2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Decentralized Design for LQ Consensus in Multi-Agent Systems

Stefan van Dijk, Paula Chanfreut, W.P.M.H. (Maurice) Heemels

Abstract— **Consensus control of multi-agent systems (MAS) with integrator dynamics is a canonical and well-studied problem in the literature. In contrast, the *optimal* distributed linear quadratic (LQ) consensus problem minimizing an appropriate quadratic cost is less studied. For this problem, most of the available design methods require global information on the interconnection graph and/or global (initial) state information. In this paper, we propose a suboptimal solution to the LQ-based distributed consensus control problem by approximating the quadratic cost function in a way that allows for decentralized design of the control gains. The resulting control protocol only uses information from neighboring agents for both control design and implementation. It has the additional benefit that the information can be exchanged periodically, reducing the communication requirements of the agent. Despite the suboptimality, asymptotic consensus is guaranteed for our control law, as we will formally prove. We illustrate by numerical simulations on a 6-agent system the effectiveness of our design and compare it to other approaches.**

I. Introduction

Consensus of MAS has been studied extensively in the literature and gained increasing popularity in recent years [1] with applications in e.g., opinion dynamics [2], complex electrical networks [3], and unmanned aerial and ground vehicles [4]. A typical approach to achieve consensus for such systems is to implement a distributed diffusive control law, scaling the differences between agents' states with a certain control gain [5].

Optimal consensus, the problem of getting agents to agree while minimizing an appropriate cost function, has received significantly less attention, but some researchers have considered this interesting problem, see, for example, [6]–[9] and the references therein. Unfortunately, many of the developed control designs in these papers make use of *global* information of the network, for example, by requiring knowledge of the graph structure and/or all the initial state of the agents, in order to solve a Lyapunov equation of dimension equal to the number of agents, or by using the smallest eigenvalue of the Laplacian matrix. In other words, in these cases, a central entity must be aware of all agents interconnections, must have access to their initial state values, and must have the computational capacity to solve a (high-dimensional) matrix equation. Due to these requirements, the computation of the theoretical global solution to the optimal consensus problem is not tractable in many real-world applications and can not be implemented by each agent individually. Examples of such systems are router networks, where the communication graph changes at a fast pace, or robot swarms, which often result in prohibitively large matrix equations.

The main focus of this paper is on alleviating these limitations for computing optimal LQ consensus gains. In other words, the aim is to create a methodology for designing a control law that (approximately) minimizes the quadratic cost, but the synthesis can be done based on local information only. Specifically, instead of minimizing the original quadratic costs exactly, we minimize a close approximation of this cost with the benefit that each agent can determine its own suboptimal gain based on local information (the number of neighbors). This leads to a control protocol that features both distributed control inputs and decentralized design. The approximate cost that we minimize is obtained by sampling the last transmitted neighboring states, which has the additional benefit of reducing the communication overhead and better represents the on-line (often digital) implementation in applications. Interestingly, for this new design, formal consensus guarantees are provided for arbitrary sampling periods.

A related method, inspiring ours, was explored in [10], where the authors design a distributed consensus control protocol, in which each individual agent controls its state towards the average value of its neighboring states, which is sampled and held constant between sampling periods, by solving an optimal LQ-tracking problem. We extend their results by removing the averaging step. In particular, we minimize a cost that is not related to the average of the neighboring states, but uses all the information available to each agent. Intuitively, this results in a performance improvement for our design, as the approximate cost is closer to the original quadratic consensus cost. This intuition is confirmed by a numerical example, where for three different network structures, the original cost is significantly lower for the design proposed in this paper. This new approach also requires new technical developments not available in [10]. Other distributed design approaches can be found in [11], where the authors design a distributed adaptive consensus protocol that achieves leader-follower consensus for a directed communication graph.

The remainder of this paper is organized as follows. Section II introduces the distributed LQ consensus problem. Section III presents the proposed sampling-based distributed control protocol featuring the decentralized control design. In Section IV, we calculate the control gain for single integrator systems by solving a Ricatti equation analytically. In Section V, we prove that resulting input achieves consensus. In Section VI our theoretical results are validated numerically.

S. van Dijk, P. Chanfreut, W.P.M.H. Heemels are with the Control Systems Technology section of the Department of Mechanical Engineering, TU/e, 5612 AE Eindhoven, The Netherlands. E-mails: s.v.dijk@tue.nl, p.chanfreut.palacio@tue.nl, M.Heemels@tue.nl.

A. Notation

For matrices $M \in \mathbb{R}^{n\times n}$, $\sigma(M)$ denotes its spectrum, and $X_+^d(M) = \operatorname{span}\{v \in \mathbb{R}^n : (\lambda, v)$ an eigenpair with $|\lambda| \geq 1\}$ is its discrete unstable eigenspace, i.e, $Mv = \lambda v,\ v \neq 0,\ \lambda \in \sigma(M)$. We write $P \succ 0$ ($P \succeq 0$), if P is symmetric positive (semi-)definite. The vector of all ones in $\mathbb{R}^n$ is given by $\mathbb{1}_n$, and the matrix of all ones in $\mathbb{R}^{n\times n}$ by $\mathbf{1}_n$. We omit the dependence on n whenever this is obvious. In addition, $\mathcal{G} = (V, E)$ denotes a graph with vertex set $V = \{1, \dots, N\}$ and edge set $E \subset V \times V$, where $(a, b) \in E$, if there is an edge from a to b. We denote the adjacency matrix of a graph by A and its degree matrix by D. Finally, $N_i = \{j : (i, j) \in E\}$ denotes the neighbor set of vertex $i \in V$.

II. Distributed LQ Consensus Problem

In this section, we introduce the distributed LQ consensus problem for multi-agent systems. Consider a MAS on the undirected connected graph $\mathcal{G} = (V, E)$ with Laplacian L such that each agent has scalar single integrator dynamics

$$\dot{x}_i(t) = u_i(t), \quad x_i(0) = x_i^0,\ \forall i \in V, \tag{1}$$

where $x_i(t) \in \mathbb{R}$ is the state of agent $i \in V$ at time $t \in \mathbb{R}_{\geq 0}$, and the initial states x_i^0 are given for all $i \in V$. Also, consider a distributed diffusive coupling input

$$u_i(t) = -g_i \sum_{j\in N_i} (x_i(t) - x_j(t)),\ \forall i \in V, \tag{2}$$

with control gains $g_i \in \mathbb{R}$. Combining (1) and (2) gives the overall network dynamics $\dot{x}(t) = -gLx(t)$ with $x(t) = [x_1(t), \dots, x_N(t)]^\top$ the state vector, $u(t) = [u_1(t), \dots, u_N(t)]^\top$ the input vector, and the diagonal control gain matrix $g = \operatorname{diag}(g_1, \dots, g_N)$. A MAS is said to achieve consensus (synchronize), if for any initial state $x_0 = x(0) \in \mathbb{R}^N$, $\lim_{t\to\infty} x(t) \in \operatorname{span} \mathbb{1}$. It is known that (1) synchronizes if and only if $g \succ 0$ [6].

In order to obtain an *optimal* control gain, consider the following quadratic cost function,

$$\begin{aligned} J(x^0, u) &= \sum_{i=1}^{N} \underbrace{\left(\int_0^\infty \sum_{j\in N_i} q(x_i(t) - x_j(t))^2 + ru_i^2(t)\mathrm{d}t\right)}_{J_i\left(x_i^0, u_i\right)} \\ &= \int_0^\infty x^\top(t)2qLx(t) + ru^\top(t)u(t)\mathrm{d}t, \end{aligned} \tag{3}$$

with state weight $q > 0$ and input weight $r > 0$. In the homogeneous gain case[1], i.e $g_i = g_j$ for all $i, j \in V$, to find the optimal gain, consider the Gramians $X_0 = \int_0^\infty e^{-\tau L} 2qLe^{-\tau L}\mathrm{d}\tau$ and $Y_0 = \int_0^\infty e^{-\tau L} re^{-\tau L}\mathrm{d}\tau$. Combining (2) and (3) and setting $\tau = gt$ yields $J(x_0, -gLx(t)) = x_0^\top \left(\frac{1}{g}X_0 + gY_0\right) x_0$. Differentiating with respect to g, we see that the optimal gain, if it exists[2], satisfies

$$g_{\text{opt}} = \left(\frac{x_0^\top X_0 x_0}{x_0^\top Y_0 x_0}\right)^{\frac{1}{2}}. \tag{4}$$

[1]The heterogeneous gain case is similar, replacing the derivative with the multivariate analogue.

[2]It does not exist in e.g. the case where $x_0 \in \ker Y_0$.

From (4) we see that the computation of the optimal control law requires both knowledge of the initial state x_0 and of the Laplacian matrix L. Furthermore, evaluating the Gramians can become prohibitively difficult, or even impossible. This holds even if the Gramians are viewed as solutions to the Lyapunov equations

$$LX_0 + X_0L = 2qL, \quad LY_0 + Y_0L = rI, \tag{5}$$

as obtaining these solutions is nontrivial for large networks. While the control law acts locally, it requires a central entity with global network and state information to compute the optimal control gain. The challenge we focus on in this paper, is computing a control gain matrix based on locally available information, that achieves consensus and minimizes (3). While this is impossible, we can approximate the minimizer of (3) by using a discounted LQ formulation, as we explain below.

III. Local Sampling-based Controller Design

In this section, we introduce a local design method, where each agent computes a control input using information available from a decentralized viewpoint. We first identify what information is available to each agent at any particular time-instant. Then, we use that information to set-up another optimization problem, whose minimization approximates the minimization of (3). While this design does not guarantee global optimality, it guarantees consensus, as we will prove.

A. Introducing decentralized costs

We make the following assumptions about the system:

Assumption 1. Each agent has access to its own state $x_i(t)$, and can manipulate its own input $u_i(t)$ at all $t \in \mathbb{R}_{\geq 0}$.

Assumption 2. At time instants kT, $k \in \mathbb{N}$, $T > 0$ the *sampling period*, all agents synchronously transmit their state to all neighbors and receive a sample in return.

Assumption 1 is natural in most applications. Assumption 2 is instrumental in our approach to realize decentralized design of the distributed consensus law, as we will show below. Indeed, if we would stick to minimizing (3), directly, in a decentralized setting agent $i \in V$ would optimize its performance if it minimizes cost $J_i\left(x_i^0, u_i\right)$ in (3). However, minimizing $J_i(x_i^0, u_i)$ requires knowledge of the future neighboring trajectories, influenced by their neighbors again, which are unknown. Inspired by [10], by sampling the neighbors' states, and assuming that they stay constant in the future, $J_i(x_i^0, u_i)$ becomes a function that can be locally optimized and thus leading to a decentralized design. This sampling approach has the added benefit of being closer to real-life digital implementations of the controller and its communications.

Without loss of generality, agent i is instructed to simultaneously track all received neighbor states, for time $t \in [kT, (k + 1)T)$, with control gain g_i, using the sampled, distributed control input

$$u_i(t) = -g_i \sum_{j\in N_i} (x_i(t) - x_j(kT)), \quad i \in V. \tag{6}$$

They then determine an individual g_i, by minimizing the *discounted local sampled tracking cost*:

$$J_{i,k}^{\alpha}(u_i) = \int_{kT}^{\infty} e^{-2\alpha t}\Big[\sum_{j\in N_i} q(x_i(t)-x_j(kT))^2 + r u_i^2(t)\Big]\mathrm{d}t, \tag{7}$$

where $\alpha > 0$ is the *discount factor*. The introduction of the discounted formulation is required as we shall see in Remark 2. Once the gains that minimize (7) are calculated for the interval $[kT, \infty)$, the corresponding control input is applied in $[kT,(k+1)T)$, see (6). At the next sampling time, the agents repeat this procedure, update their input, and generate a control law on $[0,\infty)$.

Remark 1: We consider the infinite time horizon, rather than minimizing over $[kt,(k+1)T)$, where the control is applied, since we would otherwise require terminal boundary conditions for the resulting Differential Ricatti Equation. Furthermore, a differential equation solver should be present in each agent, which could be difficult to implement.

B. The Discounted LQ Tracking Problem

In this section we recall the general solution to the discounted LQ tracking problem, detailed in, e.g., [10], [12], and apply it to (7). Consider the system $\dot z(t) = Hz(t) + Bu(t)$, with $z(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^p$ the state and input vectors at time $t \in \mathbb{R}_{\geq 0}$, $z(0) = z_0$ the initial state, and H and B matrices of appropriate dimensions. Suppose now that a constant reference signal $\rho \in \mathbb{R}^n$ and cost matrices $Q \succeq 0, R \succ 0$ are given. Then, the discounted LQ tracking problem, is the problem of finding a minimizer for

$$J(u) = \int_0^{\infty} e^{-2\alpha t}[(z(t)-\rho)^\top Q(z(t)-\rho) + u^\top(t)Ru(t)]\mathrm{d}t. \tag{8}$$

It is known that the optimal input is given by:

$$u(t) = -R^{-1}B^\top X_1 z(t) - R^{-1}B^\top X_2\rho, \tag{9}$$

where $X_1, X_2 \in \mathbb{R}^{n\times n}$ are block components of the smallest positive definite solution X, of the ARE

$$H_e^\top X + XH_e - XB_eR^{-1}B_e^\top X + Q_e = 0, \tag{10}$$

with matrices

$$\begin{aligned} H_e &= \begin{bmatrix} H-\alpha I & 0 \\ 0 & -\alpha I\end{bmatrix}, & B_e &= \begin{bmatrix} B \\ 0\end{bmatrix}, \\ Q_e &= \begin{bmatrix} Q & -Q \\ -Q & Q\end{bmatrix}, & X &= \begin{bmatrix} X_1 & X_2 \\ X_2^\top & X_3\end{bmatrix}. \end{aligned} \tag{11}$$

Lemma 1: A unique solution X to (10) exists, if (H_e, B_e) is stabilizable and (Q_e, H_e) is detectable [13, Exercise 10.4].

Suppose now that agent i has d_i neighbors. Given the dynamics in (6) define $z_i = \mathbb{1}_{d_i} x_i$ and $\rho_i(kT) = \begin{bmatrix} x_{j_1}(kT) & \dots & x_{j_{d_i}}(kT)\end{bmatrix}^\top$ for $j_\bullet \in N_i$. A unique minimizer to (7) is then obtained by applying these results, with

$$H = 0, \quad B = \mathbb{1}, \quad Q = qI, \quad \text{and } R = rI. \tag{12}$$

The ARE in (10) then takes the form

$$-2\alpha r\begin{bmatrix} X_1 & X_2 \\ X_2^\top & X_3\end{bmatrix} - \begin{bmatrix} X_1\mathbf{1}X_1 & X_1\mathbf{1}X_2 \\ X_2^\top\mathbf{1}X_1 & X_2^\top\mathbf{1}X_1\end{bmatrix} = rq\begin{bmatrix} -I & I \\ I & -I\end{bmatrix}, \tag{13}$$

which can be solved by first finding a solution to

$$2\alpha r X_1 + X_1\mathbf{1}X_1 = rqI, \tag{14}$$

and setting $-X_2 = X_3 = X_1 \succ 0$. A Schur decomposition [14, Theorem 1.12] shows that this implies $X \succeq 0$, and thus X is the unique solution to (13) by Lemma 1, which yields the minimizer to (7) through (9).

Remark 2: For (H_e, B_e) to satisfy the conditions of Remark 1, with matrices as in (12), it is necessary that the discount factor is strictly positive. The control law (9) guarantees exponential asymptotic decay of the tracking error $e^{-\alpha t}(z_i(t) - \rho_i(kT)) \to 0$ for $t \to \infty$. Since α can be taken arbitrarily small, we can control the decay rate of the error, and make (7) approximate the summands in (3).

IV. Optimizing Locally

In this section, we provide an analytic expression for the unique positive semi-definite solution of (14), and use it to obtain an expression for the gains g_i. Since the solution of an ARE is a linear feedback of the state, the resulting solution takes the form of (6).

Lemma 2: Equation (14) has a unique positive definite solution $X_1 \in \mathbb{R}^{d_i\times d_i}$ given by:

$$X_1 = (a-b)I + b\mathbf{1} \text{ with} \tag{15}$$

$$b = \left(-\alpha r - \frac{qd_i}{2} + \sqrt{\alpha^2 r^2 + d_i q r}\right)\frac{1}{d_i^2}, \quad a = \frac{q}{2\alpha} + b. \tag{16}$$

Proof: We can verify that this choice of a and b satisfies (14). In addition, we show how these values were obtained, for which we use the fact that $\sigma(hI + M) = h + \sigma(M)$, for any $h \in \mathbb{R}$ and any matrix $M \in \mathbb{R}^{n\times n}$.

Note that $\sigma(X_1) = (a-b) + b\sigma(\mathbf{1})$, which gives eigenvalues

$$\lambda_1 = a + (d_i - 1)b, \quad \lambda_2 = \ldots = \lambda_{d_i} = a - b. \tag{17}$$

Next, notice that $X_1\mathbb{1} = \lambda_1\mathbb{1}$ and $X_1\mathbf{1}X_1 = c\mathbf{1}$, where

$$c = (a-b)^2 + 2d_i b(a-b) + d_i^2 b^2. \tag{18}$$

Substituting (18) and (15) in (14), we obtain $(2\alpha r(a-b) - rq)I + (2rb + c)\mathbf{1} = 0$, which holds if

$$a - b = \frac{q}{2\alpha}, \quad 2rb + c = 0. \tag{19}$$

Combining (14), (17) and (19) and taking determinants:

$$\begin{aligned} &\det(2\alpha r I + X_1\mathbf{1})\det(X_1) = \det(qrI) \\ &\implies (2\alpha r)^{d_i-1}(2\alpha r + d_i\lambda_1)\lambda_1\left(\frac{q}{2\alpha}\right)^{d_i-1} = (qr)^n \\ &\implies d_i\lambda_1^2 + 2\alpha r\lambda_1 - qr = 0. \end{aligned} \tag{20}$$

Solving this quadratic equation yields the positive solution $\lambda_1 = (-\alpha r + \sqrt{\alpha^2 r^2 + d_i q r})/d_i$. Combining the latter with (19) results in a system of equations in a, b whose solution is (16). Notice that $\lambda_1 > 0$ by construction and $a - b = q/2 > 0$ and hence $X_1 \succ 0$ as required. ■

From Lemma 2, the gains g_i in (6) as well as the overall control law can be computed for the entire network.

Theorem 1: For a MAS on a graph $\mathcal{G} = (V, E)$ with agent dynamics $\dot{x}_i(t) = u_i(t)$, $t \in [kT, (k+1)T)$, $i \in V$, $k \in \mathbb{N}$, the overall control input minimizing (7) is:

$$u(t) = GDx(t) - GAx(kT), \tag{21}$$

where $G = \operatorname{diag}(-g_1, \ldots, -g_N)$,

$$g_i = \frac{\sqrt{\alpha^2 r^2 + d_i q r} - \alpha r}{r d_i} > 0, \tag{22}$$

and d_i is the number of neighbors of agent i.

Proof: Applying the results of Lemma 2, and considering (9), $z_i = \mathbb{1} x_i$, and the matrices in (11) on $t \in [kt, (k+1)T)$ yields

$$\begin{aligned} u_i(t) &= -\frac{1}{r}(\lambda_1 d_i x_i - \lambda_1 \mathbb{1}^\top \rho_i(kT)) \\ &= -g_i \sum_{j \in N_i} (x_i(t) - x_j(kT)), \end{aligned} \tag{23}$$

with g_i as in (22). Finally, (21) follows from (23). ■

Remark 3: Notice that the gains $g_i > 0$ are independent of k and T, and hence can be calculated once, provided the graph topology remains constant. If it changes, the agents need only to know the changes in the number of neighbors to adapt their gain. Clearly, decentralized local design of the gains is possible without need for global information.

V. Consensus Guarantees

In this section, we provide consensus guarantees for the control protocol in Theorem 1. In this regard, a particular difference equation for the system dynamics at the sample periods is considered. By analyzing the discrete-time system, it is proven that the system synchronizes.

We observe that for each $k \in \mathbb{Z}_+$ the global dynamics of the networked system in the interval $[kT, (k+1)T)$ are given by $\dot{x}(t) = GDx(t) - GAx(kT)$. The state trajectory can then be calculated as

$$x(t) = e^{GDt}\left[e^{-GDkT} - \int_{kT}^{t} e^{-GD\tau} GA \mathrm{d}\tau\right] x(kT), \tag{24}$$

where the integral term can be computed as

$$\int_{kT}^{t} e^{-GD\tau} GA \mathrm{d}\tau = \left(e^{-GDkT} - e^{-GDt}\right) D^{-1} A. \tag{25}$$

Combining (24) and (25), and evaluating at $t = (k+1)T$, we obtain the discrete-time system $x((k+1)T) = \Gamma x(kT)$ with system matrix

$$\Gamma = e^{GDT} + \left(I - e^{GDT}\right) D^{-1} A. \tag{26}$$

Therefore, $x(kT) = \Gamma^k x(0)$ and hence $\lim_{k\to\infty} x(kT) \in X_+^d(\Gamma)$, the unstable eigenspace of Γ. We shall analyze the spectrum of Γ to prove that the control law (21) gives consensus. First, we shall need the following lemma.

Lemma 3: Given a control law u as in (9) the network achieves consensus if and only if $X_+^d(\Gamma) = \operatorname{span} \mathbb{1}$.

Proof: By definition, the network achieves consensus for any arbitrary $x(0)$ if and only if for all i and all $j \in N_i$ we have that $\lim_{t\to\infty} x_i(t) - x_j(t) = 0$. Then, observe that:

$$\begin{aligned} \lim_{t\to\infty} x_i(t) - x_j(t) = 0 &\iff \lim_{t\to\infty} x(t) \in \operatorname{span} \mathbb{1}, \\ &\iff \lim_{k\to\infty} x(kT) \in \operatorname{span} \mathbb{1}, \\ &\iff X_+^d(\Gamma) = \operatorname{span} \mathbb{1}, \end{aligned} \tag{27}$$

and thus $\lim_{k\to\infty} Lx(kT) = 0$. We only need to argue the second equivalence, which establishes that to prove consensus, it is sufficient to check the system state at the sample times. The forward direction is obvious, the backwards direction follows from $\lim_{k\to\infty} Lx(kT) = 0 \implies \lim_{k\to\infty} u_i(kT) = 0$. ■

Proposition 1: The matrix Γ satisfies $\sigma(\Gamma) \subset (-1, 1]$ and furthermore $(1, \mathbb{1})$ is a simple eigenpair. Hence, its unstable eigenspace is given by $X_+^d(\Gamma) = \operatorname{span} \mathbb{1}$.

Proof: By substituting $D = L - A$ in (26) we get

$$\Gamma = I + \left(e^{GDT} - I\right) D^{-1} L. \tag{28}$$

Next, we define $\Gamma = I + \Delta L$ with $\Delta = (e^{GDT} - I)D^{-1}$ and observe this is a diagonal matrix. Furthermore, ΔL satisfies

$$(-\Delta)^{-\frac{1}{2}} \Delta L (-\Delta)^{\frac{1}{2}} = (-\Delta)^{\frac{1}{2}} L (-\Delta)^{\frac{1}{2}}, \tag{29}$$

and thus ΔL is similar to a symmetric matrix whose spectrum is real. Adding I to a matrix shifts its eigenvalues by 1 and hence $\sigma(\Gamma) \subset \mathbb{R}$.

To localize the eigenvalues, we use Gershgorin's Circle Theorem [15, Theorem 6.1.1]. Since $D^{-1}A$ has 0 in its diagonals, from (26) we see that the Gershgorin centra of Γ are given by $c_i = e^{-g_i d_i T}$, and since $g_i > 0$ we have $c_i \in (0, 1]$. From (28), the radii are $\delta_i = 1 - e^{-g_i d_i T} \in [0, 1)$, and since the eigenvalues are real, the Gershgorin Circles are given by $C_i = \{\zeta \in \mathbb{R} : |\zeta - c_i| \leq 1 - c_i\}$. Hence, $-1 + 2c_i \leq \zeta \leq 1$, and since $c_i > 0$, we have $\zeta \in (-1, 1]$ and thus $\sigma(\Gamma) \subset (-1, 1]$.

We obtain that $(1, \mathbb{1})$ is an eigenpair by considering (28). It remains to show that 1 is a simple-eigenvalue. Consider $(1, v)$ an eigenpair, then, from (26):

$$v = \Gamma v \implies \Delta L v = 0 \implies v \in \operatorname{span} \mathbb{1}, \tag{30}$$

since Δ is invertible and $\mathcal{G}$ is connected. ■

Theorem 2: The control law (6), with gains g_i computed as in (22) results in consensus of the agents.

Proof: The result is a direct consequence of Lemma 3 and Proposition 1. ■

Remark 4: Theorem 2 gives asymptotic consensus guarantees independent of T. However, if T is large, then the smallest eigenvalue of Γ approaches -1, which greatly reduces the speed of convergence.

Remark 5: In the language of ergodic consensus problems [16], the results above show that Γ is a stochastic, irreducible, aperiodic matrix and, hence, consensus of the discrete-time system is guaranteed. In the special case of regular graphs, Γ is in fact doubly stochastic and thus we get consensus to the mean.

VI. Simulations

This section presents simulations of the control protocol designed in this paper. The trajectories are compared to the results obtained using (i) the averaged controller in [10], and (ii) the centralized control law that optimizes the homogeneous case of global index (3) (i.e., $g = g_{\text{opt}} I$). While the decentralized design uses heterogeneous gains, we shall see that it does not outperform the homogeneous global minimum in any scenario that we test, presumably due to the latter's use of global information. In what follows, these approaches will be referred to as *local*, *averaged* and *global*, respectively, and plotted with dashed, dotted and straight lines. Note that the gains in the averaged controller are *independent of graph structure*, and as such constant for all graphs, unlike the local or global methods. Finally, we consider three different graphs with 6 agents, i.e.,

- A circular graph $\mathcal{G}_{\text{circ}}$ such that $N_i = \{i-1, i+1\}$, for all $i \in \{2, \ldots, 5\}$, $N_1 = \{6, 2\}$, and $N_6 = \{1, 5\}$.
- A path graph $\mathcal{G}_{\text{path}}$ such that $N_i = \{i-1, i+1\}$, for all $i \in \{2, \ldots, 5\}$, $N_1 = \{2\}$, and $N_6 = \{5\}$.
- A complete graph $\mathcal{G}_{\text{comp}}$ such that $N_i = \{1, \ldots, 6\} \setminus \{i\}$, for all $i \in \{1, \ldots, 6\}$.

The initial state is always $x^0 = [1, 2, -1, -2, 1, 3]^\top$, and other parameters are $\alpha = 0.01$, $q = 2$ and $r = 1$. The cumulative cost obtained in each of the simulations with two different sampling times T are given in Table I, where it can be seen that the local method outperforms the averaged one.

The cumulative cost is computed by simulating each method for $t \in [0, 8]$, and sampling the states and inputs every $\Delta t = 0.001$. Particularly, defining $t_n = n\Delta t$ with $n \in \mathbb{Z}_{\geq 0}$ we get

$$\Delta t \sum_{n=1}^{10/\Delta t} \left(2qx^\top(t_n) L x(t_n) + u^\top(t_n) r I u(t_n)\right). \qquad (31)$$

For $\mathcal{G}_{\text{circ}}$, the gains are given by $g_{\text{opt}} \approx 1.7061$, $g_{\text{loc}} = 0.9950$, and $g_{\text{av}} = 1.402$. This g_{av} remains constant for all other graphs. The resulting state trajectories are plotted for $T = 0.1$ in Figure 1, and for $T = 1$ in Figure 2. We observe that while the local method synchronizes quicker than the averaged method, it is still slower than the optimal one. The latter illustrates the compromise reached between optimizing the overall performance and decentralizing the computation of the gains. Note also that the smoothness of the trajectories is highly dependent on the choice of $T = 0.1$ or $T = 1$. Nonetheless, the proposed controller guarantees consensus in both cases, as expected by the developed theory.

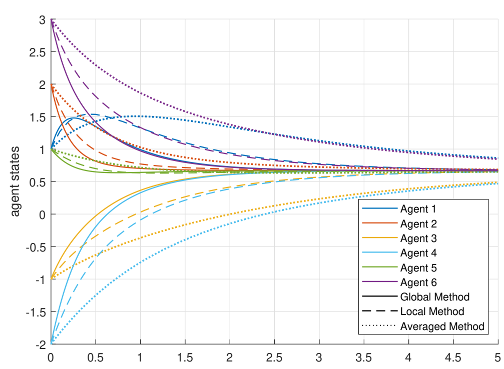

Fig. 1. State evolution for all agents on the cycle graph, with $T = 0.1$.

.

For $\mathcal{G}_{\text{path}}$, the resulting state trajectories using $T = 0.1$ are given in Figure 3, with gains $g_{\text{loc},i} = 0.9950$ for $i = 2, \ldots, 5$, $g_{\text{loc},i} = 1.4042$ for $i = 1, 6$ and $g_{\text{opt}} = 1.7472$. While all three methods converge, the local and averaged approaches reach consensus to a different state value than the global one (see Remark 5), since the path graph is not regular.

Consider now $\mathcal{G}_{\text{comp}}$, for which the gains are $g_{\text{loc}} = 0.6305$, and $g_{\text{opt}} \approx 0.701$, and the resulting state trajectories are plotted in Figure 4. We claimed in Section I that the local approach allows each agent to consider more of its available information than the averaged method. As such, it is expected to perform significantly better for graphs with bigger information flow, and that is why we consider the complete graph. The performance benefits can be seen in both the rate of convergence and the cumulative cost in Table I. Notice that for complete graphs the Lyapunov equations (5) do not have a unique solution, and, hence, calculating the optimal gain is difficult. In this regard, we calculated the optimal gain g_{opt} by testing values in the interval $[0, 2]$, which is not feasible for larger networks.

Finally, consider an additional performance measure for the three methods. We define the *consensus error* of a state trajectory at time t as $e(t) = \max_{i,j \in [1,\ldots,6]} \{x_i(t) - x_j(t)\}$, i.e., the largest deviation between any two agent states at time t. Since the goal is consensus, this value is a measure of the *speed of convergence* of each method. We simulate the three of them for $t \in [0, 8]$ on all three graphs, and evaluate $e(t)$ in the middle of the sampling intervals. The

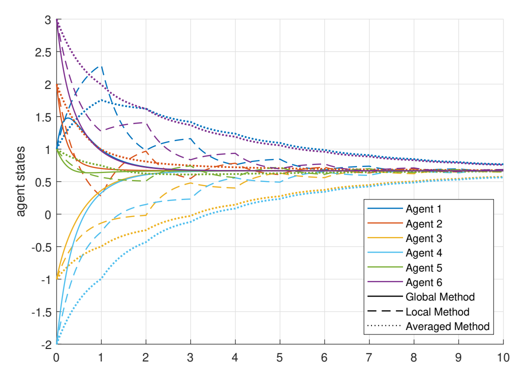

Fig. 2. State evolution for all agents on the cycle graph, with $T = 1$.

TABLE I
Cumulative performance costs in different simulations.

Graph	Sampling	Local	Global	Averaged
Cycle	$T = 0.1$	49.7	44.1	82.9
Cycle	$T = 1$	73.2	44.1	106.9
Path	$T = 0.1$	44.9	40.8	73.8
Path	$T = 1$	55.5	40.8	95.0
Complete	$T = 0.1$	87.1	85.9	160.3
Complete	$T = 1$	92.2	85.9	160.3

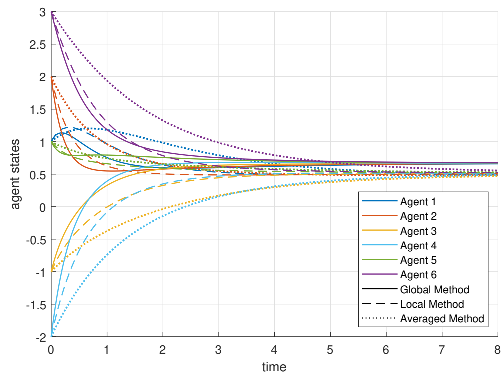

Fig. 3. State evolution for all agents on the path graph, $T = 0.1$.

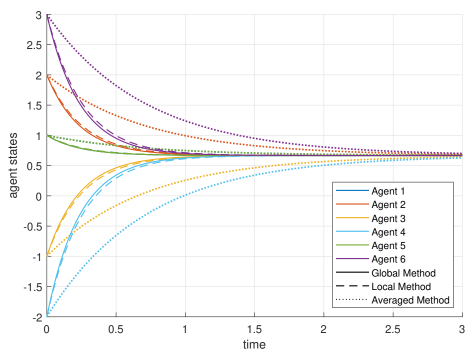

Fig. 4. State evolution for all agents on the complete graph, $T = 0.1$.

results are shown in Figure 5 with logarithmic scale. As can be seen, the local method outperforms the average method on all tested graphs, and approaches the global performance for the complete graph.

VII. Conclusion & Further Work

This paper presented a decentralized suboptimal method for the design of distributed consensus laws for integrator MAS. The approach, inspired by [10], is based on transitioning to a sampled, discounted approximation of the original quadratic consensus cost, for which the computation of the control gains can be performed by the individual agents based on local information only (essentially the number of neighbors). Due to the proven consensus guarantees, the sampled-data implementation and the local design, this method could be attractive for certain real-world (sub)optimal consensus problems.

The presented single integrator theory serves as a proof of concept, and we believe, due to preliminary simulation results, that the basic principles can be extended to other kinds of systems with more general dynamics, (dynamic) directed graphs, and heterogeneous cost weights. We plan to investigate these extensions in further works. Another line of future work is abandoning the period sampling requirement, calling for clock synchronization throughout the network, and allowing also asynchronous updating and communication of state information by the agents.

VIII. Aknowledgments

We would like to thank H. L. Trentelman who proposed this topic as a Bachelors' thesis to Stefanos. Without him and his insights, this paper would not have been possible. Furthermore, we thank J. Jiao for providing his simulation code, so that the methods could adequately be compared.

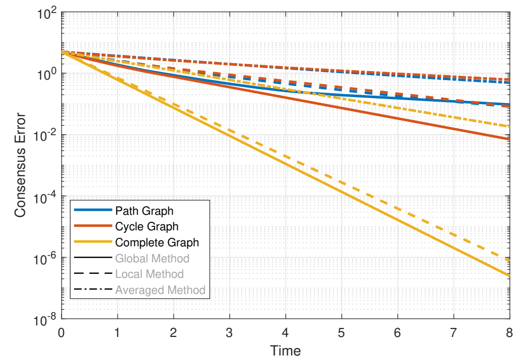

Fig. 5. Consensus error between methods, $T = 0.1$.

References

[1] X.-M. Zhang, Q.-L. Han, and X. Yu, "Survey on recent advances in networked control systems," *IEEE Transactions on industrial informatics*, vol. 12, no. 5, pp. 1740–1752, 2015.

[2] A. V. Proskurnikov, A. S. Matveev, and M. Cao, "Opinion dynamics in social networks with hostile camps: Consensus vs. polarization," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1524–1536, 2015.

[3] S. Kar, G. Hug, J. Mohammadi, and J. M. Moura, "Distributed state estimation and energy management in smart grids: A consensus innovations approach," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1022–1038, 2014.

[4] Y. Kuriki and T. Namerikawa, "Consensus-based cooperative formation control with collision avoidance for a multi-uav system," in *2014 American Control Conference*, pp. 2077–2082, IEee, 2014.

[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[6] F. Borrelli and T. Keviczky, "Distributed LQR design for identical dynamically decoupled systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1901–1912, 2008.

[7] Z. Qiu, S. Liu, and L. Xie, "Distributed constrained optimal consensus of multi-agent systems," *Automatica*, vol. 68, pp. 209–215, 2016.

[8] R. M. Murray *et al.*, "Optimization-based control," *California Institute of Technology, CA*, pp. 111–128, 2009.

[9] Z. Li, Z. Duan, G. Chen, and L. Huang, "Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 213–224, 2009.

[10] J. Jiao, H. L. Trentelman, and M. K. Camlibel, "Distributed linear quadratic optimal control: compute locally and act globally," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 67–72, 2019.

[11] Z. Li, G. Wen, Z. Duan, and W. Ren, "Designing fully distributed consensus protocols for linear multi-agent systems with directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 1152–1157, 2014.

[12] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Transactions on Automatic control*, vol. 59, no. 11, pp. 3051–3056, 2014.

[13] H. L. Trentelman, A. A. Stoorvogel, and M. Hautus, *Control theory for linear systems*. Springer Science & Business Media, 2012.

[14] F. Zhang, *The Schur complement and its applications*, vol. 4. Springer Science & Business Media, 2006.

[15] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[16] A. Tahbaz-Salehi and A. Jadbabaie, "A necessary and sufficient condition for consensus over random networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 791–795, 2008.