\times\tilde{T}}\times\mathbb{R}^{n-1}. \tag{2.7}$$

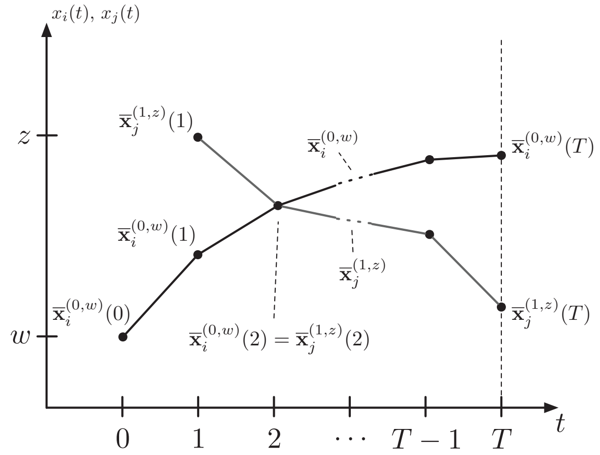

$x_i(t)$, $x_j(t)$
z
w
$\overline{\mathbf{x}}_j^{(1,z)}(1)$
$\overline{\mathbf{x}}_i^{(0,w)}$
$\overline{\mathbf{x}}_i^{(0,w)}(T)$
$\overline{\mathbf{x}}_i^{(0,w)}(1)$
$\overline{\mathbf{x}}_j^{(1,z)}$
$\overline{\mathbf{x}}_i^{(0,w)}(0)$
$\overline{\mathbf{x}}_j^{(1,z)}(T)$
$\overline{\mathbf{x}}_i^{(0,w)}(2) = \overline{\mathbf{x}}_j^{(1,z)}(2)$
0
1
2
$\cdots$
$T-1$
T
t

FIG. 2.1. *Illustration of notation for optimal trajectories generated by the OCP.*

When the initial condition used in $OCP(t, T, x(t))$ is unambiguously understood, we use the shorthand notation $\overline{\mathbf{x}}^t$, $\overline{\mathbf{u}}^t$ and $\overline{\mu}^t$ for the state and control trajectories and dual multipliers of all agents. Figure 2.1 gives an interpretation of our adopted notation.

Using our notation, we formally state the celebrated principle of optimality for dynamic programming in the context of the OCP. This corresponds to solving a sequence of OCPs where the initial condition of the subsequent problem is chosen to be along the optimal trajectory of the system.

LEMMA 2.1 (principle of optimality). *The optimal trajectories generated by* $OCP(t, T, z)$ *and* $OCP(t+1, T, w)$ *with* $w = z + \overline{\mathbf{u}}^{(t,z)}(t)$ *satisfy*

$$\left(\overline{\mathbf{x}}^{(t,z)}(\tau),\, \overline{\mathbf{u}}^{(t,z)}(\tau),\, \overline{\mu}^{(t,z)}\right) = \left(\overline{\mathbf{x}}^{(t+1,w)}(\tau),\, \overline{\mathbf{u}}^{(t+1,w)}(\tau),\, \overline{\mu}^{(t+1,w)}\right),\ \tau = t+1, \dots, T.$$

Proof. Concerning the primal solution, the initial condition for $OCP(t+1, T, w)$ corresponds to the point $\overline{\mathbf{x}}^{(t,z)}(t+1)$. The remaining statement is a direct application of the principle of optimality for dynamic programming [1], and its uniqueness is due to the strict convexity of the problem statement. The statement concerning the dual solution is a direct consequence of the first statement; in particular, we have $\overline{\mathbf{x}}^{(t,z)}(T) = \overline{\mathbf{x}}^{(t+1,w)}(T)$. A necessary condition for optimality is $\frac{\partial \mathcal{L}}{\partial \overline{\mathbf{x}}(T)} = Q(\overline{\mathbf{x}}^{(\cdot,\cdot)}(T) - \xi) + E\overline{\mu}^{(\cdot,\cdot)} = 0$, where $Q = \mathrm{diag}\{q_1, \dots, q_n\}$. The corresponding system of equations admits a unique solution since the communication graph $\mathcal{G}$ is restricted to a spanning tree. ▯

2.2. Distributed dual subgradient solution method. The original problem we aim to solve is $OCP(t_0, T, x_0)$. Note that the problem $OCP(t_0, T, x_0)$ is completely separable in its objective and is coupled only through the terminal time constraint (2.3). A standard procedure for solving $OCP(t_0, T, x_0)$ in a distributed fashion is thus through a *dual decomposition subgradient* algorithm [15] that we review here. The algorithm solves the dual problem, based on the Lagrangian,

$$\mathcal{L}(\mathbf{x}^{t_0}, \mathbf{u}^{t_0}, \mu) = \sum_{i=1}^{n} J_i(t_0, T, \mathbf{x}_i^{t_0},\, \mathbf{u}_i^{t_0}) + \mu' E' \mathbf{x}^{t_0}(T).$$

Since μ is associated with each edge in $\mathcal{G}$, we can consider instead the variable associated with each agent by defining

$$\gamma = E\mu. \tag{2.8}$$

The Lagrangian can be written as a function of γ as

$$\mathcal{L}(\mathbf{x}^{t_0}, \mathbf{u}^{t_0}, \gamma) = \sum_{i=1}^{n} J_i(t_0, T, \mathbf{x}_i^{t_0}, \ \mathbf{u}_i^{t_0}) + \gamma_i \mathbf{x}_i^{t_0}(T). \tag{2.9}$$

As we have already mentioned, the dual function is obtained by minimizing (2.9) subject to the dynamic constraint (2.5), $g(\gamma) = \min_{\mathbf{x}^{t_0}, \mathbf{u}^{t_0}} \mathcal{L}(\mathbf{x}^{t_0}, \mathbf{u}^{t_0}, \gamma)$.

The important feature of this formulation is that the dual function $q(\gamma)$ is completely separable across each agent. This then motivates the dual subgradient algorithm, which can be stated as follows. At each iteration step k of the algorithm, the dual function is computed for a fixed value of $\hat{\gamma}^{[k]}$. (For $k = 0$ the multiplier is initialized to some arbitrary value.) That is, each agent solves the following quadratic program, $QP_i(k)$:

$$(\hat{\mathbf{x}}_i^{[k+1]}, \hat{\mathbf{u}}_i^{[k+1]}) = \arg \min_{\hat{\mathbf{x}}_i^{[k]}, \hat{\mathbf{u}}_i^{[k]}} J_i(t_0, T, \hat{\mathbf{x}}_i^{[k]}, \hat{\mathbf{u}}_i^{[k]}) + \hat{\gamma}_i^{[k]} \hat{\mathbf{x}}_i^{[k]}(T) \tag{2.10}$$

$$\text{s.t. } \hat{\mathbf{x}}_i^{[k]} = \mathbb{1}'_{\tilde{T}} x_{i0} + \hat{\mathbf{u}}_i^{[k]} B'_{\tilde{T}}. \tag{2.11}$$

Here we have temporarily abused our notation to facilitate this discussion. The superscript, as in $\gamma^{[k]}$, denotes the iteration count for the subgradient algorithm, and the notation $(\hat{\mathbf{x}}_i^{[k]}, \hat{\mathbf{u}}_i^{[k]})$ denotes the optimization variables for $QP_i(k)$. While ensuring that the initial values of the dual variables satisfy $\gamma^{[0]} = E\mu^{[0]}$, the next step is then to update the multiplier using the subgradient as

$$\hat{\gamma}^{[k+1]} = \hat{\gamma}^{[k]} + \alpha^{[k]} E E' \hat{\mathbf{x}}^{[k+1]}(T). \tag{2.12}$$

The subgradient for the edge multiplier μ is precisely $E'\hat{\mathbf{x}}^{[k]}(T)$, and using (2.8) leads to (2.12). The matrix EE' is the *graph Laplacian* of $\mathcal{G}$ [5]. The choice of the step-size $\alpha^{[k]}$ is critical for the convergence properties of this algorithm. While there are many step-size rules that can guarantee convergence of this algorithm, tuning the step-size to achieve desirable convergence rates can be nontrivial. With a suitable choice for the step-size, the subgradient algorithm will converge to the optimal solution of $OCP(t_0, T, x_0)$,

$$\lim_{k \to \infty} (\hat{\mathbf{x}}^{[k]}, \hat{\mathbf{u}}^{[k]}, \hat{\gamma}^{[k]}) = (\overline{\mathbf{x}}^{(t_0, x_0)}, \overline{\mathbf{u}}^{(t_0, x_0)}, E\overline{\mu}^{(t_0, x_0)}).$$

For a more detailed discussion of appropriate step-size rules and subgradient methods the reader is referred to [15].

The appeal of this method is that the update rule (2.12) is inherently distributed. That is, each agent can compute the value $\gamma_i^{[k+1]}$ to use in the next iteration step solely through communication with its neighbors, as defined by the communication graph $\mathcal{G}$. In particular, agent i must only send the value $\hat{\mathbf{x}}_i^{[k]}(T)$ to all neighboring agents.

While the subgradient algorithm is attractive due to its distributed and relatively simple architecture, we note that this algorithm must be performed *before* each agent can begin moving along its optimal trajectory. Indeed, for good convergence of the algorithm, it may be required to run for a time significantly longer than the desired horizon time T. This then motivates the question of whether it is possible to derive an algorithm that can be implemented *online*. That is, we would like to develop an algorithm where each iteration step corresponds to the actual physical time, while additionally propagating the agents along a calculated trajectory. Such an algorithm should also negotiate the final consensus value in real time and, if possible, satisfy the terminal time constraint at the real time T while simultaneously minimizing the local performance index for each agent.

3. Shrinking horizon preference agreement algorithm. The need for a real-time distributed algorithm for solving problem $OCP(t_0, T, x_0)$ is based on the assumption that the time required to compute a sufficiently good solution using an offline algorithm corresponds to a period where agents must remain idle. If we consider the horizon time T as an absolute deadline, then an optimal strategy would require each agent to move toward their preference state in order to minimize their individual objectives before maneuvering to the consensus state.[1]

In this direction, we propose a real-time preference-based agreement algorithm inspired by the subgradient algorithm. The general strategy of this algorithm is to physically propagate the states forward at each iteration. The corresponding subproblem to be solved at the next time step is the quadratic program $QP_i(t)$, described in (2.10), but with the horizon window reduced; instead of minimizing from $t = 0$ to the horizon T, we minimize from $t = 1$. It can be considered as a *shrinking-horizon subgradient algorithm*. Here we recall that the state signal $x_i(t)$ corresponds to the true physical state of agent i at time t, and the vectors $\hat{\mathbf{x}}_i^t$ and $\hat{\mathbf{u}}_i^t$ correspond to the optimization variables associated with problem $QP_i(t)$. Note also that as time progresses, the window is shrinking, and $\hat{\mathbf{x}}_i^t, \hat{\mathbf{u}}_i^t \in \mathbb{R}^{\tilde{T}}$, $\tilde{T} = T - t$. See Algorithm 1 for a description.

Algorithm 1. SHPA algorithm.

```
Data: Initial conditions $x_i(0) = x_{i0}$ and $\mu(0) = \mu_0$; $t = 0$.
begin
    for $t := 0$ to $T$-$1$ do
        $\gamma^t = E\mu(t)$, $\tilde{T} = T - t$
        Each agent solves the subproblem $QP_i(t)$:

        (3.1)  $\min_{\hat{\mathbf{x}}_i(t), \hat{\mathbf{u}}_i(t)} J_i(t, T, \hat{\mathbf{x}}_i^t, \hat{\mathbf{u}}_i^t) + \gamma_i^t \hat{\mathbf{x}}_i^t(T)$  s.t.  $\hat{\mathbf{x}}_i^t = \mathbb{1}_{\tilde{T}} x_i(t) + B_{\tilde{T}} \hat{\mathbf{u}}_i^t$.

        The physical state and multipliers are propagated forward using the
        solution of $QP_i(t)$:

        (3.2)  $x_i(t+1) = x_i(t) + \hat{\mathbf{u}}_i^t(t), \; i = 1, \dots, n,$
        (3.3)  $\mu(t+1) = \mu(t) + \alpha(t) E' \hat{\mathbf{x}}^t(T),$

        where $\alpha(t)$ satisfies some step-size rule.
```

[1] This reasoning assumes that T is sufficiently large. For a shorter horizon each agent might not have enough time to reach its preference.

At the discrete time instant $t < T$, each agent i solves an OCP with the finite horizon $\tilde{T} = T - t$, using the given $\mu(t)$ for the estimated terminal constraint multiplier value. The optimal solution of $QP_i(t)$ is then used to propagate the actual *physical system state*, $x_i(t)$, forward. The new state is then used as the initial condition for the subsequent iteration. The key point here is that at each step of the algorithm, the agents are physically moving along the optimal trajectory calculated for a given multiplier value.

The relation of the SHPA algorithm to the dual subgradient methods should be clear from the update equation of the multiplier $\mu(t)$. The primary difference, as already mentioned, is that at each time step the physical state of the system is changing, and the corresponding subproblem $QP_i(t)$ is also modified. In this way, the SHPA algorithm can be interpreted as a *dynamic* negotiation protocol to determine the consensus value. The multipliers $\gamma_i(t)$ can then be considered as a kind of estimate by each agent of the preferences of neighboring agents.

A main result of this work is that the trajectories produced by Algorithm 1 are equivalent to the trajectories of an LTV dynamical system. The following theorem summarizes this result.

THEOREM 3.1. *Algorithm* 1 *is equivalent to the linear dynamical system*

$$\begin{bmatrix} x(t+1) \\ \mu(t+1) \end{bmatrix} = \begin{bmatrix} I - P(\tilde{T}) & -R^{-1}K(\tilde{T})E \\ \alpha(t)E'K(\tilde{T}) & I - \alpha(t)E'Q^{-1}P(\tilde{T})E \end{bmatrix} \begin{bmatrix} x(t) \\ \mu(t) \end{bmatrix} + \begin{bmatrix} P(\tilde{T}) \\ E'(I - \alpha(t)K(\tilde{T})) \end{bmatrix} \xi \tag{3.4}$$

with $\tilde{T} = T - t$, $Q = \operatorname{diag}\{q_1, \ldots, q_n\}$, $R = \operatorname{diag}\{r_1, \ldots, r_n\}$ *the weights of the optimization problem, and* $P(\tilde{T}) = \operatorname{diag}\{P_1(\tilde{T}), \ldots, P_n(\tilde{T})\}$, $K(\tilde{T}) = \operatorname{diag}\{K_1(\tilde{T}), \ldots, K_n(\tilde{T})\}$ *with each component satisfying the recursion*

$$P_i(\tilde{T}+1) = \frac{1 + \frac{r_i}{q_i}P_i(\tilde{T})}{1 + \frac{r_i}{q_i} + \frac{r_i}{q_i}P_i(\tilde{T})}, \qquad P_i(1) = \frac{q_i}{r_i + q_i}, \tag{3.5}$$

$$K_i(\tilde{T}+1) = \frac{r_i}{q_i}\frac{K_i(\tilde{T})}{1 + \frac{r_i}{q_i} + \frac{r_i}{q_i}P_i(\tilde{T})}, \quad K_i(1) = \frac{r_i}{r_i + q_i}. \tag{3.6}$$

The proof of the theorem is fairly lengthy and is therefore presented in Appendix A. The main effort for the proof is the derivation of the recursion expressions for the constants $P_i(\tilde{T})$ and $K_i(\tilde{T})$. We also make use of the analytic solutions for $\hat{\mathbf{u}}_i^t(t)$ and $\hat{\mathbf{x}}^t(T)$, which can be derived directly from the quadratic program subproblem in the SHPA algorithm; we present the expresions here for completeness:

$$\hat{\mathbf{u}}^t(t) = -P(\tilde{T})(x(t) - \xi) - R^{-1}K(\tilde{T})E\mu(t), \tag{3.7}$$

$$\hat{\mathbf{x}}^t(T) = K(\tilde{T})(x(t) - \xi) + \xi - Q^{-1}P(\tilde{T})E\mu(t). \tag{3.8}$$

System (3.4) has strong similarities to the classical finite horizon linear quadratic regulator (LQR) control problem. In fact, the time-varying constants $P_i(\tilde{T})$ are precisely the time-varying finite horizon LQR gains.

COROLLARY 3.2. *The gains* $P_i(\tilde{T})$ *are the time-varying finite horizon LQR controller gains for agent* i *with state and control weights* q_i *and* r_i, *respectively.*

The details of the proof are omitted for brevity, noting the equivalence can be obtained directly from the LQR Riccati and controller recursions [8].

Although the similarity of the $x(t)$ dynamics to a finite horizon LQR problem is not too surprising, it is remarkable that the same controller gains reappear in the update rule for the multipliers $\mu(t)$. Note that these gains can be computed *offline* and independently of the algorithm, as they require no knowledge of the global objective or communication topology. Furthermore, the LQR gains associated with each agent are in fact unrelated to the general statement of our control problem; there is no information contained in those gains about the team objective.

The structure of (3.4) also shows that each agent is driven by its deviation from its preference state (the term $P(\tilde{T})(x(t)-\xi)$) and a terminal state correction term via the multiplier $\gamma(t)$ (the term $R^{-1}K(\tilde{T})E\mu(t)$). We also note that the update equation for the multiplier values resembles that of the standard consensus protocol. The $\mu(t)$ dynamics contain the term $E'Q^{-1}P(\tilde{T})E$, which is a weighted edge Laplacian of the graph $\mathcal{G}$. Dynamics induced by the edge Laplacian were shown to be equivalent to the standard consensus problem in [19].

The LTV representation (3.4) of Algorithm 1 is compelling for a few reasons. With this representation we are able to directly analyze the stability and convergence properties of the algorithm using tools from linear systems theory. In particular, we note that the only free parameter in (3.4) is the step-size $\alpha(t)$.[2] The proper choice for $\alpha(t)$ can then be cast as a stabilization and performance problem for the system in (3.4).

Having established that the shrinking horizon algorithm evolves like a linear time-varying system with a consensus-like structure, it remains to analyze the convergence properties of the algorithm, which we show in what follows.

4. Performance and convergence analysis.

Recall that the coupling constraint of the OCP requires all agents to agree upon a common state at the end of the horizon T, as stated in (2.3). A reasonable measure for the performance of this algorithm, therefore, is how far the agents are from consensus at the horizon time T. This is captured by the norm of the consensus state,

$$\|E(\mathcal{G})'x(T)\|. \tag{4.1}$$

Note that for the centralized solution (e.g., problem $OCP(t,T,x)$) this quantity is precisely zero. In this direction, we observe that at each time t and state $x(t)$, there is a corresponding optimal multiplier value, $\overline{\mu}^t$, associated with the centralized OCP, $OCP(t,T,x(t))$, that will lead to perfect consensus. This value will, in general, be different from $\mu(t)$, generated by (3.4). Therefore, the error between $\overline{\mu}^t$ and $\mu(t)$ is an indicator of the performance of the SHPA algorithm.

We can make this statement more explicit by first deriving an explicit expression for the multiplier associated with $OCP(t,T,x(t))$. We use the fact that the optimal control input and the optimal terminal state can be computed from the expressions (3.7) and (3.8) by replacing the multiplier $\mu(t)$ with the optimal multiplier $\overline{\mu}^t$, i.e.,

$$(\overline{\mathbf{u}}^t(t), \overline{\mathbf{x}}^t(T)) = (\hat{\mathbf{u}}^t(t), \hat{\mathbf{x}}^t(T))|_{\mu(t)=\overline{\mu}^t}. \tag{4.2}$$

COROLLARY 4.1. *The optimal multiplier values $\overline{\mu}^{\,t}$ corresponding to the problem $OCP(t,T,x(t))$ is given as*

$$\overline{\mu}^{\,t} = \left(E'Q^{-1}P(\tilde{T})E\right)^{-1} E' \left[K(\tilde{T})(x(t)-\xi)+\xi\right]. \tag{4.3}$$

Proof. The primal feasibility of the problem $OCP(t,T,x(t))$ guarantees the satisfaction of the terminal time constraint, $E'\overline{\mathbf{x}}^{(t,x(t))}(T)=0$. The terminal state value as a function of the optimal multiplier $\overline{\mu}^{\,t}$ can be computed directly from (3.8) and (4.2). The invertability of the matrix $E'Q^{-1}P(\tilde{T})E$ is given since $Q^{-1}P(\tilde{T})$ is a positive

[2] Although specified as a time-varying function, we note that a constant step-size rule will also suffice.

diagonal matrix. Thus $E'Q^{-1}P(\tilde{T})E$ is a weighted edge Laplacian of a tree, having only non-zero eigenvalues [19]. □

The key feature, which we restate here, is that at each time horizon and state pair, there exists a *unique* optimal multiplier $\overline{\mu}^{(t,x(t))}$ leading to perfect consensus. sequence $\overline{\mu}^{t}$ perfectly, then the system should reach the consensus state exactly, and $E(\mathcal{G})'x(T) = 0$. If the multipliers generated by (3.4) are able to track the multipliers $\overline{\mu}^{t}$ associated with a sequence of OCPs varying along the state trajectories of (3.4), then the terminal constraint can be met exactly. We are now prepared to consider the *multiplier estimation error* between the system (3.4) and the optimal multiplier values $\overline{\mu}^{t}$,

$$\epsilon(\tilde{T}, x(t)) = \mu(t) - \overline{\mu}^{(t,x(t))}, \tag{4.4}$$

for performance analysis. For notational convenience we will sometimes write $\epsilon(t)$ in place of $\epsilon(\tilde{T}, x(t))$.

This highlights a significant difference between an implicit objective of the shrinking horizon agreement algorithm and the static dual decomposition subgradient algorithm used to solve $OCP(t_0, T, x_0)$. In particular, for the dual decomposition subgradient algorithm, it is desired that the multiplier at each iteration k, $\mu^{[k]}$ converge to the multiplier for the centralized problem, $\overline{\mu}^{0} := \overline{\mu}^{(t_0,x_0)}$ ($\lim_{k\to\infty} \|\mu^{[k]} - \overline{\mu}^{0}\| \to 0$). In contrast, with the shrinking horizon agreement algorithm, we want the multiplier estimate to satisfy

$$\lim_{t\to T} \|\mu(t) - \overline{\mu}^{(t,x(t))}\| \to 0.$$

We can now analyze how the error $\epsilon(t)$ evolves along the trajectories of the system (3.2), (3.3). A main result of this work, therefore, is the observation that the error evolves as a time-varying linear system with the step-size $\alpha(t)$ as a parameter. We summarize the result in the following theorem.

THEOREM 4.2. *The error $\epsilon(t) = \mu(t) - \overline{\mu}^{(t,x(t))}$ evolves according to the time-varying linear dynamics*

$$\epsilon(t+1) = \left((E'Q^{-1}P(\tilde{T}-1)E)^{-1} - \alpha(t)I\right) E'Q^{-1}P(\tilde{T})E\epsilon(t). \tag{4.5}$$

The proof is provided in Appendix B.

The LTV presentation of the error dynamics highlights the important role of the step-size $\alpha(t)$. Indeed, a poorly chosen step-size can in fact destabilize the system. This mirrors the same difficulties encountered with choosing step-sizes for subgradient methods. However, considering the real-time version of the dual subgradient algorithm gives rise to a novel phenomenon, which has no correspondence in the static case. In this direction, we first present a definition describing a basic notion of stability for finite horizon discrete time systems.

DEFINITION 4.3. *The autonomous finite horizon discrete time system $x(t+1) = A(t)x(t)$ for $t = 0, \ldots, T-1$ is said to be* uniformly decreasing *if $\|x(t+1)\| < \|x(t)\|$ for each time t and independent of the initial condition.*

The existence of a desirable step-size can be stated as a feasibility problem of a corresponding semidefinite program. In particular, if the eigenvalues of the state matrix for (4.5) lie strictly in the unit disc for all time, then the error will be uniformly decreasing over the time horizon. This can be stated as an LMI condition.

LEMMA 4.4. *There exists a step-size $\alpha(t)$ such that the error dynamics in* (4.5) *is uniformly decreasing if and only if the following LMI condition is feasible:*

$$-I \leq L_t^{1/2} L_{t+1}^{-1} L_t^{1/2} - \alpha(t) L_t \leq I \tag{4.6}$$

with $L_t = E'Q^{-1}P(\tilde{T})E$.

Proof. The state matrix in (4.5) is similar to the symmetric matrix $L_t^{1/2}L_{t+1}^{-1}L_t^{1/2} - \alpha(t)L_t$. The LMI in (4.6) ensures the eigenvalues of (4.5) lie in the interval $(-1, 1)$. ◻

It is worth emphasizing that this LMI condition depends only on the communication graph and the LQR gains of each agent. For the special case where each agent has identical state and control weights, but different preference values, we can obtain an analytic condition indicating whether the LMI in (4.6) is feasible.

COROLLARY 4.5. *Assume $Q = qI$ and $R = rI$. Then there exists a step-size $\alpha(t)$ such that* (4.5) *is uniformly decreasing if and only if*

$$\frac{\lambda_{\max}(E'E)}{\lambda_{\min}(E'E)} < 3 + 2\left(\left(\frac{q}{r}\right)^2 + 3\frac{q}{r}\right).$$

The proof is provided in Appendix C. When this condition is satisfied, we obtain the following interval for step-sizes leading to uniformly decreasing error dynamics:[3]

$$\frac{q}{\lambda_{\min}(E'E)P(\tilde{T})}\left(\frac{P(\tilde{T})}{P(\tilde{T}-1)} - 1\right) < \alpha(t) < \frac{q}{\lambda_{\max}(E'E)P(\tilde{T})}\left(\frac{P(\tilde{T})}{P(\tilde{T}-1)} + 1\right). \tag{4.7}$$

In fact, the lower and upper bounds can be obtained explicitly from the recursion (3.5). Due to the monotonicity of the recursion, the lower and upper bounds will be obtained at either the beginning or the end of the recursion (i.e., at $P(1)$ and $P(2)$ or $P(T-1)$ and $P(T-2)$), and this depends on whether the ratio $r_i q_i^{-1}$ is greater or less than unity.

This result is compelling for a few reasons. On the one hand, it provides a relatively easy and tractable way to find (if one exists) a step-size. This is in stark contrast to static subgradient methods, where the choice of the step-size can be nontrivial. On the other hand, and contrary to static subgradient methods, this highlights that under certain conditions there will *not* be a step-size that can guarantee a uniformly decreasing error dynamics. Furthermore, the importance of the communication graph itself is highlighted in this result. Indeed, for general graphs including cycles, the eigenvalue ratio is related to graph *expanders* [5]. For the case of spanning trees, this ratio has not been explored much in the literature, and this result points to an interesting connection between optimization performance and structural properties of the graph.

The above results relate to the performance of the error system for the multiplier values. It is also worth investigating how this impacts the error of the primal system and, in particular, the error of the terminal state as described in (4.1). Recall that the algorithm computes at each time step a prediction of the terminal state $\hat{\mathbf{x}}^t(T)$ and uses the next-step optimal control to propagate the state forward at each time. Therefore, the terminal state $x(T)$ is precisely equal to the predicted state $\hat{\mathbf{x}}^{T-1}(T)$ at the last step in the algorithm. This motivates a study of the "predicted disagreement" for the system

$$\mathbf{e}(t) = E'\hat{\mathbf{x}}^t(T). \tag{4.8}$$

It is clear that $E'x(T) = \mathbf{e}(T-1)$.

[3] Note that when the control and state weights are identical, the LQR gains $P_i(\tilde{T})$ are also identical.

THEOREM 4.6. *The predicted disagreement* $\mathbf{e}(t) = E'\hat{\mathbf{x}}^t(T)$ *evolves according to the LTV dynamics*

$$\mathbf{e}(t+1) = \Big(I - \alpha(t)E'Q^{-1}P(\tilde{T}-1)E\Big)\,\mathbf{e}(t). \tag{4.9}$$

Proof. Using (3.8), we can express the predicted disagreement at time t, $\mathbf{e}(t) = E'\hat{\mathbf{x}}^t(T)$, in terms of the multiplier value $\mu(t)$. Alternatively, we can express this term as a function of the estimation error $\epsilon(t)$ as

$$\begin{aligned}\mathbf{e}(t) &= E'K(\tilde{T})\,(x(t)-\xi) - E'Q^{-1}P(\tilde{T})E\mu(t) + E'\xi\\ &= E'K(\tilde{T})(x(t)-\xi) - E'Q^{-1}P(\tilde{T})E\overline{\mu}^{(\tilde{T},x(t))} + E'\xi - E'Q^{-1}P(\tilde{T})E\epsilon(t).\end{aligned} \tag{4.10}$$

Note that if the optimal multiplier $\overline{\mu}^{(\tilde{T},x(t))}$ is used to compute the state trajectories at time t, then the final consensus error will be identically zero. (This is equivalent to the centralized solution to $OCP(t,T,x(t))$.) Therefore, all terms except the last one in (4.10) vanish, and what remains is an expression relating the predicted terminal state error as a function of the multiplier error, $\epsilon(\tilde{T},x(t))$,

$$\mathbf{e}(t) = -E'Q^{-1}P(\tilde{T})E\epsilon(\tilde{T},x(t)). \tag{4.11}$$

Propagating the error state forward and using the dynamics for the multiplier error dynamics in (4.5) leads to the desired result. □

This result can be used to determine bounds on the agreement error (4.1) by examining the state-transition matrix. We note that these bounds will depend explicitly on the parameter $\alpha(t)$, the properties of the graph, and the LQR gains for each agent. In this work, we focus on how this result leads to a new condition for the step-size rule $\alpha(t)$ that guarantees the error $\mathbf{e}(t)$ is uniformly strictly decreasing.

COROLLARY 4.7. *The predicted terminal disagreement* $\|\mathbf{e}(t)\|$ *is uniformly decreasing if and only if*

$$0 < \alpha(t) < 2\lambda_{\max}^{-1}(E'Q^{-1}P(\tilde{T}-1)E), \tag{4.12}$$

where $\lambda_{\max}(E'Q^{-1}P(\tilde{T}-1)E)$ *is the largest eigenvalue of the positive definite matrix* $E'Q^{-1}P(\tilde{T}-1)E$.

Proof. For $\alpha(t)$ satisfying (4.12), all eigenvalues of the symmetric matrix $(I - \alpha(t)E'Q^{-1}P(\tilde{T}-1)E)$ are located between $(-1,1)$. □

This result has several implications. Depending on the problem parameters it might be impossible to achieve a uniform decrease of the dual error $\|\epsilon(t)\|$ while still being possible to achieve a uniform decrease of the predicted final disagreement $\|\mathrm{e}(t)\|$. However, in order to achieve perfect agreement, e.g., $E'x(T) = 0$, it is necessary that $\epsilon(\tilde{T}-1) = 0$. This suggests that although a step-size can be chosen to ensure the consensus error decreases at all times, it cannot guarantee that the error is made *arbitrarily small* in finite time.

As with the multiplier error dynamics, we are also able to state a stronger result when the control and state weights for each agent are identical.

COROLLARY 4.8. *Assume* $Q = qI$ *and* $R = rI$. *Then there exists a step-size* $\alpha(t)$ *that achieves a uniform decrease of* $\|\mathbf{e}(t)\|$ *if and only if*

$$0 < \alpha(t) < 2\frac{q}{P(T-1)\lambda_{\max}(E'E)},\ t = 0,\dots,T-1.$$

Here we observe that the largest eigenvalue of the graph Laplacian will limit the range of allowable step-sizes. This is in contrast to the case for the multiplier error dynamics, which also depends on the smallest eigenvalue.

Our discussion until now has assumed the horizon time is given in the problem formulation. It is also worth investigating the complementary question asking what is the minimum horizon time that can guarantee the SHPA terminates with a terminal disagreement less than some bound.

COROLLARY 4.9. *Consider the SHPA algorithm over a fixed spanning tree $\mathcal{G}$ with a constant step-size α satisfying Corollary* 4.7. *For any $\delta > 0$ with $\delta < \|e(0)\|$, there exists a $T(\delta)$ satisfying*

$$T(\delta) = \frac{\ln\left(\frac{\delta}{\|e(0)\|}\right)}{\ln\left(1 - \alpha\lambda_{\min}(E'Q^{-1}\overline{P}E)\right)} + 1$$

with $0 < \overline{P} \leq P(\tilde{T})$ for all t such that for any $T \geq T(\delta)$, the terminal disagreement error $e(T-1) = E'x(T)$ satisfies the bound $\|e(T-1)\| < \delta$.

Proof. The critical horizon time $T(\delta)$ can be obtained directly from the dynamics of the predicted disagreement (4.9). In particular, we have that

$$e(T-1) = \prod_{t=0}^{T-2}\left(I - \alpha E'Q^{-1}P(\tilde{T}-1)E\right)e(0),$$

and using the submultiplicative property of the spectral norm we obtain

$$\|e(T-1)\| \leq \left(\prod_{t=0}^{T-2}\left\|\left(I - \alpha E'Q^{-1}P(\tilde{T}-1)E\right)\right\|\right)\|e(0)\|.$$

It is straightforward to verify that there exists a lower bound on the recursions given in (3.5) that depends only on the weights q_i and r_i and is independent of the horizon T. Stated in another way, there exists a matrix $\overline{P} > 0$ that depends only on q_i and r_i such that $\overline{P} \leq P(\tilde{T})$ for all t and T. This in turn implies that $\|(I - \alpha E'Q^{-1}P(\tilde{T}-1)E)\| \leq \|(I - \alpha E'Q^{-1}\overline{P}E)\|$ for all t and T, leading to

$$\|e(T)\| \leq \left(1 - \alpha\lambda_{\min}(E'Q^{-1}\overline{P}E)\right)^{T-1}\|e(0)\| \leq \delta.$$

Taking the logarithm of the above expression and solving for T leads to the desired bound. □

Corollary 4.9 provides a condition for determining if a specified horizon is "long enough" for the SHPA algorithm to terminate within a certain tolerance of an agreement state. Note that this bound will depend explicitly on the choice of the spanning tree, the control and state weights of each agent, and the initial conditions. In fact, as $\lambda_{\min}(E'Q^{-1}\overline{P}E)$ corresponds to the smallest nontrivial eigenvalue of a weighted Laplacian matrix [19], one can conclude that trees with large algebraic connectivity will lead to shorter horizon times for a given bound on δ.

We conclude this section with a brief discussion on the optimality gap between the trajectories generated by $OCP(t_0, T, x(t_0))$ and the SHPA algorithm. As discussed, the SHPA cannot be expected to achieve an exact consensus in T steps. However, Corollary 4.9 also implies that as the horizon time grows, the terminal agreement error can be arbitrarily close to zero. This mimics the behavior of the traditional subgradient methods that only achieve the optimal solution asymptotically.

To compare the different trajectories in a meaningful way, we must consider the evaluation of the Lagrangian function (2.6) along the different trajectories. To simplify the discussion, we denote the optimal trajectories generated by $OCP(t_0, T, x(t_0))$ as $(\overline{\mathbf{x}}, \overline{\mathbf{u}})$, and the trajectories generated by the SHPA algorithm as $(x(t), u(t))$.[4] For evaluating the Lagrangian, we also use the optimal multiplier value associated with $OCP(t_0, T, x(t_0))$, denoted $\overline{\mu}$. The first observation that follows directly from the discussion of section 2.1 is that if the multiplier state in the SHPA algorithm is initialized as $\mu(t_0) = \overline{\mu}$, then the generated trajectories will be optimal; that is, $\mathcal{L}(\overline{\mathbf{x}}, \overline{\mathbf{u}}, \overline{\mu}) = \mathcal{L}(x, u, \overline{\mu})$. For any other initial condition, one must have that $\mathcal{L}(\overline{\mathbf{x}}, \overline{\mathbf{u}}, \overline{\mu}) < \mathcal{L}(x, u, \overline{\mu})$. Therefore, the optimality gap can be characterized by the ratio

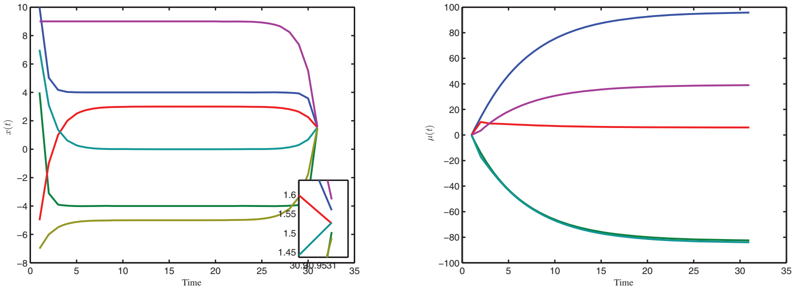
$$\Delta = \frac{\mathcal{L}(x, u, \overline{\mu})}{\mathcal{L}(\overline{\mathbf{x}}, \overline{\mathbf{u}}, \overline{\mu})}. \tag{4.13}$$

The performance of the SHPA algorithm can be measured using this gap and will depend on the horizon time T, the state and control weights for each agent, and the spectrum of the communication graph. The saddle-point property stated above guarantees that this ratio will always be greater than unity, providing a clear measure of suboptimality. The effects of these system parameters will be explored in the following via simulation.

5. Simulation results.

5. Simulation results. We provide two different simulations to support our results. In both studies we simulate a system with six agents over a time horizon of 30 steps. The communication graph is chosen as a random tree in each simulation. Each agent has a unique initial condition, preference state, and individual weights for its state and control, which are all chosen randomly. We use a constant step-size which is chosen such that, at the initial time, the eigenvalues of $E'Q^{-1}P(1)E$ are contained in the unit disc.

For the first simulation, which is shown in Figures 5.1 and 5.2, the values of q_i and r_i are chosen randomly as integers from the interval $[0, 10]$. The constant step-size used for the simulations is $\alpha = 3.4025$. Figure 5.1 shows the $x(t)$ and $\mu(t)$ trajectories produced by Algorithm 1. Note that the agents approach the consensus state but do not attain it exactly. Figure 5.2 shows the two error functions, $\epsilon(t)$ and $\mathbf{e}(t)$. Both

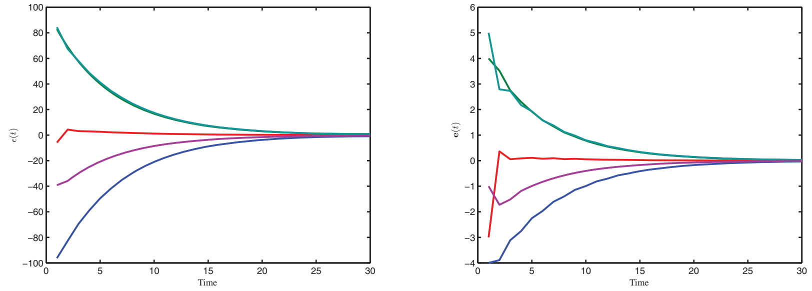


(a)

(b)

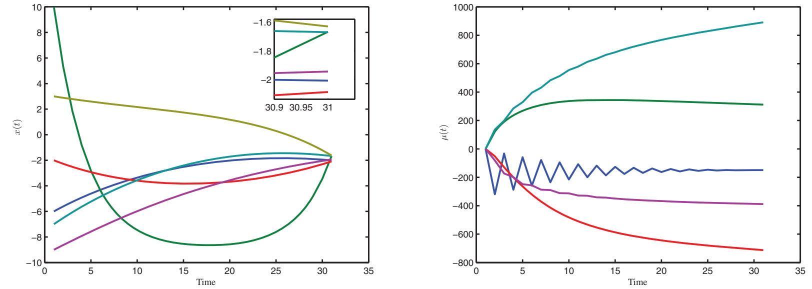
FIG. 5.1. *Simulation of SHPA algorithm with r_i of $\mathcal{O}(1)$;* (a) *trajectories for $x(t)$ and* (b) *trajectories for $\mu(t)$.*

[4] The control generated by the SHPA algorithm can be obtained from the states $x(t)$ and $\mu(t)$ using (3.7).



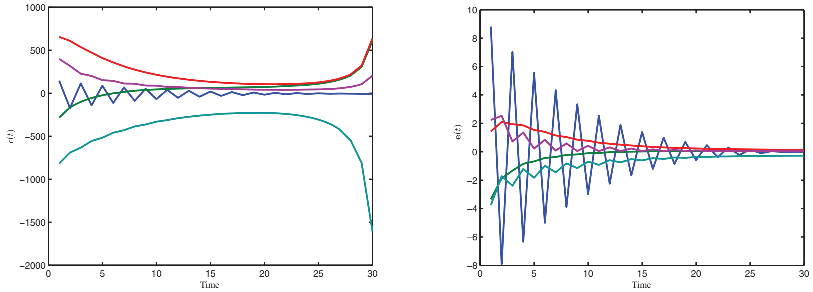
(a) (b)

FIG. 5.2. *Dual error* $\epsilon(t)$ *and predicted terminal disagreement* $\mathbf{e}(t)$ *for the SHPA algorithm with* r_i *of* $\mathcal{O}(1)$.



(a) (b)

FIG. 5.3. *Simulation of SHPA algorithm with* r_i *of* $\mathcal{O}(10^4)$*:* (a) *trajectories of* $x(t)$ *and* (b) *trajectories of* $\mu(t)$.



(a) (b)

FIG. 5.4. *Dual error* $\epsilon(t)$ *and predicted terminal disagreement* $\mathbf{e}(t)$ *for the SHPA algorithm, with* r_i *of* $\mathcal{O}(10^4)$*. Note the unstable trajectory of* $\epsilon(t)$ *while* $\mathbf{e}(t)$ *is uniformly decreasing.*

functions are uniformly decreasing. One can verify that both conditions 4.12 and 4.6 are satisfied at all times.

The second simulation, illustrated in Figures 5.3 and 5.4, shows a different result. The problem setup is similar, i.e., initial conditions, preferences, and state weights q_i are randomly chosen in the same order of magnitude as for the first simulation.

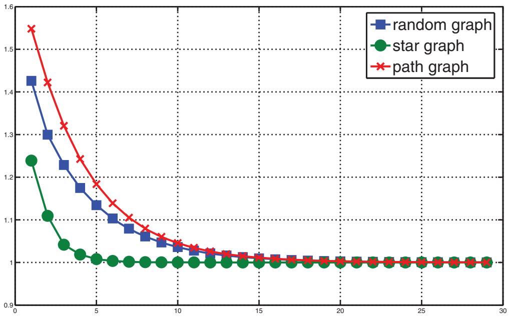


FIG. 5.5. *A plot of the optimality ratio between the optimal trajectories generated by $OCP(t_0, T, x(t_0))$ and the trajectories generated by the SHPA algorithm. The trajectories were computed for increasing time horizons T (x-axis) and for three different graphs.*

But the control weights r_i are now chosen as integers from the interval $[0, 10^4]$. The step-size used for this simulation is $\alpha = 36.2263$. The primal trajectories $x(t)$ still approximate a consensus state at the end of the horizon. However, it can be clearly seen that the dual error $\epsilon(t)$ is increasing as the time is approaching the end of the horizon. It can be verified that the LMI condition 4.6 is infeasible for $t > 20$. However, condition 4.12 is satisfied at all times and the error $\mathbf{e}(t)$ is uniformly decreasing. These results are fully according to the theoretical analysis provided in section 4.

Finally, we provide a simulation illustrating the implications of Corollary 4.9 and the discussion on the optimality gap Δ given in (4.13). Figure 5.5 plots Δ as a function of the horizon time T for different trajectories using three different graphs: the path graph, the star graph, and a random graph (all on five nodes). Observe that as the horizon increases, the ratio approaches unity, while the structure of the communication graph affects the rate of convergence.

6. Concluding remarks.

This work presented a real-time and distributed algorithm for solving a finite-time preference-based consensus problem. In this setting, the finite-time horizon becomes a critical feature that motivated our proposed algorithm. In particular, the algorithm negotiates the consensus value in real time while simultaneously propagating the physical state of each agent along a trajectory believed to be optimal. This is in contrast to methods that require the consensus state to be negotiated *before* the agents are able to move along their trajectories.

The SHPA algorithm is closely related to dual subgradient methods. We showed that our proposed algorithm is equivalent to an LTV dynamical system, parameterized by a step-size variable. The LTV system also contains a set of augmented states representing a dynamic version of the Lagrange multiplier associated with the terminal constraint. Analysis of the SHPA algorithm reveals a close connection between the Lagrange multiplier associated with a centralized version of the preference-based agreement protocol and the trajectories of the multipliers generated by the LTV description. This leads to an LTV error system description for both the multipliers and the terminal state estimate. Analysis of these systems provides conditions for when a step-size rule exists that can guarantee the error is uniformly decreasing. In particular, we show that for certain instances of the problem, e.g., certain communication graphs and state and control weights, no step-size rule exists that guarantees uniform decreasing of the multiplier error dynamics.

One of the more salient features of this work is the numerically tractable and conceptually simple guideline for checking for the existence of and choosing a step-size rule for achieving good performance. This is in stark contrast to general methods for subgradient algorithms, where choice of the step-size rule can be nontrivial. Furthermore, the role the communication graph plays in this setting points to a new role for the largest and smallest eigenvalues of the graph Laplacian. The presented work provides a foundation for considering various extensions, including the effects of a state-dependent or switching communication structure and more general agent dynamics.

Appendix A. Proof of Theorem 3.1. We begin by presenting an explicit solution to $QP_i(t)$, as stated in (3.1). Note that the subproblems solved at time t are linear quadratic OCPs with the initial condition $x(t)$ and the time horizon $\tilde{T} = T - t$. As noted in section 2.2, these problems are linearly constrained quadratic programs and therefore admit an analytic solution [3]. The SHPA algorithm requires the next-step optimal control input and the terminal state value,

$$\hat{\mathbf{u}}_i^t(t) = e'_{1,\tilde{T}}\hat{\mathbf{u}}_i^t, \quad \hat{\mathbf{x}}_i^t(T) = e'_{\tilde{T},\tilde{T}}\hat{\mathbf{x}}_i^t.$$

We can now express $\hat{\mathbf{u}}_i^t(t)$ and $\hat{\mathbf{x}}_i^t(T)$ from the analytic solution of $QP_i(t)$ as

$$\begin{aligned}\hat{\mathbf{u}}_i^t(t) = {} & -r_i^{-1}e'_{1,\tilde{T}}B'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}}x_i(t) + r_i^{-1}e'_{1,\tilde{T}}B'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}}\xi_i \\ & - r_i^{-1}q_i^{-1}e'_{1,\tilde{T}}B'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}}\gamma_i(t),\end{aligned} \tag{A.1}$$

$$\begin{aligned}\hat{\mathbf{x}}_i^t(T) = {} & q_i^{-1}e'_{\tilde{T},\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}}x_i(t) - (q_i^{-1}e'_{\tilde{T},\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}} - 1)\xi_i \\ & + q_i^{-1}(q_i^{-1}e'_{\tilde{T},\tilde{T}}\Gamma_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}} - 1)\gamma_i(t).\end{aligned} \tag{A.2}$$

The matrix $\Gamma_i(\tilde{T})$ for each agent, parameterized by the horizon $\tilde{T}$, appears in the analytic solution of $QP_i(t)$ and is defined as

$$\Gamma_i(\tilde{T}) := q_i^{-1}I_{\tilde{T}} + r_i^{-1}B_{\tilde{T}}B'_{\tilde{T}}. \tag{A.3}$$

Note that these expressions are functions of the current state $x_i(t)$, the preference value ξ_i, and the multiplier value associated with each node $\gamma_i(t)$. We can improve this analytic description by noting that the elements of the matrix $\Gamma_i^{-1}(\tilde{T})$ used in (A.1) and (A.2) evolve according to a recursion. Therefore, we now examine the following four time-dependent constants:

$$e'_{1,\tilde{T}}B'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}}, \quad e'_{1,\tilde{T}}B'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}}, \quad e'_{\tilde{T},\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}}, \quad e'_{\tilde{T},\tilde{T}}\Gamma_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}}.$$

The structure of $B_{\tilde{T}}$ allows us to precisely characterize $\Gamma_i^{-1}(\tilde{T})$. To begin, we employ the matrix inverse identity $(I + D^{-1})^{-1} = D(I + D)^{-1}$ [12] to obtain

$$\Gamma_i^{-1}(\tilde{T}) = r_i\left(B_{\tilde{T}}B'_{\tilde{T}}\right)^{-1}\left(I_{\tilde{T}} + r_iq_i^{-1}\left(B_{\tilde{T}}B'_{\tilde{T}}\right)^{-1}\right)^{-1} = r_i\left(B_{\tilde{T}}B'_{\tilde{T}}\right)^{-1}F_i^{-1}(\tilde{T}). \tag{A.4}$$

We now note that $(B_{\tilde{T}}B'_{\tilde{T}})^{-1}$ is a tridiagonal matrix (this is shown, e.g., in [18]),

$$\left(B_{\tilde{T}}B'_{\tilde{T}}\right)^{-1} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}. \tag{A.5}$$

We focus the analysis now on the four expressions related to the matrix $F_i^{-1}(\tilde{T})$:

$$\begin{aligned} P_i(\tilde{T}) &= e'_{\tilde{T},\tilde{T}}F_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}}, \qquad & S_i(\tilde{T}) &= e'_{1,\tilde{T}}F_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}}, \\ U_i(\tilde{T}) &= e_{\tilde{T},\tilde{T}}F_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}}, \qquad & W_i(\tilde{T}) &= e'_{1,\tilde{T}}F_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}}. \end{aligned}$$

Each of these expressions selects a specific element or a row sum of the matrix $F_i^{-1}(\tilde{T})$. We aim to derive recursive formulas of the form $P_i(\tilde{T}+1) = f(P_i(\tilde{T}))$ for the computation of the four variables. Our approach utilizes ideas presented in [16, Chap. 17] for the derivation of Kalman filters. First observe that $F_i(\tilde{T}+1)$ is related to $F_i(\tilde{T})$ via a rank-one matrix update and bordering by one row and column:

$$F_i(\tilde{T}+1) = \left[\begin{array}{c|c} F_i(\tilde{T}) + r_i q_i^{-1} e_{\tilde{T},\tilde{T}} e'_{\tilde{T},\tilde{T}} & -r_i q_i^{-1} e_{\tilde{T},\tilde{T}} \\ \hline -r_i q_i^{-1} e'_{\tilde{T},\tilde{T}} & 1 + r_i q_i^{-1} \end{array}\right] =: \begin{bmatrix} \mathcal{T} & \mathcal{U} \\ \mathcal{V} & \mathcal{W} \end{bmatrix}. \tag{A.6}$$

The inverse is given as

$$F_i^{-1}(\tilde{T}+1) := \begin{bmatrix} \mathcal{L} & \mathcal{M} \\ \mathcal{N} & \mathcal{P} \end{bmatrix} = \begin{bmatrix} \mathcal{T}^{-1} + \mathcal{T}^{-1}\mathcal{U}\mathcal{P}\mathcal{V}\mathcal{T}^{-1} & -\mathcal{T}^{-1}\mathcal{U}\mathcal{P} \\ -\mathcal{P}\mathcal{V}\mathcal{T}^{-1} & (\mathcal{W} - \mathcal{V}\mathcal{T}^{-1}\mathcal{U})^{-1} \end{bmatrix}. \tag{A.7}$$

Computing $F_i^{-1}(\tilde{T}+1)$ requires computation of $\mathcal{T}^{-1}$. One can use the *Sherman–Morrison–Woodbury–Schur* formula [16] to express $\mathcal{T}^{-1}$ as follows:

$$\mathcal{T}^{-1} = F_i^{-1}(\tilde{T}) - r_i q_i^{-1} \frac{F_i^{-1}(\tilde{T}) e_{\tilde{T},\tilde{T}} e'_{\tilde{T},\tilde{T}} F_i^{-1}(\tilde{T})}{1 + r_i q_i^{-1} e'_{\tilde{T},\tilde{T}} F_i^{-1}(\tilde{T}) e_{\tilde{T},\tilde{T}}}. \tag{A.8}$$

This leads to an expression of $F_i^{-1}(\tilde{T}+1)$ in terms of $F_i^{-1}(\tilde{T})$. Consider the term $P_i(\tilde{T}) = e'_{\tilde{T},\tilde{T}}F_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}}$ and note that

$$\begin{aligned} &P_i(\tilde{T}+1) \\ &\quad = (\mathcal{W} - \mathcal{V}\mathcal{T}^{-1}\mathcal{U})^{-1} = (1 + r_i q_i^{-1} - (r_i q_i^{-1})^2 e'_{\tilde{T},\tilde{T}} \mathcal{T}^{-1} e_{\tilde{T},\tilde{T}})^{-1} \\ &\quad = \left(1 + r_i q_i^{-1} - (r_i q_i^{-1})^2 e'_{\tilde{T},\tilde{T}} \left(F_i^{-1}(\tilde{T}) - r_i q_i^{-1} \frac{F_i^{-1}(\tilde{T}) e_{\tilde{T},\tilde{T}} e'_{\tilde{T},\tilde{T}} F_i^{-1}(\tilde{T})}{1 + r_i q_i^{-1} e'_{\tilde{T},\tilde{T}} F_i^{-1}(\tilde{T}) e_{\tilde{T},\tilde{T}}}\right) e_{\tilde{T},\tilde{T}}\right)^{-1} \\ &\quad = \left(1 + r_i q_i^{-1} - (r_i q_i^{-1})^2 (P_i(\tilde{T}) - r_i q_i^{-1} \frac{P_i(\tilde{T}) P_i(\tilde{T})}{1 + r_i q_i^{-1} P_i(\tilde{T})})\right)^{-1} \\ &\quad = \frac{1 + r_i q_i^{-1} P_i(\tilde{T})}{1 + r_i q_i^{-1} + r_i q_i^{-1} P_i(\tilde{T})}. \end{aligned}$$

By considering additionally that $P_i(1) = (1 + r_i q_i^{-1})^{-1}$, we have found a recursive formula for the computation of $P_i(\tilde{T})$. Following a similar procedure, recursions for the other three variables can be defined and are summarized in Table A.1. Additionally, we introduce a fifth variable, which allows further simplifications later on. Define

$$K_i(\tilde{T}) = 1 - U_i(\tilde{T}) = 1 - e'_{\tilde{T},\tilde{T}} F_i^{-1}(\tilde{T}) e_{\tilde{T},\tilde{T}}. \tag{A.9}$$

A recursion for $K_i(\tilde{T})$ can be directly derived from the recursion for $U_i(\tilde{T})$ and is given in Table A.1.

TABLE A.1
Recursions for computation of the variables associated with Theorem 3.1.

$P_i(\tilde{T}+1) = \frac{1+r_iq_i^{-1}P_i(\tilde{T})}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})}$	$S_i(\tilde{T}+1) = r_iq_i^{-1}\frac{S_i(\tilde{T})}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})}$
$U_i(\tilde{T}+1) = \frac{1+r_iq_i^{-1}(U_i(\tilde{T})+P_i(\tilde{T}))}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})}$	$K_i(\tilde{T}+1) = r_iq_i^{-1}\frac{K_i(\tilde{T})}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})}$
$W_i(\tilde{T}+1) = W_i(\tilde{T}) + r_iq_i^{-1}S_i(\tilde{T})\dfrac{1-U_i(\tilde{T})}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})}$	
$P_i(1) = S_i(1) = U_i(1) = W_i(1) = (1+r_iq_i^{-1})^{-1}, K_i(1) = 1-(1+r_iq_i^{-1})^{-1}$	

The variables in Table A.1 are intrinsically related to each other, summarized in the following proposition.

PROPOSITION A.1. *The time-varying constants provided in Table* A.1 *satisfy the following:*

$$\begin{array}{llrl}
(A.10) & \text{(i)} & r_iS_i(\tilde{T}) & = q_iK_i(\tilde{T}), \\
(A.11) & \text{(ii)} & W_i(\tilde{T}) & = P_i(\tilde{T}), \\
(A.12) & \text{(iii)} & r_i^{-1}e'_{1,\tilde{T}}B'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}} & = P_i(\tilde{T}), \\
(A.13) & \text{(iv)} & r_i^{-1}q_i^{-1}e'_{1,\tilde{T}}B'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}} & = r_i^{-1}K_i(\tilde{T}), \\
(A.14) & \text{(v)} & q_i^{-1}e'_{\tilde{T},\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}} & = K_i(\tilde{T}), \\
(A.15) & \text{(vi)} & 1-q_i^{-1}e'_{\tilde{T},\tilde{T}}\Gamma_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}} & = P_i(\tilde{T}).
\end{array}$$

Proof. (i) It can be directly seen that both obey the same recursion, since $K_i(1) = r_i(q_i+r_i)^{-1} = r_iq_i^{-1}S_i(1)$.

(ii) We show the statement by induction. We have already shown the base case holds, $W_i(1) = P_i(1)$, so we proceed to the inductive step, assuming $P_i(\tilde{T}) = W_i(\tilde{T})$. We work with the expression $W_i(\tilde{T}+1)$ using the recursion

$$W_i(\tilde{T}+1) = W_i(\tilde{T}) + \frac{r_iq_i^{-1}S_i(\tilde{T})(1-U_i(\tilde{T}))}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})}.$$

We now invoke (A.9) and (A.10) stating $S_i(\tilde{T}) = q_ir_i^{-1}(1-U_i(\tilde{T}))$ along with the inductive step to obtain

$$\begin{aligned}
W_i(\tilde{T}+1) &= P_i(\tilde{T}) + \frac{(1-U_i(\tilde{T}))^2}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})} \\
&= \frac{1+r_iq_i^{-1}P_i(\tilde{T})}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})} + \frac{P_i(\tilde{T})(1+r_iq_i^{-1}P_i(\tilde{T}))+(1-U_i(\tilde{T}))^2-1}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})} \\
&= P_i(\tilde{T}+1) + (1+r_iq_i^{-1}+r_iq_i^{-1}P_i(t))^{-1}\Big(P_i(\tilde{T})(1+r_iq_i^{-1}P_i(\tilde{T})) \\
&\qquad\qquad + U_i(\tilde{T})(U_i(\tilde{T})-2)\Big).
\end{aligned}$$

We show in the next step that $P_i(\tilde{T})(1+r_iq_i^{-1}P_i(\tilde{T})) = U_i(\tilde{T})(2-U_i(\tilde{T}))$, which implies that the term

$$(1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T}))^{-1}\Big(P_i(\tilde{T})(1+r_iq_i^{-1}P_i(\tilde{T}))+U_i(\tilde{T})(U_i(\tilde{T})-2)\Big) = 0.$$

The base case is easily verified:

$$\begin{aligned}
P_i(1)(1+r_iq_i^{-1}P_i(1)) &= (1+r_iq_i^{-1})^{-1}(1+r_iq_i^{-1}(1+r_iq_i^{-1})^{-1}) = \frac{1+2r_iq_i^{-1}}{(1+r_iq_i^{-1})^2}\\
&= \frac{1+2r_iq_i^{-1}+1-1}{(1+r_iq_i^{-1})^2} = \frac{2+2r_iq_i^{-1}}{(1+r_iq_i^{-1})^2} + \frac{-1}{(1+r_iq_i^{-1})^2}\\
&= (1+r_iq_i^{-1})^{-1}(2-(1+r_iq_i^{-1})^{-1})\\
&= U_i(1)(2-U_i(1)).
\end{aligned}$$

Next we show the inductive step,

$$\begin{aligned}
&P_i(\tilde{T}+1)(1+r_iq_i^{-1}P_i(\tilde{T}+1))\\
&= \left(\frac{1+r_iq_i^{-1}P_i(\tilde{T})}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})}\right)\left(1+r_iq_i^{-1}\frac{1+r_iq_i^{-1}P_i(\tilde{T})}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})}\right)\\
&= \frac{(1+2r_iq_i^{-1})(1+r_iq_i^{-1}P_i(\tilde{T}))+r_iq_i^{-1}U_i(\tilde{T})(2-U_i(\tilde{T}))+(r_iq_i^{-1})^2U_i(\tilde{T})(2-U_i(\tilde{T}))}{(1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T}))^2},
\end{aligned} \tag{A.16}$$

$$\begin{aligned}
&U_i(\tilde{T}+1)(2-U_i(\tilde{T}+1))\\
&= \left(\frac{1+r_iq_i^{-1}P_i(\tilde{T})+r_iq_i^{-1}U_i(\tilde{T})}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})}\right)\times\left(\frac{1+2r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})-r_iq_i^{-1}U_i(\tilde{T})}{1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T})}\right)\\
&= \frac{(1+2r_iq_i^{-1})(1+r_iq_i^{-1}P_i(\tilde{T}))+r_iq_i^{-1}U_i(\tilde{T})(2-U_i(\tilde{T}))+(r_iq_i^{-1})^2U_i(\tilde{T})(2-U_i(\tilde{T}))}{(1+r_iq_i^{-1}+r_iq_i^{-1}P_i(\tilde{T}))^2}.
\end{aligned} \tag{A.17}$$

This concludes the inductive proof, as (A.16) is equal to (A.17).

(iii)

$$r_i^{-1}e'_{1,\tilde{T}}B'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}} = r_i^{-1}\mathbb{1}'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}} = r_i^{-1}r_ie'_{1,\tilde{T}}F_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}} = W_i(\tilde{T}) = P_i(\tilde{T}).$$

(iv)

$$\begin{aligned}
r_i^{-1}q_i^{-1}e'_{1,\tilde{T}}B'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}} &= r_i^{-1}q_i^{-1}\mathbb{1}'_{\tilde{T}}\Gamma_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}} = q_i^{-1}e'_{1,\tilde{T}}F_i^{-1}(\tilde{T})e_{\tilde{T},\tilde{T}}\\
&= q_i^{-1}S_i(\tilde{T}) = r_i^{-1}K_i(\tilde{T}).
\end{aligned}$$

(v) We define a recursion for the expression on the left side and show that it is equivalent to the recursion for $K_i(\tilde{T})$. Note that

$$\begin{aligned}
q_i^{-1}e'_{\tilde{T},\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}} &= q_i^{-1}r_i\begin{bmatrix} 0 & \cdots & 0 & -1 & 1\end{bmatrix}F_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}}\\
&= r_iq_i^{-1}(e'_{\tilde{T},\tilde{T}}F_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}} - e'_{\tilde{T}-1,\tilde{T}}F_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}}).
\end{aligned}$$

Define the auxiliary variable $v_i^1(\tilde{T}) = q_i^{-1}e'_{\tilde{T},\tilde{T}}\Gamma_i^{-1}(\tilde{T})\mathbb{1}_{\tilde{T}}$ and note that $v_i^1(1) = r_i(q_i+r_i)^{-1} = W_i(1)$. We must show that $v_i^1(\tilde{T}+1) = W_i(\tilde{T}+1)$. Note that

$$v_i^1(\tilde{T}+1) = r_iq_i^{-1}\left(e'_{\tilde{T}+1,\tilde{T}+1}F_i^{-1}(\tilde{T}+1)\mathbb{1}_{\tilde{T}+1} - e'_{\tilde{T},\tilde{T}+1}F_i^{-1}(\tilde{T}+1)\mathbb{1}_{\tilde{T}+1}\right).$$

Now the expressions (A.6), (A.7), and (A.8) can be used to obtain (after algebraic manipulation)

$$\begin{aligned} v_i^1(\tilde{T}+1) &= r_i q_i^{-1} U_i(\tilde{T}+1) - r_i q_i^{-1} e'_{\tilde{T},\tilde{T}}(\mathcal{T}^{-1} + \mathcal{T}^{-1}\mathcal{U}P_i(\tilde{T}+1)\mathcal{V}\mathcal{T}^{-1})\mathbb{1}_{\tilde{T}} \\ &\qquad + r_i q_i^{-1} e'_{\tilde{T},\tilde{T}}(-\mathcal{T}^{-1}\mathcal{U}P_i(\tilde{T}+1)) \\ &= r_i q_i^{-1} \frac{1 - U_i(\tilde{T})}{1 + r_i q_i^{-1} + r_i q_i^{-1} P_i(\tilde{T})} \\ &= r_i q_i^{-1} \frac{K_i(\tilde{T})}{1 + r_i q_i^{-1} + r_i q_i^{-1} P_i(\tilde{T})} = K_i(\tilde{T}+1). \end{aligned}$$

This gives the desired conclusion.

(vi) We show this relation again by induction. Define $v_i^2(\tilde{T}) = 1 - q_i^{-1} e'_{\tilde{T},\tilde{T}} \Gamma_i^{-1} (\tilde{T}) e_{\tilde{T},\tilde{T}}$. Note that $v_i^2(1) = 1 - q_i^{-1}(q_i^{-1} + r_i^{-1})^{-1} = P_i(1)$. Again, after some algebraic manipulation, we arrive at

$$\begin{aligned} v_i^2(\tilde{T}+1) &= 1 - q_i^{-1}\left(e'_{\tilde{T}+1,\tilde{T}+1} F_i^{-1}(\tilde{T}+1) e_{\tilde{T}+1,\tilde{T}+1} - e'_{\tilde{T},\tilde{T}+1} F_i^{-1}(\tilde{T}+1) e_{\tilde{T}+1,\tilde{T}+1}\right) \\ &= 1 - q_i^{-1} P_i(\tilde{T}+1) + q_i^{-1}\left(e'_{\tilde{T},\tilde{T}} \mathcal{T}^{-1}\mathcal{U}P_i(\tilde{T}+1)\right) \\ &= 1 - q_i^{-1} P_i(\tilde{T}+1)\left(1 - \frac{r_i q_i^{-1} P_i(\tilde{T})}{1 + r_i q_i^{-1} P_i(\tilde{T})}\right) \\ &= \frac{1 + \frac{r_i}{q_i} P_i(\tilde{T})}{1 + r_i q_i^{-1} + r_i q_i^{-1} P_i(\tilde{T})} = P_i(\tilde{T}+1). \qquad \square \end{aligned}$$

Using the results of Proposition A.1 in the expressions (A.1) and (A.2) gives the desired statement of Theorem 3.1.

Appendix B. Proof of Theorem 4.2.

Consider the error after one step of the algorithm, $\epsilon(t+1) = \mu(t+1) - \overline{\mu}^{(t+1,x(t+1))}$, and recall that $\mu(t+1) = \mu(t) + \alpha(t)E'\hat{\mathbf{x}}^t(\tilde{T})$. Using the analytic expression for $\hat{\mathbf{x}}^t(\tilde{T})$ given in (3.8) gives

$$\mu(t+1) = \mu(t) - \alpha(t)E'Q^{-1}P(\tilde{T})E\left(\mu(t) - \overline{\mu}^{(t,x(t))}\right).$$

Now we can consider $\overline{\mu}^{(t+1,x(t+1))}$. Using Corollary 4.1 one can write

$$\begin{aligned} \overline{\mu}^{(t+1,x(t+1))} &= (E'Q^{-1}P(\tilde{T}-1)E)^{-1}E'(K(\tilde{T}-1)(x(t)+u(t)-\xi)+\xi) \\ &= \overline{\mu}^{(t+1,x(t)+\overline{\mathbf{u}}^t(t))} + (E'Q^{-1}P(\tilde{T}-1)E)^{-1}(E'K(\tilde{T}-1)(u(t)-\overline{\mathbf{u}}^t(t)). \end{aligned}$$

Additionally, we have $(u(t) - \overline{\mathbf{u}}^t(t)) = -R^{-1}K(\tilde{T})E(\mu(t) - \overline{\mu}^{(t,x(t))})$. Using Lemma 2.1, which states that $\overline{\mu}^{(t+1,x(t)+\overline{\mathbf{u}}^t(t))} = \overline{\mu}^{(t,x(t))}$, one can write

$$\begin{aligned} &\overline{\mu}^{(t+1,x(t+1))} \\ &\quad = \overline{\mu}^{(t,x(t))} - (E'Q^{-1}P(\tilde{T}-1)E)^{-1}(E'K(\tilde{T}-1)R^{-1}K(\tilde{T})E(\mu(t) - \overline{\mu}^{(t,x(t))}). \end{aligned}$$

The following equivalence can then be used.

PROPOSITION B.1.

$$K(\tilde{T}-1)R^{-1}K(\tilde{T}) = Q^{-1}(P(\tilde{T}) - P(\tilde{T}-1)).$$

Proof. The proof follows directly from Proposition A.1(ii) and application of the recursions in Table A.1. □

This leads to the following expression:

$$\overline{\mu}^{(t+1,x(t+1))} = \overline{\mu}^{(t,x(t))} + \epsilon(t) - (E'Q^{-1}P(\tilde{T}-1)E)^{-1}E'Q^{-1}P(\tilde{T})E\epsilon(t).$$

By combining the two results for $\mu(t+1)$ and $\overline{\mu}^{(t+1,x(t+1))}$, one obtains

$$\begin{aligned}\epsilon(t+1) &= \mu(t) - \alpha(t)E'Q^{-1}P(\tilde{T})E\epsilon(t)\\ &\quad - \overline{\mu}^{(t,x(t))} - \epsilon(t) + (E'Q^{-1}P(\tilde{T}-1)E)^{-1}(E'Q^{-1}P(\tilde{T})E\epsilon(t)\\ &= -\alpha(t)E'Q^{-1}P(\tilde{T})E\epsilon(t) + (E'Q^{-1}P(\tilde{T}-1)E)^{-1}E'Q^{-1}P(\tilde{T})E\epsilon(t)\\ &= \Big((E'Q^{-1}P(\tilde{T}-1)E)^{-1} - \alpha(t)I\Big)\,E'Q^{-1}P(\tilde{T})E\epsilon(t).\end{aligned}$$

Appendix C. Proof of Corollary 4.5.

We consider the case where each agent has identical control and state weights (e.g., $Q = qI_n$ and $R = rI_n$). With this assumption, we also note that the control gains for each agent are identical; that is, $P_i(\tilde{T}) = P_j(\tilde{T})$ for all i, j. To simplify the notation, we write $P(\tilde{T}) = \tilde{P}(\tilde{T})I_n$, where $\tilde{P}(\tilde{T})$ is a scalar. Performing a coordinate transformation $U\tilde{\epsilon}(t) = \epsilon(t)$ on the system (4.5) gives[5]

$$\left(\frac{q}{\tilde{P}(\tilde{T}-1)}\Lambda^{-1} - \alpha(t)I\right)^{-1}\tilde{\epsilon}(t+1) = \frac{\tilde{P}(\tilde{T})}{q}\Lambda\tilde{\epsilon}(t). \tag{C.1}$$

This diagonalized system can be used to determine a range for $\alpha(t)$ that guarantees the eigenvalues are inside the unit disc for all times $t = 0, \dots, T-1$,

$$\frac{q}{\lambda_i(E'E)\tilde{P}(\tilde{T})}\left(\frac{\tilde{P}(\tilde{T})}{\tilde{P}(\tilde{T}-1)} - 1\right) \leq \frac{q}{\lambda_j(E'E)\tilde{P}(\tilde{T})}\left(\frac{\tilde{P}(\tilde{T})}{\tilde{P}(\tilde{T}-1)} + 1\right),$$

for i, $j = 1, 2, \dots, n-1$. The bound in (4.7) follows directly from above. Observing that $\tilde{P}(\tilde{T}) > \tilde{P}(\tilde{T}-1)$ and that the inequality should hold for all time and all combinations of graph eigenvalues leads us to the following inequality:

$$\frac{\lambda_{\max}(E'E)}{\lambda_{\min}(E'E)} \leq \frac{\tilde{P}(\tilde{T}) + \tilde{P}(\tilde{T}-1)}{\tilde{P}(\tilde{T}) - \tilde{P}(\tilde{T}-1)}.$$

Using the recursion relationship for $\tilde{P}(\tilde{T})$, we arrive at the statement of the corollary.

Acknowledgments. The authors would like to thank the German Research Foundation (DFG) for financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

REFERENCES

[1] R. Bellman, *Dynamic Programming*, Dover Publications, Mineola, NY, 2003.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*, Prentice-Hall, Old Tappan, NJ, 1989.

[5] Such a transformation exists since $E'E$ is the *edge Laplacian* for the graph $\mathcal{G}$ [19], which is symmetric and positive definite.

[3] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[4] M. Bürger, G. S. Schmidt, and F. Allgöwer, *Preference based group agreement in cooperative control*, in Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems, 2010, pp. 149–154.

[5] C. D. Godsil and G. Royle, *Algebraic Graph Theory*, Springer, New York, 2001.

[6] B. Johansson, M. Rabi, and M. Johansson, *A randomized incremental subgradient method for distributed optimization in networked systems*, SIAM J. Optim., 20 (2009), pp. 1157–1170.

[7] B. Johansson, A. Speranzon, M. Johansson, and K. H. Johansson, *On decentralized negotiation of optimal consensus*, Automatica, 44 (2008), pp. 1175–1179.

[8] D. E. Kirk, *Optimal Control Theory: An Introduction*, Dover Publications, Mineola, NY, 2004.

[9] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, Princeton, NJ, 2010.

[10] A. Nedic and A. Ozdaglar, *Distributed subgradient methods for multi-agent optimization*, IEEE Trans. Automat. Control, 54 (2009), pp. 48–61.

[11] R. Olfati-Saber, J. A. Fax, and R. M. Murray, *Consensus and cooperation in networked multi-agent systems*, Proc. IEEE, 95 (2007), pp. 215–233.

[12] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, version 20081110 (2008), http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3274.

[13] A. Rantzer, *Using Game Theory for Distributed Control Engineering*, Technical report ISRN LUTFD2/TFRT–7620–SE, Department of Automatic Control, Lund University, Sweden, 2008.

[14] W. Ren, R. W. Beard, and E. M. Atkins, *A survey of consensus problems in multi-agent coordination*, in Proceedings of the American Control Conference, 2005, pp. 1859–1864.

[15] A. Ruszczynski, *Nonlinear Optimization*, Princeton University Press, Princeton, NJ, 2006.

[16] G. Strang and K. Borre, *Linear Algebra, Geodesy, and GPS*, Wellesley-Cambridge Press, Wellesley, MA, 1997.

[17] J. N. Tsitsiklis, *Problems in Decentralized Decision Making and Computation*, Ph.D. thesis, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, 1984.

[18] R. A. Usmani, *Inversion of a Tridiagonal Jacobi Matrix*, Linear Algebra Appl., 212 (1994), pp. 413–414.

[19] D. Zelazo and M. Mesbahi, *Edge agreement: Graph-theoretic performance bounds and passivity analysis*, IEEE Trans. Automat. Control, 56 (2011), pp. 554–555.

[20] M. Zhu and S. Martinez, *On distributed convex optimization under inequality and equality constraintss*, IEEE Trans. Automat. Control, 57 (2012), pp. 151–164.