Control of Multiagent Networks as Systems: Finite-Time Algorithms, Time Transformation, and Separation Principle*

Dzung Tran†, Tansel Yucelen‡, and Burak Sarsilmaz◇

***Abstract*—This paper contributes to the studies in control of *multiagent networks as systems*. This class of multiagent networks consists of *floating agents* (i.e., agents that exchange local information) and *driver agents* (i.e., agents that not only exchange local information but also take input and output roles), where control algorithms are applied to the actuators of the driver agents based on the measurements collected from their sensors for the purpose of influencing the overall behavior of the resulting system. Specifically, we consider time-critical applications in the control of multiagent networks as systems. To this end, a finite-time control approach is proposed based on a recent *time transformation* method. The key feature of this method is that it guarantees execution of control algorithms over a prescribed time interval $[0, T)$, where T is a user-defined convergence time, based on analysis performed over a stretched, infinite-time interval $[0, \infty)$. Utilizing this method for finite-time control of multiagent networks as systems, we discuss user-defined finite-time convergence of the resulting system regardless of the initial conditions of agents and show a separation principle of the proposed time-critical algorithm. A numerical example is also presented to demonstrate the proposed system-theoretical results.**

I. Introduction

This paper contributes to the studies in control of *multiagent networks as systems* (see, e.g., [Chapter 10, 1]). This class of multiagent networks consists of *floating agents* and *driver agents*, where the former agents exchange local information through consensus or consensus-like algorithms and the latter agents not only exchange local information but also take input and output roles in the system. Here, control algorithms of interest are applied to the actuators of the driver agents based on the measurements collected from their sensors for the purpose of influencing the overall behavior of the resulting system. An example multiagent network as a system is depicted in Figure 1.

†D. Tran is a Graduate Research Assistant of the Department of Mechanical Engineering and a member of the Laboratory for Autonomy, Control, Information, and Systems (LACIS, `http://lacis.eng.usf.edu/`) at the University of South Florida, Tampa, Florida 33620, United States of America (email: `dtran3@mail.usf.edu`).

‡T. Yucelen is an Assistant Professor of the Department of Mechanical Engineering and the Director of the Laboratory for Autonomy, Control, Information, and Systems (LACIS, `/http://lacis.eng.usf.edu/`) at the University of South Florida, Tampa, Florida 33620, United States of America (email: `yucelen@usf.edu`).

◇Burak Sarsilmaz is a Graduate Research Assistant of the Department of Mechanical Engineering and a member of the Laboratory for Autonomy, Control, Information, and Systems (LACIS, `http://lacis.eng.usf.edu/`) at the University of South Florida, Tampa, Florida 33620, United States of America (email: `sarsilmaz@mail.usf.edu`).

*This research was supported by the Army Research Office under Grant `W911NF-17-1-0582`.

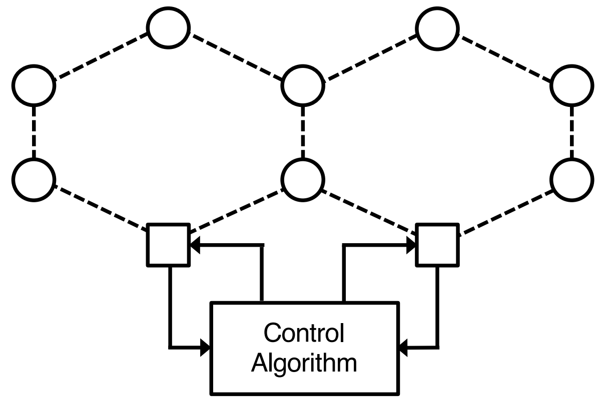

Fig. 1. An example multiagent network as a system, where *circles* denote the floating agents, *squares* denote the driver agents, *dashed lines* denote the local information exchange between all agents (graph topology), and *solid lines* denote the input-output (feedback) interaction between driver agents and a given control algorithm of interest.

In this paper, we consider time-critical applications in the control of multiagent networks as systems. In particular, a finite-time control approach is proposed based on a recent *time transformation* method [2], [3]. The key feature of this method is that it guarantees execution of control algorithms over a prescribed time interval $[0, T)$, where T is a user-defined convergence time, based on analysis performed over a stretched, infinite-time interval $[0, \infty)$. Utilizing this method for finite-time control of multiagent networks as systems, we discuss user-defined finite-time convergence of the resulting system regardless of the initial conditions of agents and show a separation principle of the proposed time-critical algorithm.

Finite-time control offers an appealing framework for time-critical applications of dynamical systems. We start with the seminal papers [4], [5], where the authors define finite-time stability for non-smooth dynamical systems. There exist many studies in the multiagent networks literature that utilize and generalize the results in these two (and similar) papers, where the finite-time convergence depends on the initial conditions of agents. The studies documented in [6]–[12] address this problem by upper bounding the finite-time convergence time and the studies documented in [2], [3], [13]–[22] propose system-theoretic tools for guaranteeing user-defined finite-time convergence regardless of the initial conditions of dynamical systems.

As noted above, the contribution of this paper builds on the novel time transformation method introduced in [2], [3] that results in smooth control algorithms (the studies in [16]–

[22] are more related than the other aforementioned ones to the contributions documented in these two papers, where we refer to [2], [3] for important differences). Our motivation behind in utilizing and generalizing the results in [2], [3] is primarily owing to the fact that their time transformation method allows one to use well-established system-theoretical tools proposed over infinite-time intervals $[0, \infty)$ for reaching guarantees over the user-defined prescribed time interval $[0, T)$. This key aspect here allows us to show the points outlined in the second paragraph.

The content of this paper is as follows. Specifically, Section II introduces the necessary mathematical preliminaries. The proposed finite-time control approach for multiagent networks as systems is given in Section III. A numerical example, which demonstrate the proposed approach, is then presented in Section IV. Finally, concluding remarks are summarized in Section V.

II. Mathematical Preliminaries

The notation used in this paper is fairly standard. Specifically, $\mathbb{R}$, $\mathbb{R}^n$, and $\mathbb{R}^{n\times m}$ respectively denote the set of real numbers, $n\times 1$ real column vectors, and $n\times m$ real matrices; $\mathbb{R}_+$ and $\mathbb{R}_+^{n\times n}$ (resp., $\overline{\mathbb{R}}_+^{n\times n}$) respectively denote the set of positive real numbers and $n \times n$ positive-definite (resp., positive semi-definite) real matrices; and 0_n, $\mathbf{1}_n$, $0_{n\times n}$, and I_n respectively denote the $n\times 1$ vector of all zeros, the $n\times 1$ vector of all ones, the $n \times n$ zero matrix, and the $n \times n$ identity matrix. In addition, we write $(\cdot)^{\mathrm{T}}$ for transpose, $(\cdot)^{-1}$ for inverse, $\|\cdot\|_2$ for the Euclidian norm, $\lambda_{\min}(A)$ (resp., $\lambda_{\max}(A)$) for the minimum (resp., maximum) eigenvalue of the symmetric matrix A, $\lambda_i(A)$ for the i-th eigenvalue of A (A is symmetric and the eigenvalues are ordered from least to greatest value), $[A]_{ij}$ for the entry of the of the matrix A on the *i*-th row and *j*-th column.

We now concisely overview several notions from graph theory (we refer to, e.g., [1] for details). In particular, an *undirected* graph $\mathcal{G}$ is defined by a set $\mathcal{V}_\mathcal{G} = \{1, \ldots, N\}$ of *nodes* and a set $\mathcal{E}_\mathcal{G} \subset \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ of *edges*. Repeated edges and self-loops are not allowed. If $(i, j) \in \mathcal{E}_\mathcal{G}$, then the nodes i and j are *neighbors* and the neighboring relation is indicated with $i \sim j$. The *degree* of a node is given by the number of its neighbors. Letting d_i be the degree of node i, then the *degree* matrix of a graph $\mathcal{G}$, $\mathcal{D}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is given by $\mathcal{D}(\mathcal{G}) = \mathrm{diag}(d)$, $d = [d_1, \ldots, d_N]^{\mathrm{T}}$. A *path* $i_0 i_1 \ldots i_L$ is a finite sequence of nodes such that $i_{k-1} \sim i_k$, $k = 1, \ldots, L$, and a graph $\mathcal{G}$ is *connected* if there is a path between any pair of distinct nodes. The *adjacency* matrix of a graph $\mathcal{G}$, $\mathcal{A}(\mathcal{G}) \in \mathbb{R}^{N\times N}$, is given by $[\mathcal{A}(\mathcal{G})]_{ij} = 1$ if $(i, j) \in \mathcal{E}_\mathcal{G}$ and $[\mathcal{A}(\mathcal{G})]_{ij} = 0$ otherwise. The *Laplacian* matrix of a graph, $\mathcal{L}(\mathcal{G}) \in \overline{\mathbb{R}}_+^{N\times N}$, is given by $\mathcal{L}(\mathcal{G}) = \mathcal{D}(\mathcal{G}) - \mathcal{A}(\mathcal{G})$. In this paper, we consider that a given multiagent network can be modeled by a connected, undirected graph $\mathcal{G}$ with nodes and edges respectively representing agents and inter-agent communication links.

Finally, the following lemma is necessary.

Lemma 1 [Fact 2.17.1, 23]. *Let $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$, and $D \in \mathbb{R}^{m\times m}$. Assume that A and D are nonsingular. Then,*

$$\begin{bmatrix} A & B \\ 0_{m\times n} & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & -A^{-1}BD^{-1} \\ 0_{m\times n} & D^{-1} \end{bmatrix}. \quad (1)$$

III. Finite-Time Control of Multiagent Networks as Systems

We first introduce the *multiagent networks as systems* setup considered in this paper. As discussed in Section I (see also Figure 1), this class of multiagent networks consists of *floating agents* and *driver agents* (see, e.g., [Chapter 10, 1]), where dynamics of each agent satisfies a single integrator form. Specifically, we propose that the floating agents execute the dynamics given by

$$\dot{x}_i(t) = \alpha\lambda(t)\big[-\sum_{i\sim j}\big(x_i(t) - x_j(t)\big)\big], \quad (2)$$

to locally exchange their state information $x_i(t)$. In addition, we propose that the driver agents execute the dynamics given by

$$\dot{x}_i(t) = \alpha\lambda(t)\big[-\sum_{i\sim j}\big(x_i(t) - x_j(t)\big) + u_i(t)\big], \quad (3)$$

and $y_i(t) = x_i(t)$ to not only locally exchange their state information $x_i(t)$ but also take input and output roles in the system, where $u_i(t)$ denotes their control inputs and $y_i(t)$ denotes their output measurements. Building on the finite-time control results documented in [2], [3], we consider that the resulting system evolves over the user-defined prescribed time interval $[0, T)$ with $T \in \mathbb{R}_+$ being a given user-defined convergence time. In addition, $\alpha \in \mathbb{R}_+$ is a gain and $\lambda(t) = 1/(T - t)$ in (2) and (3).

The above multiagent network as a system setup, which consists of a total of N agents that exchange information using their local measurements according to a connected, undirected graph $\mathcal{G}$, can be compactly written as

$$\dot{x}(t) = \alpha\lambda(t)\big(-\mathcal{L}(\mathcal{G})x(t) + Bu(t)\big),\ x(0) = x_0, \quad (4)$$

$$y(t) = B^{\mathrm{T}}x(t), \quad (5)$$

where $x(t) = \big[x_1(t),\ x_2(t),\ \ldots,\ x_N(t)\big]^{\mathrm{T}} \in \mathbb{R}^N$ denotes the aggregated state vector that captures the individual states of floating and driver agents, $u(t) \in \mathbb{R}^p$ denotes the aggregated control vector that captures the inputs applied to the set of driver agents, and $y(t) \in \mathbb{R}^p$ denotes the aggregated output vector that captures the output measurements received from the set of driver agents. Here, $\mathcal{L}(\mathcal{G}) \in \mathbb{R}^{N\times N}$ is the resulting Laplacian matrix (see Section II). In addition, $B \in \mathbb{R}^{N\times p}$ (resp., $B^{\mathrm{T}} \in \mathbb{R}^{p\times N}$) is the input (resp., output) matrix of the form $B = \begin{bmatrix} e_i & e_j & e_k & \ldots \end{bmatrix}$, where $i, j, k, \ldots$ are the corresponding indices of driver agents in the system and e_i is the column vector with *i*-th element being equal to one and other elements being equal to zero. Finally, for the following results in this paper, it is considered that the pair

$(-\mathcal{L}(\mathcal{G}), B)$ is stabilizable. Since $\mathcal{L}(\mathcal{G})$ is symmetric, the pair $(-\mathcal{L}(\mathcal{G}), B^{\mathrm{T}})$ is detectable by duality.

We now present the proposed control algorithm, which is applied to the actuators of the driver agents based on the measurements collected from their sensors for the purpose of influencing the overall behavior of the resulting system (see Figure 1). Specifically, we propose the finite-time control algorithm over user-defined prescribed time interval $[0, T)$ given by

$$u(t) = K_1\hat{x}(t) + K_2 z(t) + K_3 c(t), \tag{6}$$

$$\dot{\hat{x}}(t) = \alpha\lambda(t)\big(-\mathcal{L}(\mathcal{G})\hat{x}(t) + Bu(t) + H(y(t) - B^{\mathrm{T}}\hat{x}(t)\big), \quad \hat{x}(0) = \hat{x}_0, \tag{7}$$

$$\dot{z}(t) = \alpha\lambda(t)\big(A_c z(t) + B_{c1}\hat{x}(t) + B_{c2}c(t)\big), \quad z(0) = z_0, \tag{8}$$

where $K_1 \in \mathbb{R}^{p\times N}$, $K_2 \in \mathbb{R}^{p\times p}$, $K_3 \in \mathbb{R}^{p\times p}$, $H \in \mathbb{R}^{N\times p}$, $A_c \in \mathbb{R}^{p\times p}$, $B_{c1} \in \mathbb{R}^{p\times N}$, and $B_{c2} \in \mathbb{R}^{p\times p}$. In the execution of the control signal given by (6) since the aggregated state vector is not available, one needs to reconstruct the aggregated state vector through the state estimation algorithm given by (7) with $\hat{x}(t) \in \mathbb{R}^N$ denoting the estimated state. In addition, to give a designer the flexibility in achieving different sets of control objectives, (8) denotes the dynamic compensator with $z(t) \in \mathbb{R}^p$ denoting the dynamic compensator state and $c(t) \in \mathbb{R}^p$ denoting a command of interest. Here, we consider that $c(t)$ and $\dot{c}(t)$ are bounded for $t \geq 0$ and $\dot{c}(t)$ is a piecewise continuous function. Note that the finite-time control algorithm given by (6), (7), and (8) are inside the bottom "box" shown in Figure 1. That is, based on the measurements $y(t)$ collected from the sensors of driver agents, an operator executes this control algorithm through injecting $u(t)$ to the actuators of these agents. The aforementioned sets of control objectives that can be achieved with this control algorithm over user-defined prescribed time interval $[0, T)$ is discussed at the end of this section.

We next discuss the stability of the controlled multiagent network as a system based on time transformation and highlight the separation principle. In particular, let the state estimation error be $\tilde{x}(t) \triangleq x(t) - \hat{x}(t)$. This error evolves according to the dynamics given by

$$\begin{aligned}
\dot{\tilde{x}}(t) &= \alpha\lambda(t)\big(-\mathcal{L}(\mathcal{G})x(t) + Bu(t) \\
&\quad + \mathcal{L}(\mathcal{G})\hat{x}(t) - Bu(t) - H(y(t) - B^{\mathrm{T}}\hat{x}(t))\big) \\
&= \alpha\lambda(t)\big(-\mathcal{L}(\mathcal{G})\tilde{x}(t) - HB^{\mathrm{T}}\tilde{x}(t)\big) \\
&= -\alpha\lambda(t)\big(\mathcal{L}(\mathcal{G}) + HB^{\mathrm{T}}\big)\tilde{x}(t) \\
&= -\alpha\lambda(t)F\tilde{x}(t), \quad \tilde{x}(0) = \tilde{x}_0,
\end{aligned} \tag{9}$$

where $F \triangleq \big(\mathcal{L}(\mathcal{G}) + HB^{\mathrm{T}}\big) \in \mathbb{R}^{N\times N}$. Here, we assume that $-F$ is Hurwitz by a proper selection of the observer gain matrix H. We note that since $(-\mathcal{L}(\mathcal{G}), B^{\mathrm{T}})$ is detectable, one can always find an observer gain matrix H such that $-F$ is Hurwitz. In addition, since $\mathcal{L}(\mathcal{G})$ is also symmetric, a proper selection of H provides that all eigenvalues of F (resp., $-F$) are positive (resp., negative). For example, one trivial selection is $H = B$ that results in $F = \big(\mathcal{L}(\mathcal{G}) + HB^{\mathrm{T}}\big) = \big(\mathcal{L}(\mathcal{G}) + BB^{\mathrm{T}}\big)$ with BB^{T} being a diagonal matrix with ones and zeros on its diagonal. From Lemma 2 in [24] or Lemma 3.3 in [25], it follows that this selection of the observer gain matrix creates a positive-definite F matrix owing to the connected, undirected graph topology.

Now, using the proposed control algorithm (6) in (4), one can write

$$\begin{aligned}
\dot{x}(t) &= \alpha\lambda(t)\Big(-\mathcal{L}(\mathcal{G})x(t) + B\big(K_1\hat{x}(t) \\
&\qquad + K_2 z(t) + K_3 c(t)\big)\Big) \\
&= \alpha\lambda(t)\Big(-\mathcal{L}(\mathcal{G})x(t) + B\big(K_1(x(t) - \tilde{x}(t)) \\
&\qquad + K_2 z(t) + K_3 c(t)\big)\Big) \\
&= \alpha\lambda(t)\Big(-(\mathcal{L}(\mathcal{G}) - BK_1)x(t) - BK_1\tilde{x}(t) \\
&\qquad + BK_2 z(t) + BK_3 c(t)\Big), \ x(0) = x_0.
\end{aligned} \tag{10}$$

In addition, the dynamic compensator given by (8) can be rewritten as

$$\dot{z}(t) = \alpha\lambda(t)\big(A_c z(t) + B_{c1}x(t) - B_{c1}\tilde{x}(t) + B_{c2}c(t)\big), \quad z(0) = z_0. \tag{11}$$

At this point, let $r(t) \triangleq \big[x^{\mathrm{T}}(t),\ z^{\mathrm{T}}(t),\ \tilde{x}^{\mathrm{T}}(t)\big]^{\mathrm{T}} \in \mathbb{R}^{2N+p}$. From (9), (10), and (11), one can now write

$$\begin{aligned}
\dot{r}(t) &= \alpha\lambda(t)\underbrace{\begin{bmatrix} -(\mathcal{L}(\mathcal{G}) - BK_1) & BK_2 & -BK_1 \\ B_{c1} & A_c & -B_{c1} \\ 0_{N\times N} & 0_{N\times p} & -F \end{bmatrix}}_{M} r(t) \\
&\qquad + \alpha\lambda(t)\underbrace{\begin{bmatrix} BK_3 \\ B_{c2} \\ 0_{N\times p} \end{bmatrix}}_{N} c(t) \\
&= \alpha\lambda(t)\big(Mr(t) + Nc(t)\big), \ r(0) = r_0.
\end{aligned} \tag{12}$$

Here, our finite-time control goal is $\lim_{t\to T^-}\big(r(t) + M^{-1}Nc(t)\big) = 0$, where "$-M^{-1}Nc(t)$" can capture different sets of control objectives. To achieve this goal, one needs to make the system matrix M in (12) Hurwitz. For this purpose, let $M \triangleq \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}$, where $M_1 \triangleq \begin{bmatrix} -(\mathcal{L}(\mathcal{G}) - BK_1) & BK_2 \\ B_{c1} & A_c \end{bmatrix}$, $M_2 \triangleq \begin{bmatrix} -BK_1 \\ -B_{c1} \end{bmatrix}$, $M_3 \triangleq 0_{N\times(N+p)}$ and $M_4 \triangleq -F$. Obviously, the spectrum of M is equal to the union of the spectrums of M_1 and M_4 owing to the upper block triangular structure of M. This shows the separation principle. Because, one can judiciously select the controller gain matrices K_1 and K_2 to make M_1 Hurwitz (see below) and select the observer gain matrix H to make M_4 Hurwitz (see the discussion below (9)). Hence, the design processes for making M_1 and M_4 both Hurwitz are independent. We now further elaborate how the controller gain matrices K_1 and K_2 can be selected to render M_1

Hurwitz. To this end, note that

$$\begin{aligned} M_1 &= \begin{bmatrix} -(\mathcal{L}(\mathcal{G})-BK_1) & BK_2 \\ B_{c1} & A_c \end{bmatrix} \\ &= \begin{bmatrix} -\mathcal{L}(\mathcal{G}) & 0_{N\times p} \\ B_{c1} & A_c \end{bmatrix} + \begin{bmatrix} BK_1 & BK_2 \\ 0_{p\times N} & 0_{p\times p} \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} -\mathcal{L}(\mathcal{G}) & 0_{N\times p} \\ B_{c1} & A_c \end{bmatrix}}_{\tilde{A}} + \underbrace{\begin{bmatrix} B \\ 0_{p\times p} \end{bmatrix}}_{\tilde{B}} \underbrace{\begin{bmatrix} K_1 & K_2 \end{bmatrix}}_{\tilde{K}} \\ &= \tilde{A} + \tilde{B}\tilde{K}. \end{aligned} \quad (13)$$

Hence, if

$$\left(\begin{bmatrix} -\mathcal{L}(\mathcal{G}) & 0_{N\times p} \\ B_{c1} & A_c \end{bmatrix}, \begin{bmatrix} B \\ 0_{p\times p} \end{bmatrix} \right), \quad (14)$$

is stabilizable, then there always exist controller gain matrices K_1 and K_2 such that M_1 is Hurwitz. From Lemma 1.26 of [26], note also that when $(-\mathcal{L}(\mathcal{G}), B)$ is stabilizable and

$$\operatorname{rank} \begin{bmatrix} -\mathcal{L}(\mathcal{G}) - \lambda \mathrm{I}_N & B \\ B_{c1} & 0_{p\times p} \end{bmatrix} = N + p, \quad (15)$$

for all $\lambda \in \sigma(A_c)$ with $\sigma(A_c)$ denoting the spectrum of A_c, then the pair in (14) is stabilizable.

We are now ready to state the main result.

Theorem 1. *Consider a multiagent network as a system given by (4) and (5), where agents exchange information using their local measurements according to a connected, undirected graph $\mathcal{G}$. In addition, if the driver agents execute the proposed controller architecture given by (6), (7) and (8) and M is Hurwitz, then*

$$\lim_{t\to T^-} \big(r(t) + M^{-1}Nc(t)\big) = 0, \quad (16)$$

and the solution to (12) is bounded for $t \in [0,\ T)$.

Due to page limitations, the proof of this result will be reported elsewhere. For interested readers, it follows by first defining $q(t) \triangleq r(t) + M^{-1}Nc(t)$. One can then write

$$\begin{aligned} \dot{q}(t) &= \dot{r}(t) + M^{-1}N\dot{c}(t) \\ &= \alpha\lambda(t)\big(M(q(t) - M^{-1}Nc(t)) + Nc(t)\big) \\ &\qquad + M^{-1}N\dot{c}(t) \\ &= \alpha\lambda(t)Mq(t) + M^{-1}Nc_d(t),\ q(0) = q_0, \end{aligned} \quad (17)$$

where $c_d(t) \triangleq \dot{c}(t)$. One can now use the time transformation method utilized in [2] and [3] (see also [Section 1.1.1.4, 27]) with the time transformation function $\theta(p) \triangleq T(1-e^{-p}) \triangleq t$ to transform the error dynamics given by (17) into

$$\psi'(p) = \alpha M\psi(p) + h(p)c_d^*\big(\theta(p)\big),\ \psi(0) = q_0, \quad (18)$$

$$h'(p) = -h(p),\ h(0) = h_0. \quad (19)$$

where $\psi(p) \triangleq \xi(t)$ is the solution to the dynamical system given by (17) and

$$h(p) \triangleq \frac{dt}{dp} = \frac{d\theta(p)}{dp} = Te^{-p}. \quad (20)$$

By considering the Lyapunov function candidate $V(\psi, h) = \psi^{\mathrm{T}}P\psi + \eta h^2$, where $\eta \in \mathbb{R}_+$ and $P \in \mathbb{R}_+^{(2N+p)\times(2N+p)}$ is the solution of the Lyapunov equation $M^{\mathrm{T}}P + PM = -\mathrm{I}$, the result then follows.

Theorem 1 also establishes the boundedness of the control signal $u(t)$ given by (6) over $[0, T)$. Note that one can also view the term "$\alpha\lambda(t)(-\mathcal{L}(\mathcal{G})x(t) + Bu(t))$" in (4) as the total control signal. Motivated from this standpoint, the boundedness of this signal can be shown by proving that $s(t) \triangleq \dot{r}(t)$ is bounded over $[0, T)$. Similar to the discussion given in the above paragraph, here one needs to form dynamics of $s(t)$, transform the resulting dynamics to the stretched time interval, and then do the analysis on that interval to conclude the boundedness of $s(t)$, which is possible when M and $\mathcal{M} \triangleq \mathrm{I} + \alpha M \in \mathbb{R}^{2N+p}$ are both Hurwitz. Once $\dot{r}(t) \triangleq \big[\dot{x}^{\mathrm{T}}(t),\ \dot{z}^{\mathrm{T}}(t),\ \dot{\tilde{x}}^{\mathrm{T}}(t)\big]^{\mathrm{T}}$ is bounded and since the total control signal (i.e., "$\alpha\lambda(t)(-\mathcal{L}(\mathcal{G})x(t) + Bu(t))$" in (4)) equals to the first entry in $\dot{r}(t)$, the boundedness of this total control signal is now immediate. Once again, details will be reported elsewhere due to page limitations.

Since $r(t)$ approaches $-M^{-1}Nc(t)$ in the sense of (16) at the user-defined convergence time T, we finally elaborate the structure of "$-M^{-1}N$" to show how it can be constructed to capture different sets of control objectives. Recall that we consider above the partitioned matrix given by $M = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}$. One can also partition the matrix N as $N = \begin{bmatrix} N_1 \\ N_2 \end{bmatrix}$ with $N_1 = \begin{bmatrix} BK_3 \\ B_{c2} \end{bmatrix}$ and $N_2 = 0_{N\times p}$. Under the conditions of Lemma 1, note that one can write

$$M^{-1} = \begin{bmatrix} M_1^{-1} & -M_1^{-1}M_2M_4^{-1} \\ 0_{N\times(N+p)} & M_4^{-1} \end{bmatrix}. \quad (21)$$

As a result, we obtain

$$-M^{-1}N = \begin{bmatrix} -M_1^{-1}N_1 \\ 0_{N\times p} \end{bmatrix}, \quad (22)$$

where the last equality results from the fact that $N_2 = 0_{N\times p}$. This implies that $\lim_{t\to T^-} \tilde{x}(t) = 0$. Next, let $\bar{r}(t) \triangleq \big[x(t),\ z(t)\big]^{\mathrm{T}} \in \mathbb{R}^{N+p}$. From (22), $\lim_{t\to T^-}\big(\bar{r}(t) + M_1^{-1}N_1c(t)\big) = 0$ clearly holds. To further elaborate the structure of the term "$-M_1^{-1}N_1$", let

$$M_1 \triangleq \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \begin{bmatrix} -(\mathcal{L}(\mathcal{G}) - BK_1) & BK_2 \\ B_{c1} & A_c \end{bmatrix}, \quad (23)$$

$$N_1 \triangleq \begin{bmatrix} N_{11} \\ N_{21} \end{bmatrix} = \begin{bmatrix} BK_3 \\ B_{c2} \end{bmatrix}, \quad (24)$$

with $M_{11}, M_{12}, M_{21}, M_{22}, N_{11}, N_{21}$ being the corresponding matrices. Since M_1 can always be made Hurwitz (see discussion before Theorem 1), we define $M_1^{-1} \triangleq \begin{bmatrix} \bar{M}_{11} & \bar{M}_{12} \\ \bar{M}_{21} & \bar{M}_{22} \end{bmatrix}$. As a result, "$-M_1^{-1}N_1$" can be equivalently written as

$$-M_1^{-1}N_1 = -\begin{bmatrix} \bar{M}_{11}N_{11} + \bar{M}_{12}N_{21} \\ \bar{M}_{21}N_{11} + \bar{M}_{22}N_{21} \end{bmatrix}. \quad (25)$$

Depending on the considered control objective, one can design the structure of (25) accordingly.

To elucidate (25), consider, for example, N agents on an undirected path graph with the first agent being the driver agent. Assume that one would like all agents in the multiagent network to converge to a spatial location, $c(t)$, at the user-defined convergence time T (i.e., rendezvous). In this case, we can set $K_1 = -\frac{1}{N}\mathbf{1}_N^{\mathrm{T}}$, $K_2 = 0$, $K_3 = 1$, $A_c = -1$, $B_{c1} = 0$, and $B_{c2} = 0$, where we assume $-(\mathcal{L}(\mathcal{G}) + \frac{1}{N}B\mathbf{1}_N^{\mathrm{T}})$ is Hurwitz, then these selections result in $-M_1^{-1}N_1 c(t) = [\mathbf{1}_N^{\mathrm{T}}, 0]^{\mathrm{T}} c(t)$. Therefore, $\lim_{t\to T^-}\big(x(t) - \mathbf{1}_N c(t)\big) = 0$ and $\lim_{t\to T^-} z(t) = 0$, where the former convergence yields $\lim_{t\to T^-}\big(x_i(t) - c(t)\big) = 0$. For another example, we also refer to the next section.

IV. Illustrative Numerical Example

In this section, we show a numerical example to illustrate the efficacy of the proposed control architecture presented in Section III. Specifically, consider a multiagent network with 4 agents (i.e., $N = 4$) subject to an undirected path graph, where agents 2 and 3 are the driver agents ($p = 2$) while agents 1 and 4 are the floating agents. This selection of driver and floating agents gives

$$B^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{26}$$

for (4) and (5). The pair $(-\mathcal{L}(\mathcal{G}), B)$ is controllable in this case. In addition, we set $\alpha = 10$ in (4) and all agents are assigned to random initial conditions.

In the below numerical example, the objective is to design a control algorithm to split the network such that agents 1 and 2 reach to a desired command ($c_1(t) = 3 + 3\sin(0.5t) + \sin(4t)$ is used) and agent 3 and 4 reach to another desired command ($c_2(t) = -2 + 1.5\cos(0.8t)$ is used) at $T = 5$ seconds. It should be noted based on Theorem 1 and (25) that

$$\lim_{t\to T^-}\big(x(t) + (\bar{M}_{11}N_{11} + \bar{M}_{12}N_{21})c(t)\big) = 0. \tag{27}$$

Therefore, with the given objective, we need to select gain matrices such that

$$S \triangleq -(\bar{M}_{11}N_{11} + \bar{M}_{12}N_{21}), \tag{28}$$

with

$$S^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{29}$$

We first concisely explain the process of designing the gain matrices $K_1, K_2, K_3, H, A_c, B_{c1}, B_{c2}$ for the proposed control architecture. Specifically, recall that $F = \big(\mathcal{L}(\mathcal{G}) + HB^{\mathrm{T}}\big)$, where H needs to be designed such that F is a matrix with positive real part eigenvalues. As discussed in the paragraph below (9), a simple choice $H = B$ makes F a positive definite matrix. In order to use the dynamic compensator (8) as an integrator in this case for allowing agents 1 and 2 to reach to $c_1(t)$ and agents 3 and 4 to reach to $c_2(t)$, we set $A_c = 0$ and

$$B_{c1} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}, \tag{30}$$

$$B_{c2} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{31}$$

Note that the condition (15) is satisfied. Thus, by the discussion before Theorem 1, we can always find K_1, K_2 such that M_1 is Hurwitz. Utilizing linear quadratic regulator design, we obtain

$$K_1 = \begin{bmatrix} -1.1956 & -1.0722 & -0.3942 & -0.1387 \\ -0.1387 & -0.3942 & -1.0722 & -1.1956 \end{bmatrix}, \tag{32}$$

$$K_2 = \begin{bmatrix} -1.4142 & 0 \\ 0 & -1.4142 \end{bmatrix}. \tag{33}$$

In other words, M_1 can now be constructed and M_1^{-1} can be obtained. As mentioned earlier, based on (16) and (25), we have $\lim_{t\to T^-}\big(x(t) + (\bar{M}_{11}N_{11} + \bar{M}_{12}N_{21})c(t)\big) = \lim_{t\to T^-}\big(x(t) - Sc(t)\big) = 0$ where S is given by (29), $N_{11} = BK_3$, $N_{21} = B_{c2}$, and $\bar{M}_{11}$ and $\bar{M}_{12}$ are directly obtained from M_1^{-1}. In this case, one can choose

$$K_3 = \begin{bmatrix} 1.9692 & -0.2462 \\ -0.2462 & 2.0308 \end{bmatrix} \tag{34}$$

to finally satisfy (29). With $\alpha = 10$ and the above gain matrices, M and $\mathcal{M} \triangleq \mathrm{I} + \alpha M \in \mathbb{R}^{2N+p}$ are both Hurwitz.

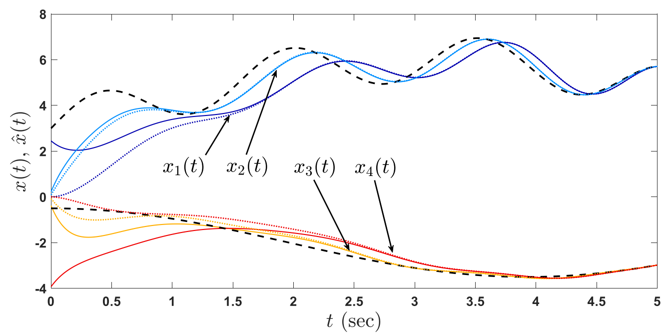

Fig. 2. Response of the multiagent network as a system (the dashed lines denote the trajectories of the commands $c_1(t)$ and $c_2(t)$, the solid lines denote actual states $x(t)$ of all agents, and the dotted lines denotes the state estimation $\hat{x}(t)$).

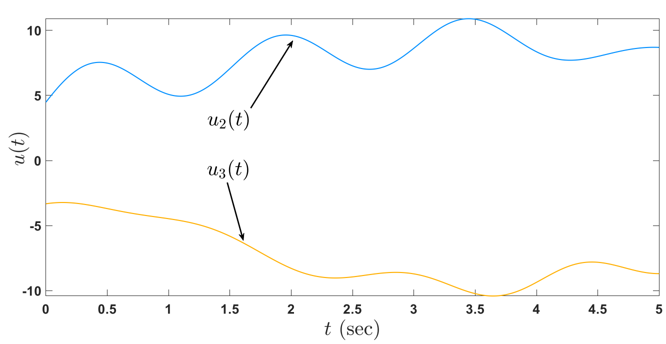

Fig. 3. The control signals of driver agents (i.e., agents 2 and 3 in the considered multiagent network as a system).

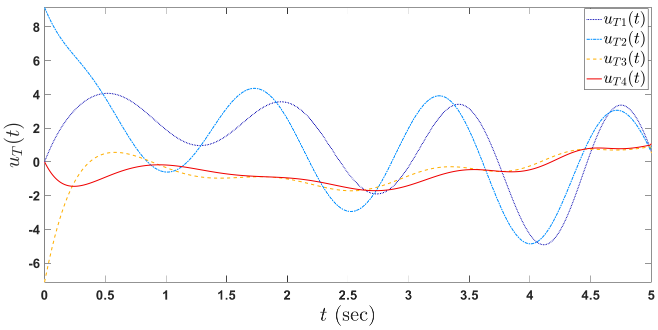

Fig. 4. The total control signals $u_T(t)$ depicted by the right hand side of (4) of all agents.

The response of the multiagent network as a system in this example given by (4) and (5) under the proposed control architecture given by (6), (7), and (8) is shown in Figure 2, where the dashed lines denote the trajectories of the commands $c_1(t)$ and $c_2(t)$, the solid lines denote the actual states $x(t)$ of all agents, and the dotted lines denote the state estimation $\hat{x}(t)$. As expected from the result (16) along with the above selection of the gain matrices, this figure shows that the states of agents 1 and 2 converge to $c_1(t)$ and the states of agents 3 and 4 converge to $c_2(t)$ at $T = 5$ seconds. Figures 3 and 4 also show the resulting control signals.

V. Conclusion

We focused on multiagent networks as systems and proposed a new finite-time control algorithm using a recent time transformation method. Specifically, based on a given user-defined finite-time interval $[0, T)$, we showed that the proposed algorithm guarantees the time-critical completion of a given system-level control objective at T seconds regardless of the initial conditions of agents. In addition, it was shown that the separation principle holds for the proposed finite-time control algorithm in the sense that one can select the observer and controller gain matrices independently. Finally, an illustrative numerical example demonstrated the efficacy of our theoretical results.

References

[1] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.
[2] Z. Kan, T. Yucelen, E. Doucette, and E. Pasiliao, "A finite-time consensus framework over time-varying graph topologies with temporal constraints," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 7, p. 071012, 2017.
[3] T. Yucelen, Z. Kan, and E. Pasiliao, "Finite-time cooperative engagement," *IEEE Transactions on Automatic Control*, 2018, (submitted).
[4] S. P. Bhat and D. S. Bernstein, "Continuous finite-time stabilization of the translational and rotational double integrators," *IEEE Transactions on automatic control*, vol. 43, no. 5, pp. 678–682, 1998.
[5] ——, "Finite-time stability of continuous autonomous systems," *SIAM Journal on Control and Optimization*, vol. 38, no. 3, pp. 751–766, 2000.
[6] M. Basin, Y. Shtessel, and F. Aldukali, "Continuous finite-and fixed-time high-order regulators," *Journal of the Franklin Institute*, vol. 353, no. 18, pp. 5001–5012, 2016.
[7] H. Hong, W. Yu, G. Wen, and X. Yu, "Distributed robust fixed-time consensus for nonlinear and disturbed multiagent systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1464–1473, 2017.
[8] B. Tian, Z. Zuo, and H. Wang, "Leader–follower fixed-time consensus of multi-agent systems with high-order integrator dynamics," *International Journal of Control*, vol. 90, no. 7, pp. 1420–1427, 2017.
[9] W. Lu, X. Liu, and T. Chen, "A note on finite-time and fixed-time stability," *Neural Networks*, vol. 81, pp. 11–15, 2016.
[10] H. B. Oza, Y. V. Orlov, and S. K. Spurgeon, "Robust finite time stability and stabilisation: A survey of continuous and discontinuous paradigms," in *2014 International Workshop on Variable Structure Systems*. IEEE, 2014, pp. 1–7.
[11] A. Polyakov, "Nonlinear feedback design for fixed-time stabilization of linear control systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 8, pp. 2106–2110, 2012.
[12] H. Ríos and A. R. Teel, "A hybrid observer for fixed-time state estimation of linear systems," in *2016 Conference on Decision and Control*. IEEE, 2016, pp. 5408–5413.
[13] J. D. Sánchez-Torres, E. N. Sanchez, and A. G. Loukianov, "Predefined-time stability of dynamical systems with sliding modes," in *2015 American Control Conference*. IEEE, 2015, pp. 5842–5846.
[14] E. Jiménez-Rodríguez, J. D. Sánchez-Torres, D. Gómez-Gutiérrez, and A. G. Loukianov, "Predefined-time tracking of a class of mechanical systems," in *2016 International Conference on Electrical Engineering, Computing Science and Automatic Control*. IEEE, 2016, pp. 1–5.
[15] E. Jiménez-Rodríguez, J. D. Sánchez-Torres, and A. G. Loukianov, "On optimal predefined-time stabilization," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 17, pp. 3620–3642, 2017.
[16] C. Yong, X. Guangming, and L. Huiyang, "Reaching consensus at a preset time: Single-integrator dynamics case," in *2012 Chinese Control Conference*. IEEE, 2012, pp. 6220–6225.
[17] ——, "Reaching consensus at a preset time: Double-integrator dynamics case," in *2012 Chinese Control Conference*. IEEE, 2012, pp. 6309–6314.
[18] C. Wang, G. Xie, and M. Cao, "Forming circle formations of anonymous mobile agents with order preservation," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3248–3254, 2013.
[19] J. Wen, C. Wang, W. Luo, and G. Xie, "Finite-time consensus of networked multiagent systems with time-varying linear control protocols," *Mathematical Problems in Engineering*, vol. 2016, 2016.
[20] N. Harl and S. Balakrishnan, "Impact time and angle guidance with sliding mode control," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 6, pp. 1436–1449, 2012.
[21] H. Wang, C. Wang, and G. Xie, "Finite-time containment control of multi-agent systems with static or dynamic leaders," *Neurocomputing*, vol. 226, pp. 1–6, 2017.
[22] Y. Wang, Y. Song, D. J. Hill, and M. Krstic, "Prescribed-time consensus and containment control of networked multiagent systems," *IEEE Transactions on Cybernetics*, 2018.
[23] D. S. Bernstein, *Matrix mathematics: Theory, facts, and formulas*. Princeton University Press, 2009.
[24] T. Yucelen and J. D. Peterson, "Active-passive networked multiagent systems," in *2014 Conference on Decision and Control*. IEEE, 2014, pp. 6939–6944.
[25] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative control of multi-agent systems: Optimal and adaptive design approaches*. Springer Science & Business Media, 2014.
[26] J. Huang, *Nonlinear output regulation: Theory and applications*. SIAM, 2004, vol. 8.
[27] P. Benner, R. Findeisen, D. Flockerzi, U. Reichl, K. Sundmacher, and P. Benner, *Large-scale networks in engineering and life sciences*. Springer, 2016.