Nonlinear Analysis: Hybrid Systems 2 (2008) 1152–1167

Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

journal homepage: www.elsevier.com/locate/nahs

PdE-based model reference adaptive control of uncertain heterogeneous multiagent networks

Jun Kim [a], Kyoung-Dae Kim [a], Vivek Natarajan [a], Scott David Kelly [b], Joseph Bentsman [a,*]

[a] Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, 1206 West Green Street, Urbana, IL 61801-2906, United States

[b] Department of Mechanical Engineering and Engineering Science, The University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223-0001, United States

ARTICLE INFO

Article history:
Received 22 March 2008
Accepted 4 September 2008

Keywords:
Model reference adaptive control
Decentralized control
Multiagent networks
Partial difference equations on graphs
Parameter uncertainty

ABSTRACT

Robust globally stable model reference adaptive control (MRAC) laws recently derived for systems described by parabolic and hyperbolic partial differential equations (PDEs) with spatially-varying coefficients under distributed sensing and actuation are extended to heterogeneous multiagent networks characterized by parameter uncertainty. The extension is carried out using partial difference equations (PdEs) on graphs that preserve parabolic- and hyperbolic-like cumulative network behavior. Unlike in the PDE case, only boundary input is specified for the reference model. The algorithms proposed directly incorporate this boundary reference input into the reference PdE to generate the distributed admissible reference evolution profile followed by the agents. The agent evolution thus depends only on the interaction with the adjacent agents, making the system fully decentralized. Numerical examples are presented as well, including the case of the switched topology associated with a sudden loss of an agent. The resulting PdE MRAC laws inherit the robust linear structure of their PDE counterparts.

1. Introduction

Control designs for systems represented by partial differential equations (PDEs) can be applied to multiagent networks described by partial difference equations (PdEs) on graphs by properly identifying elements of the theory of differential operators on graphs with their analogs in the spatially-continuous setting. This perspective, developed in [1,2,6], and references therein, permits viewing the collective behavior of coordinated agents in a network as the qualitative analog of the solution behavior of a PDE with differential operator structure matching, in some sense, that of the multiagent network dynamics. If the structure of such a PDE admits specifying some desirable properties of its spatiotemporal evolution, then forcing the multiagent network to track this evolution can be, in turn, specified as the control objective. Realization of this viewpoint, in the context of relating the double integrator dynamics of a single agent to the wave equation, for the multiagent configuration was first carried out in [14]. Subsequently decentralized control schemes for the consensus problem based on the damped wave equation were developed in [7]. The present paper extends the robust model reference adaptive control (MRAC) technique for spatially-varying PDEs under distributed sensing and actuation recently introduced in [9] – an approach based on the use of local information only – to the decentralized adaptive control of heterogeneous multiagent networks comprising agents with parameter uncertainty. The resulting control setting is capable of addressing non-trivial collective behavior tasks [12] for multiagent networks, such as, for example, mobile decentralized autonomous underwater sensor arrays. In the latter context, the setting proposed admits forming a decentralized reference evolution

* Corresponding author. Tel.: +1 217 244 1076; fax: +1 217 244 6534.
E-mail address: jbentsma@uiuc.edu (J. Bentsman).

1751-570X/$ – see front matter
doi:10.1016/j.nahs.2008.09.008

profile [13] for a network consisting of agents with not necessarily equal masses and forcing this network to track this profile under slow changes of individual agents' masses over time due to fuel depletion. The desired network evolution profiles are directly incorporated into the MRAC algorithm structure. The MRAC laws presented ensure tracking and fully inherit the linear structure of their PDE progenitors, yielding globally stable behavior. Numerical examples are given to demonstrate the performance of the algorithms proposed.

In the agent topology considered, some of the agents are assumed to have access to the desired trajectories through, for example, specialized sensing and/or communication tools, whereas the motion of the rest of the agents is generated solely through communication with their neighbors. The relevant structures of [9], are determined by the control objective. The resulting topologies permit, for example, mobile sensor array areal sweeps, where a large number of agents is required to navigate vast domains, but it is expensive to equip each of the agents with complex sensing and communication tools.

2. Preliminaries

Following [1,6], a multiagent network with a certain communication structure will be represented as an undirected unweighted graph $G = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N} = \{1, 2, \dots, N\}$ is the set of nodes, with each node corresponding to an individual agent, and $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is the set of edges, with each edge corresponding to a communication link between two agents. We will identify a node in $\mathcal{N}$ with the corresponding agent's index and denote the latter by x, so that $x \in \mathcal{N}$. The undirectedness of such a graph indicates that each of any two agents linked by an edge can access the other's state as well as its own. A graph G is said to be *connected* if every node in $\mathcal{N}$ can be reached from every other node in $\mathcal{N}$ along a sequence of edges in $\mathcal{E}$.

Functions on graphs and operators on such functions are defined as follows [1,6]. Given a vector-valued function $f : \mathcal{N} \mapsto \mathbb{R}^n$, its integral over G is defined as

$$\int_G f = \sum_{x \in \mathcal{N}} f(x).$$

$L^2(G)$ is defined as the Hilbert space comprising such functions, with the inner product defined by

$$(f, g) = \int_G f^{\mathrm{T}} g,$$

where f^{T} represents the transpose of the column vector $f \in \mathbb{R}^n$, and the corresponding norm defined by

$$\|f\| = \sqrt{\left(\int_G f^{\mathrm{T}} f\right)}.$$

Partial differentiation on G is defined as

$$\partial_y f(x) = f(y) - f(x), \qquad \partial_y^2 f(x) = \partial_y f(y) - \partial_y f(x) = -\partial_y f(x)$$

for $x, y \in \mathcal{N}$. The Laplacian on G is, therefore, given by

$$\Delta f(x) = -\sum_{y \sim x} \partial_y^2 f(x) = \sum_{y \sim x} \partial_y f(x),$$

where $y \sim x$ means that nodes x and y are connected by an edge. We will make use of the following properties of the Laplacian.

Lemma 1. *The operator $-\Delta$ is non-negative, i.e.,*

$$(-\Delta f, f) \geq 0$$

for all $f \in L^2(G)$ [4].

Proof. Note that

$$-(\partial_{y_1} f(x_1))^{\mathrm{T}} f(x_1) = -(f(y_1) - f(x_1))^{\mathrm{T}} f(x_1) = -f(y_1)^{\mathrm{T}} f(x_1) + f(x_1)^{\mathrm{T}} f(x_1) \tag{1}$$

and

$$-(\partial_{x_1} f(y_1))^{\mathrm{T}} f(y_1) = -(f(x_1) - f(y_1))^{\mathrm{T}} f(y_1) = -f(x_1)^{\mathrm{T}} f(y_1) + f(y_1)^{\mathrm{T}} f(y_1) \tag{2}$$

for $y_1 \sim x_1$. Adding (1) and (2), we obtain

$$-(\partial_{y_1} f(x_1))^{\mathrm{T}} f(x_1) - (\partial_{x_1} f(y_1))^{\mathrm{T}} f(y_1) = f(x_1)^{\mathrm{T}} f(x_1) - 2f(x_1)^{\mathrm{T}} f(y_1) + f(y_1)^{\mathrm{T}} f(y_1) = \|f(x_1) - f(y_1)\|^2. \tag{3}$$

Since

$$(-\Delta f, f) = -\left(\sum_{y\sim x} \partial_y f(x), f(x)\right) = -\sum_{x\in\mathcal{N}}\sum_{y\sim x} (\partial_y f(x))^{\mathrm{T}} f(x)$$

is the sum of terms of form (3), $-\Delta$ is a positive operator. ■

Depending on the context, f is used throughout the paper to denote generic function, force, or control input. To admit time evolution of objects associated with graph nodes, let us generalize the function $f : \mathcal{N} \mapsto \mathbb{R}^n$ to the form $f : \mathcal{N} \times \mathbb{R}^+ \to \mathbb{R}^n$. Then the following is true.

Lemma 2. *For $f : \mathcal{N} \times \mathbb{R}^+ \to \mathbb{R}^n$, i.e., $f(\cdot, t) \in L^2(G)$ for all $t \in \mathbb{R}^+$,*

$$\frac{\mathrm{d}}{\mathrm{d}t}(\Delta f, f) = 2(\Delta f, \dot{f}) = 2(\Delta \dot{f}, f)$$

where $\dot{f}(x, t)$ denotes the time derivative of $f(x, t)$.

Proof. Note that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left((\partial_{y_1} f(x_1, t))^{\mathrm{T}} f(x_1, t)\right) = \dot{f}(y_1, t)^{\mathrm{T}} f(x_1, t) - \dot{f}(x_1, t)^{\mathrm{T}} f(x_1, t) + f(y_1, t)^{\mathrm{T}} \dot{f}(x_1, t) - f(x_1, t)^{\mathrm{T}} \dot{f}(x_1, t) \tag{4}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}\left((\partial_{x_1} f(y_1, t))^{\mathrm{T}} f(y_1, t)\right) = \dot{f}(x_1, t)^{\mathrm{T}} f(y_1, t) - \dot{f}(y_1, t)^{\mathrm{T}} f(y_1, t) + f(x_1, t)^{\mathrm{T}} \dot{f}(y_1, t) - f(y_1, t)^{\mathrm{T}} \dot{f}(y_1, t) \tag{5}$$

for $y_1 \sim x_1$. Adding (4) and (5), we obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left((\partial_{y_1} f(x_1, t))^{\mathrm{T}} f(x_1, t) + (\partial_{x_1} f(y_1, t))^{\mathrm{T}} f(y_1, t)\right) &= 2\left(f(y_1, t) - f(x_1, t)\right)^{\mathrm{T}} \left(\dot{f}(x_1, t) - \dot{f}(y_1, t)\right) \\ &= 2(\partial_{y_1} f(x_1, t))^{\mathrm{T}} \dot{f}(x_1, t) + 2(\partial_{x_1} f(y_1, t))^{\mathrm{T}} \dot{f}(y_1, t) \\ &= 2(\partial_{y_1} \dot{f}(x_1, t))^{\mathrm{T}} f(x_1, t) + 2(\partial_{x_1} \dot{f}(y_1, t))^{\mathrm{T}} f(y_1, t).\end{aligned}$$

Thus, $\frac{\mathrm{d}}{\mathrm{d}t}(\Delta f, f) = 2(\Delta f, \dot{f}) = 2(\Delta \dot{f}, f)$. ■

Throughout this paper Hadamard (elementwise) product and division of vectors u and v are represented as $u \circ v$ and $\frac{[u]}{[v]}$ respectively. A scalar k operating with a vector is represented as $[k]$, where $[k]$ is a vector having all entries as k. The dimension of $[k]$ is assumed to admit the well-defined elementwise operations.

3. Embedding of heterogeneous networks into MRAC under agent parameter uncertainty

3.1. Physical network

In designing control laws for the multiagent network associated with a graph $G = \{\mathcal{N}, \mathcal{E}\}$ in terms of $\mathbb{R}^n$-valued functions on $\mathcal{N}$, the space $\mathbb{R}^n$ may be invoked to represent agent positions or velocities. We focus on networks of agents moving physically in n-dimensional space, with the fixed communication structure specified by $\mathcal{E} = \mathcal{E}_o$, and the dynamics of the agent corresponding to each node x modeled by the equations

$$\begin{aligned}\dot{p}(x, t) &= v(x, t), \\ m(x)\, \dot{v}(x, t) &= f(x, t),\end{aligned} \tag{6}$$

where the n-vectors p, v, and f represent position, velocity, and force, respectively, and scalar $m(x)$ represents mass. The network may be heterogeneous in that $m(x) \neq m(y)$ for some $x, y \in \mathcal{N}$, and the masses of all agents are assumed to be unknown. The embedding of (6) into MRAC framework is carried out through specifying plant models relevant to the agents' collective behavior tasks, while retaining the network communication structure. For certain tasks – such as coordinated maneuvering – an agent might be required to track the reference velocity trajectory, calling for the single integrator model corresponding to the second line in (6). For other tasks – such as data gathering or network relocation [3] – tracking of the reference position trajectory is of interest, calling for the use of the double integrator model corresponding to the full system (6). These two cases are treated in Sections 4.1 and 4.2, respectively.

3.2. Plant and reference models

3.2.1. Plant models

To introduce the plant models, we follow the definition of a PdE as presented in [6]. Let $u(x,t) : \mathcal{N} \times \mathbb{R}^+ \mapsto \mathbb{R}^n$ be a function of two independent variables and represent the plant model variable of interest. Focusing first on the agent velocity as the primary object of control, we associate the plant model variable $u \in \mathbb{R}^n$ with the agent model variable v in (6), and introduce the equation

$$\dot{u}(x,t) = a(x)f(x,t), \qquad u(x,0) = \tilde{u}(x), \tag{7}$$

that mimics the second line of Eq. (6). Eq. (7) defines a continuous time PdE on graph with initial conditions $\tilde{u}(x)$, where $f(x,t)$ is the control input and $a(x)$ is an unknown positive constant for each $x \in \mathcal{N}$. To complete the plant model, we endow (7) with the communication structure $\mathcal{E}_o$ of the original network. The resulting representation, owing to its single time derivative form, will be henceforth referred to as the *single integrator network*. In analogy to spatiotemporal evolution of a PDE represented by $u(x,t)$ when x is a continuous variable, we will refer to $u(x,t)$ introduced above as representing a *spatiotemporal evolution* of a PdE.

Similarly, focusing on the agent position as the primary object of control and associating the plant model variable u with the agent model variable p in (6), we introduce the *double integrator network*

$$\ddot{u}(x,t) = a(x)f(x,t), \qquad u(x,0) = \tilde{u}_1(x), \qquad \dot{u}(x,0) = \tilde{u}_2(x) \tag{8}$$

where $f(x,t)$ is the control input and $a(x)$ is an unknown positive constant with known lower bound $\mathcal{A} \le a(x)$ for each $x \in \mathcal{N}$.

The orders of time derivatives in the plant models (7) and (8) as well as decentralized (local) sensing and actuation in both of them are now seen to closely resemble those of the parabolic and hyperbolic PDEs, respectively, under distributed sensing and actuation. This suggests that MRAC laws for uncertain parabolic and hyperbolic PDEs introduced in [9], if properly extended to graphs, could be applied to (7) and (8). This extension is accomplished in Sections 4.1 and 4.2, with the algorithms developed paralleling the MRAC laws of [9].

3.2.2. Reference models, motion assignment, and reference inputs

MRAC laws typically assume that once a *reference model* is selected, a *reference input* is specified that generates the desired reference model behavior. In the context of plant models (7) and (8) the latter can be thought of as the behavior of the *reference agents*. In the present work, instead, the desired spatiotemporal evolution is specified, and only for part of the agents. The latter agents are further referred to as the *leading reference agents*, while the rest — as the *trailing reference agents*. The latter, then, are governed by the interplay between the reference model structure and the leading reference agents' evolution. The motion of the latter agents is, then, generated through the interplay of the leading reference agents' evolution, the network communication structure, and the reference model dynamics. These features give rise to an unconventional reference generation structure. The assigned evolution profiles are assumed throughout this work to be sufficiently smooth to guarantee continuity of all the functions involved.

With this setting at hand, the embedding of plant models (7) and (8) into MRAC laws of [9] is carried out by specifying the reference models in the form of PdEs that represent the desired network behavior, endowing these models with the communication structure of the corresponding plant models, selecting the leading reference agents and assigning the desired evolution to them to address specific tasks, and calculating the corresponding reference inputs that induce their assigned evolution. These reference inputs are then applied to the reference models, yielding the reference models with the explicit leading agents' motion assignment. The goal of a MRAC law is to force the plant to follow the resulting reference model evolution under plant parameter uncertainty, while shaping the closed-loop dynamics to conform to those of the reference model. Let $w(x,t) : \mathcal{N} \times \mathbb{R}^+ \mapsto \mathbb{R}^n$ represent the reference model variable of interest. The physical units in which variable $w \in \mathbb{R}^n$ is expressed depend on the application of interest and are identical to those of the corresponding plant model variable $u \in \mathbb{R}^n$.

3.2.2.1. Single integrator network. To clearly bring out the solution behavior of a PDE indicated in the Introduction that could guide the collective behavior of coordinated agents in a network in a decentralized context, consider the parabolic equation governing the distribution of temperature in the rod under Dirichlet boundary control [5,10]:

$$\begin{aligned}
&\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2},\\
&T(0,t) = T(1,t) = \begin{Bmatrix} 1, & 0 \le t < 10, \\ t-9, & 10 \le t < 11, \\ 2, & t \ge 11, \end{Bmatrix},\\
&T(x,0) = 0.
\end{aligned} \tag{9}$$

Fig. 1 shows the evolution $T(x,t)$ of the rod temperature governed by (9). Although (9) specifies only the boundary temperature increase, as a consequence of heat diffusion the temperature throughout the entire spatial domain of the

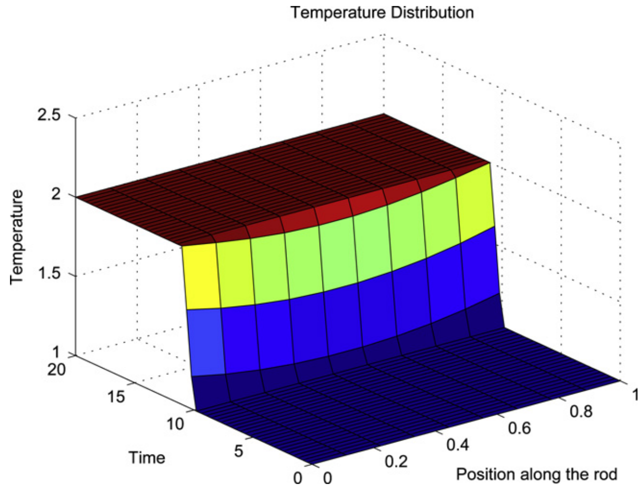

Fig. 1. Evolution of temperature profile in a rod with the preassigned boundary motion.

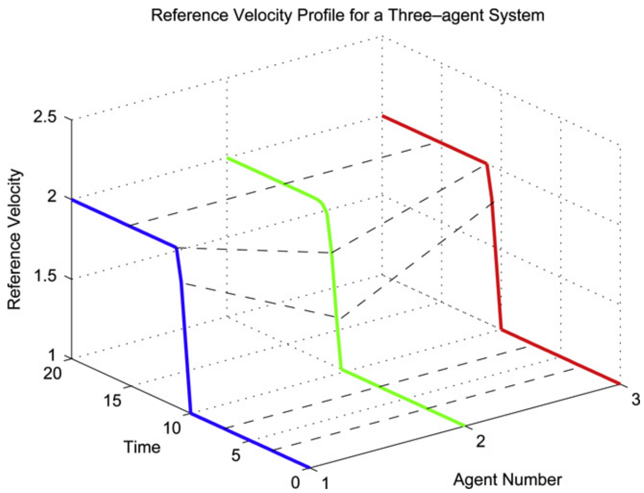

Fig. 2. Reference velocity profile for a one-dimensional three-agent system.

rod is seen to increase. This solution behavior can be interpreted as "casting a net" by specifying initial data and then "dragging" this net forward in time by the boundary motion assignment, or "herding" the intermediate points of the solution by its boundary points, to satisfy an objective of interest.

The qualitative analog of (9) is, then, a single integrator network represented by the reference model

$$\dot{w}(x,t) = \Delta w + r(x,t) \tag{10}$$

and the given *spatiotemporal plant velocity evolution profile* $w_d(x,t)$, further simply referred to as the *desired velocity profile*, specified only for boundary nodes and induced by input $r(x,t)$ applied only to these nodes. Further on, in analogy with (9), motion of the intermediate nodes is specified implicitly only through the diffusion-like term Δw, with reference input $r(x,t)$ for these nodes being absent, or equivalently, set to zero. The extension of the definition of $w_d(x,t)$ to the entire domain is obtained by simply setting $w_d(x,t) = w(x,t)$ for the trailing nodes, implying that for the nodes with unspecified $w_d(x,t)$, the desired profile is implicitly defined as that given by the solution of the reference model. Then, assuming that (10) executes the desired motion, i.e., $w(x,t) = w_d(x,t)$ for the entire domain, $r(x,t)$ can be expressed as

$$\begin{aligned} r(x,t) &= \dot{w}_d - \Delta w_d \quad \text{for the leading nodes;} \\ r(x,t) &= 0 \quad \text{for the trailing nodes.} \end{aligned} \tag{11}$$

Eq. (11) defines $r(x,t)$ for the leading nodes implicitly due to the presence of the undefined velocity profiles $w(x,t)$ of the trailing nodes adjacent to the leading ones. Substituting (11) into (10) and setting $w(x,0) = w_d(x,0)$ for the leading nodes yield

$$\begin{aligned} \dot{w} &= \Delta(w - w_d) + \dot{w}_d, \qquad w(x,0) = w_d(x,0) \quad \text{for the leading nodes;} \\ \dot{w}(x,t) &= \Delta w, \qquad w(x,0) = \tilde{w}(x) \quad \text{for the trailing nodes.} \end{aligned} \tag{12}$$

Eq. (12) can now be solved, yielding the reference velocity profile $w(x,t)$, along with the explicit value of $r(x,t)$ given by

$$r(x,t) = \dot{w} - \Delta w \quad \text{for the leading nodes;} \tag{13}$$
$$r(x,t) = 0 \quad \text{for the trailing nodes.}$$

As shown in Fig. 2, the behavior of the reference model (12) is the exact qualitative copy of that of (9), with boundary nodes herding the intermediate nodes. The motion of the latter nodes is, thus, driven purely by the coupling terms that represent communication of these nodes with the adjacent ones. Generalizing the case of boundary profile assignment, $w_d(x,t)$ can thus be assigned to any set of nodes on the graph to herd the intermediate nodes.

To illustrate the reference model/reference input generation, let subscripts u and r denote, respectively, the variables associated with the plant and the reference model, and consider a single integrator three-agent network

$$\dot{v}_{u1} = f_1, \qquad \dot{v}_{u2} = 2f_2, \qquad \dot{v}_{u3} = 4f_3, \tag{14}$$

where $v_{ui} \in \mathbb{R}$ is the single-component velocity of the ith agent, and the network information structure is given by the direct communication between agents 1 and 2 and between agents 2 and 3. The corresponding reference model of form (10) is then given by

$$\begin{aligned}\dot{v}_{r1} &= v_{r2} - v_{r1} + r_1,\\ \dot{v}_{r2} &= v_{r3} - 2v_{r2} + v_{r1} + r_2,\\ \dot{v}_{r3} &= v_{r2} - v_{r3} + r_3.\end{aligned} \tag{15}$$

Let the desired velocity profile be given by the boundary conditions of (9):

$$v_{d1} = \left\{\begin{matrix} 1, & 0 \le t < 10, \\ t-9, & 10 \le t < 11, \\ 2, & t \ge 11, \end{matrix}\right\}, \qquad v_{d3} = \left\{\begin{matrix} 1, & 0 \le t < 10, \\ t-9, & 10 \le t < 11, \\ 2, & t \ge 11, \end{matrix}\right\}, \tag{16}$$

where $v_{di} \in \mathbb{R}$, $i = 1, 3$, represents the desired velocity for the ith agent, and for the second agent no desired velocity profile is specified. The corresponding reference input computed using (11) takes the form

$$\begin{aligned}r_1 &= \dot{v}_{d1} + v_{d1} - v_{r2},\\ r_2 &= 0,\\ r_3 &= \dot{v}_{d3} + v_{d3} - v_{r2}.\end{aligned} \tag{17}$$

Substituting (17) into (15) yields the reference model of form (12) with decentralized velocity profile assignment given by

$$\begin{aligned}\dot{v}_{r1} &= -v_{r1} + \dot{v}_{d1} + v_{d1}, \qquad v_{r1}(0) = v_{d1}(0),\\ \dot{v}_{r2} &= v_{r3} - 2v_{r2} + v_{r1},\\ \dot{v}_{r3} &= -v_{r3} + \dot{v}_{d3} + v_{d3}, \qquad v_{r3}(0) = v_{d3}(0),\end{aligned} \tag{18}$$

where the reference input for the second agent is simply dropped and the velocity of this agent is seen to be determined purely by the plant (14) communication structure and the form of the reference model (15). Fig. 2 shows the reference velocity profile generated by the reference model (18). As is seen, this profile is the exact qualitative copy of the evolution of (9), with two boundary nodes herding the intermediate node.

3.2.2.2. Double integrator network. The collective behavior of coordinated agents in a double integrator network can be similarly guided by the hyperbolic equation governing the displacement in a string with the specified boundary displacement given by

$$\begin{aligned}&\frac{\partial^2 X}{\partial t^2} = \frac{\partial^2 X}{\partial x^2} - \frac{\partial X}{\partial t},\\ &X(0,t) = X(1,t) = \left\{\begin{matrix} 1, & 0 \le t < 3, \\ 6(t-3)^5 - 15(t-3)^4 + 10(t-3)^3 + 1, & 3 \le t < 4, \\ 2, & t \ge 4, \end{matrix}\right\},\\ &X(x,0) = 1, \qquad \dot{X}(x,0) = 0.\end{aligned} \tag{19}$$

Fig. 3 shows the displacement of the string following the preassigned boundary motion relatively closely along the entire domain, thereby giving rise to the same herding behavior as in the case of (9).

The counterpart of the reference model (10) for a double integrator network (8) then takes the form

$$\ddot{w} = \Delta w - \dot{w} + r, \tag{20}$$

together with the *desired spatiotemporal plant position evolution profile* w_d, further referred to as the *desired position profile*, specified only for the leading nodes. Following the parabolic case above, $r(x,t)$ can be expressed as

$$r(x,t) = \ddot{w}_d + \dot{w}_d - \Delta w_d \quad \text{for the leading nodes;} \tag{21}$$
$$r(x,t) = 0 \quad \text{for the trailing nodes,}$$

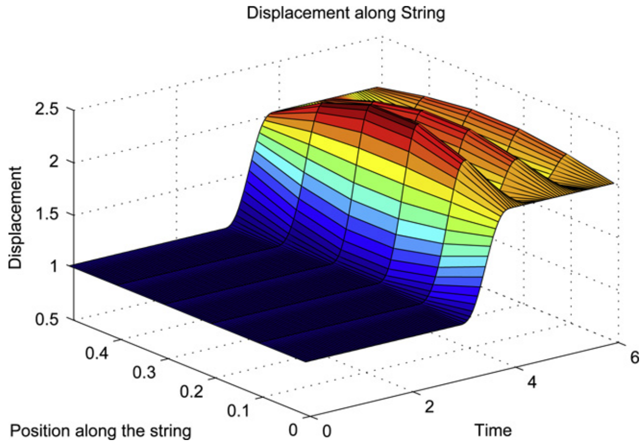

Fig. 3. Evolution of displacement profile in a string with the preassigned boundary motion.

and incorporation of the assigned position profile w_d into (20) yields

$$\begin{aligned} &\ddot{w} = \Delta(w - w_d) - (\dot{w} - \dot{w}_d) + \ddot{w}_d, \\ &w(x,0) = w_d(x,0), \qquad \dot{w}(x,0) = \dot{w}_d(x,0) \quad \text{for the leading nodes;} \\ &\ddot{w} = \Delta w - \dot{w}, \qquad w(x,0) = \tilde{w}_1(x), \\ &\dot{w}(x,0) = \tilde{w}_2(x) \quad \text{for the trailing nodes,} \end{aligned} \tag{22}$$

along with the explicit value of $r(x, t)$ given by

$$\begin{aligned} &r(x,t) = \ddot{w} + \dot{w} - \Delta w \text{ for the leading nodes;} \\ &r(x,t) = 0 \quad \text{for the trailing nodes.} \end{aligned} \tag{23}$$

Similarly to the single integrator case, the behavior of the reference model (22) driven by the assigned boundary nodes' motion is shown in Fig. 8 in Section 5.2 to be the exact qualitative copy of the behavior of (19), and the comments made above for (12) hold for (22) as well. The solution $w(x, t)$ of (22) will then serve as the *reference position profile*, and its projection onto $\mathbb{R}^n$ will be further referred to as the *reference position trajectory*.

4. MRAC law design

Once $w(x, t)$ and $r(x, t)$ are available, the development of the MRAC laws can proceed in a conventional manner by assuming that the reference models (10) and (20) are given along with the particular reference inputs that induce the desired reference velocity and position profiles, respectively. We define

$$\begin{aligned} &e(x,t) = w(x,t) - u(x,t), \qquad \eta_w^*(x) = \frac{1-a(x)}{a(x)}, \qquad \eta_r^*(x) = \frac{1}{a(x)}, \\ &\xi_w(x,t) = \eta_w(x,t) - \eta_w^*(x), \qquad \xi_r(x,t) = \eta_r(x,t) - \eta_r^*(x), \end{aligned} \tag{24}$$

where $\eta_w(x, t)$ and $\eta_r(x, t)$ are to be used as tunable control parameters; here $e(x, t)$ denotes the state error and $\xi_w(x, t)$ and $\xi_r(x, t)$ denote the parameter errors. We also note that $\dot{\eta}_w = \dot{\xi}_w$ and $\dot{\eta}_r = \dot{\xi}_r(x, t)$.

4.1. Single integrator networks

We now seek to design the control input f in (7), along with the adaptation law, such that $u(x, t)$ of (7) will follow $w(x, t)$ of (10) for a given r in spite of uncertainty in $a(x)$.

Proposition 1. *Assume that in the reference model (10) both r and Δw are continuous and bounded for all t. Following definition (24), for network (7) and the reference model (10) set the control law to be*

$$f = \Delta u + g \tag{25}$$

with adaptive part

$$g = ke + \eta_w \circ \Delta w + \eta_r \circ r, \tag{26}$$

where k is a positive constant, the coefficients η_w and η_r are adapted according to

$$\dot{\eta}_w = \varepsilon_w e \circ \Delta w, \quad \dot{\eta}_r = \varepsilon_r e \circ r, \tag{27}$$

ε_w and ε_r are positive constants. Then, $\|e\|$ tends to zero as $t \to \infty$.

Proof. The error $e(x, t)$ evolves according to

$$\begin{aligned} \dot{e} &= \dot{w} - \dot{u} = \Delta w + r - (a \circ \Delta u + a \circ g) \\ &= \Delta w - a \circ \Delta w + a \circ \Delta w - a \circ \Delta u + r - a \circ (ke + \eta_w \circ \Delta w + \eta_r \circ r) \\ &= a \circ \left(\frac{[1-a]}{[a]}\right) \circ \Delta w - a \circ \eta_w \circ \Delta w + a \circ \left(\frac{[1]}{[a]} - \eta_r\right) \circ r + a \circ \Delta e - ka \circ e \\ &= a \circ (\eta_w^* - \eta_w) \circ \Delta w + a \circ (\eta_r^* - \eta_r) \circ r + a \circ \Delta e - ka \circ e \\ &= a \circ (\Delta e - ke - \xi_w \circ \Delta w - \xi_r \circ r). \end{aligned} \tag{28}$$

Consider the Lyapunov function

$$V = \frac{1}{2}\left(\frac{[e]}{[a]}, e\right) + \frac{1}{2\varepsilon_w}(\xi_w, \xi_w) + \frac{1}{2\varepsilon_r}(\xi_r, \xi_r) \tag{29}$$

where $(\cdot, \cdot)$ denotes the inner product in $L^2(G)$. Clearly,

$$\dot{V} = \left(\frac{[\dot{e}]}{[a]}, e\right) + \frac{1}{\varepsilon_w}(\dot{\xi}_w, \xi_w) + \frac{1}{\varepsilon_r}(\dot{\xi}_r, \xi_r),$$

and it follows from (24) and (28) that

$$\begin{aligned} \dot{V} &= (\Delta e - ke - \xi_w \circ \Delta w - \xi_r \circ r, e) + (e \circ \Delta w, \xi_w) + (e \circ r, \xi_r) \\ &= (\Delta e, e) - k(e, e). \end{aligned} \tag{30}$$

From Lemma 1 it follows that $(-\Delta e, e) \geq 0$, so that $-(\Delta e, e) \geq 0$ and $(\Delta e, e) \leq 0$. Since $(e, e) \geq 0$ and k is positive, $k(e, e) - (\Delta e, e)$ is positive semi-definite. Further on, due to the assumption of Proposition 1, the error system (27) and (28) is a linear system with bounded time-varying coefficients. Hence all the conditions of the Lasalle/Yoshizawa theorem ([8], Thm. 8.4, p. 323) are satisfied globally and therefore $\|e\| \to 0$ as $t \to \infty$.

Remark 1. Boundedness of r does not limit the range of closed-loop system velocities, since, as seen in (13), r is computed though the derivatives of the assigned velocity profiles. Boundedness of Δw simply precludes assignment of velocity profiles in $\mathbb{R}^n$ that result in unbounded separation between agents. ■

Combining the results of this subsection with the single integrator network results of Section 3.2 yields MRAC schematics shown in Fig. 4.

Fig. 5 shows the computation structure for a case of n agents in which agent 1 and n are provided the reference inputs. The ith agent communicates only with the $(i-1)$th and the $(i+1)$th agents. Agents $\{2, \ldots, n-1\}$ move solely based on the inputs from their neighbors. As seen, all the computations required for obtaining the control signal are performed locally by each agent.

4.2. Double integrator networks

Similarly to the single integrator case, we now seek to design the control input f in (8), along with the adaptation law, such that $u(x, t)$ of (8) will follow $w(x, t)$ of (20) in spite of uncertainty in $a(x)$.

Proposition 2. *Assume that in the reference model (20) both r and Δw are continuous and bounded for all t. Following definition (24), for network (8) and the reference model (20) set the control law to be*

$$f = \Delta u - \dot{u} + g \tag{31}$$

with adaptive part

$$g = k_1 e + k_2 \dot{e} + \eta_w \circ \Delta w - \eta_w \circ \dot{w} + \eta_r \circ r, \tag{32}$$

where k_1 and k_2 are positive constants such that $k_2 + 1 > 1/\mathcal{A}$, the coefficients η_w and η_r are adapted according to

$$\dot{\eta}_w = \varepsilon_w (e + \dot{e}) \circ (\Delta w - \dot{w}), \qquad \dot{\eta}_r = \varepsilon_r (e + \dot{e}) \circ r, \tag{33}$$

ε_w and ε_r are positive constants. Then, $\|e\|$ will tend to zero as $t \to \infty$.

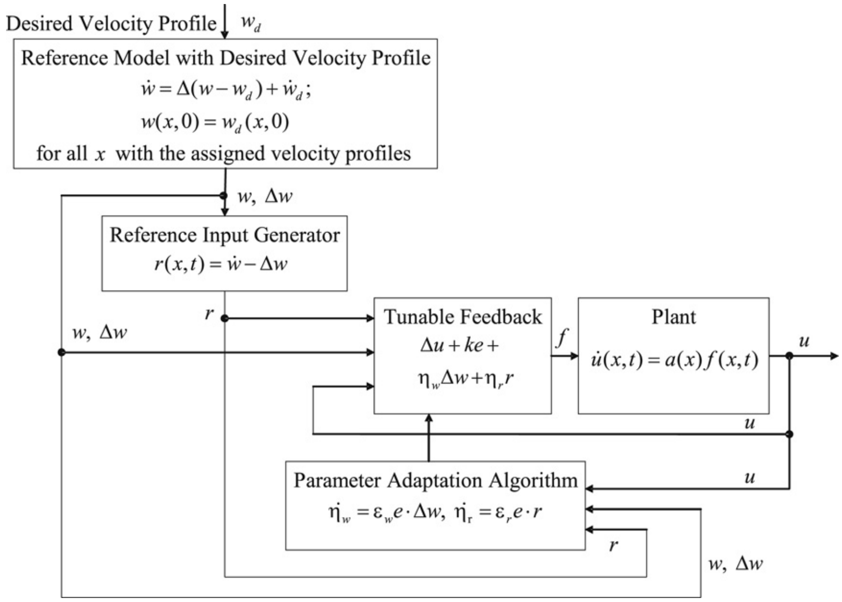

Fig. 4. Implementation schematics of the MRAC law for the single integrator network.

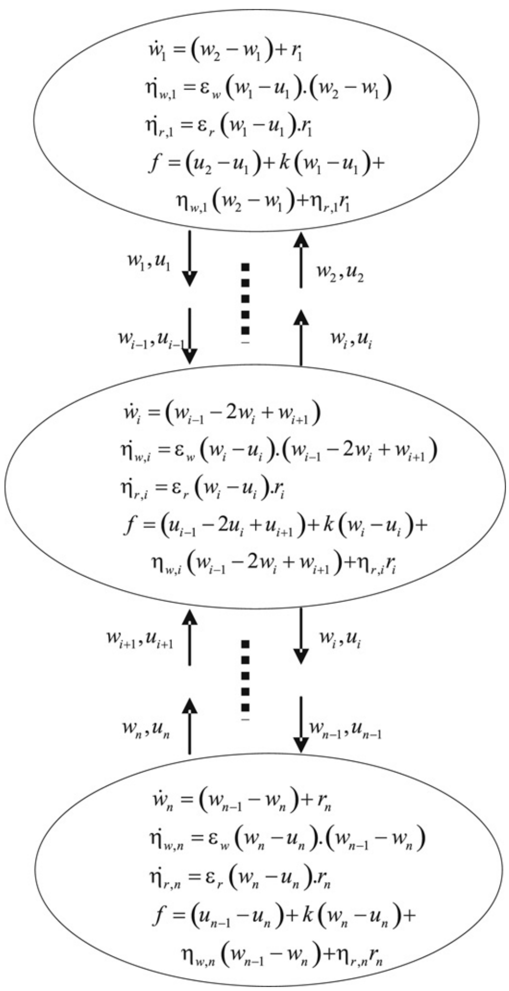

Fig. 5. Localized computation of the desired reference trajectory and the control signal.

Proof. The error $e(x,t)$ evolves according to

$$\begin{aligned}\ddot{e} &= \ddot{w} - \ddot{u} = \Delta w - \dot{w} + r - (a \circ \Delta u - a \circ \dot{u} + a \circ g) \\ &= \Delta w - a \circ \Delta w + a \circ \Delta w - \dot{w} + a \circ \dot{w} - a \circ \dot{w} - a \circ \Delta u + a \circ \dot{u} + r \\ &\quad - a \circ (k_1 e + k_2 \dot{e} + \eta_w \circ \Delta w - \eta_w \circ \dot{w} + \eta_r \circ r) \\ &= a \circ \left(\frac{[1-a]}{[a]}\right) \circ \Delta w - a \circ \left(\frac{[1-a]}{[a]}\right) \circ \dot{w} - a \circ \eta_w \circ \Delta w + a \circ \eta_w \circ \dot{w} \\ &\quad + a \circ \left(\frac{[1]}{[a]} - \eta_r\right) \circ r + a \circ (\Delta e - \dot{e} - k_1 e - k_2 \dot{e}) \\ &= a \circ ((\eta_w^* - \eta_w) \circ \Delta w - (\eta_w^* - \eta_w) \circ \dot{w} + (\eta_r^* - \eta_r) \circ r + (\Delta e - \dot{e} - k_1 e - k_2 \dot{e})) \\ &= a \circ (\Delta e - \dot{e} - k_1 e - k_2 \dot{e} - \xi_w \circ \Delta w + \xi_w \circ \dot{w} - \xi_r \circ r). \end{aligned} \tag{34}$$

Consider the Lyapunov function

$$\begin{aligned} V &= \frac{1}{2}\left(\frac{[1]}{[a]} \circ (\dot{e} + e), \dot{e} + e\right) + \frac{k_1}{2}(e, e) + \frac{1}{2}\left(\left(k_2 - \frac{[1]}{[a]} + 1\right) e, e\right) + \frac{1}{2}(-\Delta e, e) \\ &\quad + \frac{1}{2\varepsilon_w}(\xi_w, \xi_w) + \frac{1}{2\varepsilon_r}(\xi_r, \xi_r). \end{aligned}$$

Note that $((k_2 - [1]/[a] + 1)e, e) \geq 0$, since $k_2 + 1 > 1/\mathcal{A}$ and $(-\Delta e, e) \geq 0$ according to Lemma 1. It follows from Lemma 2 that

$$\dot{V} = \left(\frac{1}{a}(\ddot{e} + \dot{e}), \dot{e} + e\right) + k_1(\dot{e}, e) + \left(\left(k_2 - \frac{[1]}{[a]} + 1\right)\dot{e}, e\right) - (\Delta e, \dot{e}) + \frac{1}{\varepsilon_w}(\dot{\xi}_w, \xi_w) + \frac{1}{\varepsilon_r}(\dot{\xi}_r, \xi_r).$$

Further on, it follows from (24) and (34) that

$$\begin{aligned} \dot{V} &= \left(\Delta e - k_1 e - \left(k_2 - \frac{[1]}{[a]} + 1\right)\dot{e} - \xi_w \circ \Delta w + \xi_w \circ \dot{w} - \xi_r \circ r, \dot{e} + e\right) \\ &\quad + k_1(\dot{e}, e) + \left(\left(k_2 - \frac{[1]}{[a]} + 1\right)\dot{e}, e\right) - (\Delta e, \dot{e}) + ((e + \dot{e}) \circ (\Delta w - \dot{w}), \xi_w) + ((e + \dot{e}) \circ r, \xi_r) \\ &= (\Delta e, e) + (\Delta \dot{e}, \dot{e}) - k_1(e, e) - \left(\left(k_2 - \frac{[1]}{[a]} + 1\right)\dot{e}, \dot{e}\right). \end{aligned}$$

Thus, $\dot{V} < 0$ unless $(e, e) = 0$. As in the proof of Proposition 1 the error system (33) and (34) is a linear system with bounded time-varying coefficients. Hence all the conditions of the Lasalle/Yoshizawa theorem ([8], Thm. 8.4, p. 323) are satisfied globally and therefore $\|e\| \to 0$ as $t \to \infty$. ■

With velocity profiles replaced by the position profiles, Remark 1 holds for Proposition 2, as well. Combining the results of this subsection with the double integrator network results of Section 3.2 yields MRAC schematics shown in Fig. 6.

5. Simulations

This section applies the control laws given in Figs. 4 and 6 to networks consisting of three agents. In the reference models considered, only the boundary agents have non-zero reference input, so that the evolution of the middle reference agent is induced by the communication structure. The nature of the latter motion is shown to resemble evolution of parabolic and hyperbolic systems in the single integrator and the double integrator cases, respectively. For simplicity, the variable of interest $z(x,t) \in \mathbb{R}^n$, $x = 1, 2, 3$, corresponding to the xth agent, is denoted by $z_i(t)$, $i \equiv x$. To enhance clarity, notation v and p of the original physical system are retained to denote agent velocity and position, respectively, for both the plant evolution and the reference profiles. To distinguish the latter from one another, subscripts u and r are utilized, respectively. Finally, subscripts x and y are used to indicate the x and y components of the plant and the reference profile variables of interest. An important feature inherited from [9] by the control laws in Figs. 4 and 6 is that their parameter adaptation does not use Δu, and hence is not directly affected by any non-idealities in communication between agents in forming Δu. This makes it less sensitive to communication noise, since no non-ideality is involved in forming Δw_r.

5.1. MRAC of a single integrator network: Parabolic evolution

In this section, the similarity between the evolution of the reference model and that of a parabolic equation is shown, and the satisfactory performance of the MRAC law for a three-agent single integrator network is demonstrated.

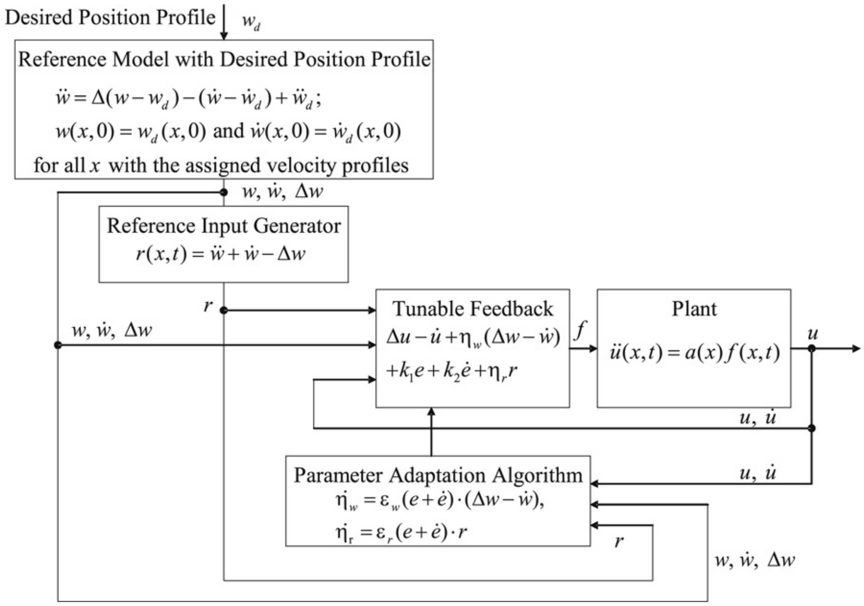

Fig. 6. Implementation schematics of the MRAC law for the double integrator network.

Consider a velocity coordination problem in which it is desired that the velocities of three agents moving in a plane and comprising the single integrator network follow the desired velocity profile. Such problems arise, for example, in the context of consensus forming [2] and surveillance. Let the network be governed by

$$\dot{v}_{u1} = f_1, \qquad \dot{v}_{u2} = 2f_2, \qquad \dot{v}_{u3} = 4f_3, \tag{35}$$

where $v_{ui} = [v_{uix}(t) \quad v_{uiy}(t)]^{\mathrm{T}} \in \mathbb{R}^2$ is the two-component velocity of the ith agent, and the network information structure be given by the direct communication between agents 1 and 2 and between agents 2 and 3. Hence the reference model (10) takes the form

$$\begin{aligned} \dot{v}_{r1} &= v_{r2} - v_{r1} + r_1, \\ \dot{v}_{r2} &= v_{r3} - 2v_{r2} + v_{r1} + r_2, \\ \dot{v}_{r3} &= v_{r2} - v_{r3} + r_3. \end{aligned} \tag{36}$$

Let the desired velocity profile be given by

$$v_{d1} = \begin{bmatrix} \begin{Bmatrix} 0, & 0 \le t < 10, \\ t - 10, & 10 \le t < 11, \\ 1, & t \ge 11, \end{Bmatrix} \\ 0 \end{bmatrix}, \qquad v_{d3} = \begin{bmatrix} \begin{Bmatrix} 0, & 0 \le t < 10, \\ t - 10, & 10 \le t < 11, \\ 1, & t \ge 11 \end{Bmatrix} \\ 1 \end{bmatrix}, \tag{37}$$

where $v_{di} = [v_{dix}(t) v_{diy}(t)]^{\mathrm{T}} \in \mathbb{R}^2, i = 1, 3$, represents the desired velocity for the ith agent. The corresponding reference input computed using (11) is given by

$$\begin{aligned} r_1 &= \begin{bmatrix} \dot{v}_{d1x} + v_{d1x} - v_{r2x} \\ \dot{v}_{d1y} + v_{d1y} - v_{r2y} \end{bmatrix}, \qquad r_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ r_3 &= \begin{bmatrix} \dot{v}_{d3x} + v_{d3x} - v_{r2x} \\ \dot{v}_{d3y} + v_{d3y} - v_{r2y} \end{bmatrix}. \end{aligned} \tag{38}$$

Substituting these r_1 and r_3 into (36) yields reference model of form (12) with decentralized velocity profile assignment given by

$$\begin{aligned} \dot{v}_{r1} &= -v_{r1} + \dot{v}_{d1} + v_{d1}, \\ \dot{v}_{r2} &= v_{r3} - 2v_{r2} + v_{r1}, \\ \dot{v}_{r3} &= -v_{r3} + \dot{v}_{d3} + v_{d3}, \end{aligned} \tag{39}$$

where for the second agent no desired velocity profile is specified and the reference input is 0. The velocity of this agent is determined purely by the choice of communication structure used in Eq. (36). Setting $v_{ri}(t = 0) = v_{di}(t = 0), i = 1, 3$, ensures that $v_{ri}(t) = v_{di}(t), i = 1, 3$, i.e. the exact following of the assigned velocity profile by agents 1 and 3. Fig. 7(a)

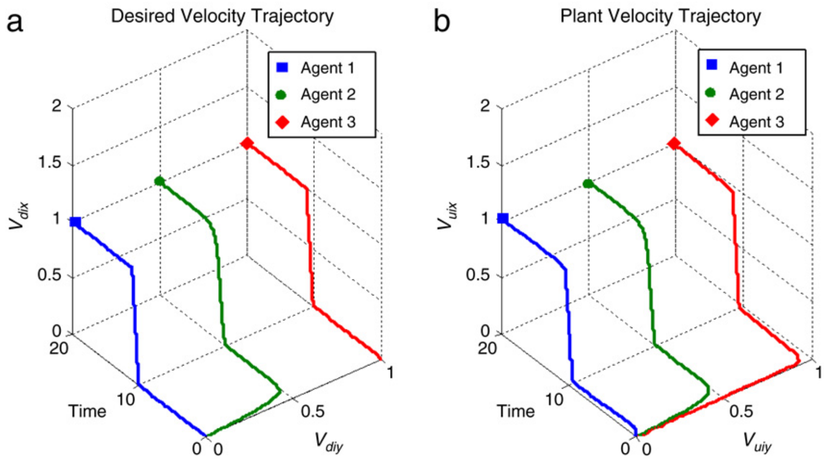

Fig. 7. The reference velocity profile (graph (a)) and the corresponding single integrator plant velocity profile (graph (b)) attained under the adaptive control law.

depicts the resulting reference velocity profile. The reference inputs r_1 and r_3 in (38) used in the control law (25)–(27) are calculated using v_{r2} determined from solving (39).

The adaptive control law defined in Proposition 1 takes the form

$$\begin{aligned} f_1 &= (v_{u2} - v_{u1}) + k(v_{d1} - v_{u1}) + \eta_{w,1} \circ (v_{r2} - v_{d1}) + \eta_{r,1} \circ r_1, \\ f_2 &= (v_{u3} - v_{u2}) + (v_{u1} - v_{u2}) + k(v_{r2} - v_{u2}) + \eta_{w,2} \circ ((v_{d3} - v_{r2}) + (v_{d1} - v_{r2})), \\ f_3 &= (v_{u2} - v_{u3}) + \eta_{w,3} \circ (v_{r2} - v_{d3}) + k(v_{d3} - v_{u3}) + \eta_{r,3} \circ r_3 \end{aligned} \quad (40)$$

with

$$\begin{aligned} \dot{\eta}_{w,1} &= \varepsilon_w (v_{d1} - v_{u1}) \circ (v_{r2} - v_{d1}), \\ \dot{\eta}_{w,2} &= \varepsilon_w (v_{r2} - v_{u2}) \circ ((v_{d3} - v_{r2}) + (v_{d1} - v_{r2})), \\ \dot{\eta}_{w,3} &= \varepsilon_w (v_{d3} - v_{u3}) \circ (v_{r2} - v_{d3}), \\ \dot{\eta}_{r,1} &= \varepsilon_r (v_{d1} - v_{u1}) \circ r_1, \qquad \dot{\eta}_{r,3} = \varepsilon_r (v_{d3} - v_{u3}) \circ r_3. \end{aligned} \quad (41)$$

In both (40) and (41) terms r_2 and $\eta_{r,2} r_2$ are omitted.

Fig. 7(b) depicts the velocity profiles of the three agents comprising the controlled network in the v_u plane, with initial conditions

$$v_{u1}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad v_{u2}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad v_{u3}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

subject to the control law (40) and (41) with $\eta_{w,1}(0) = \eta_{w,2}(0) = \eta_{w,3}(0) = \eta_{r,1}(0) = \eta_{r,2}(0) = \eta_{r,3}(0) = 0$. The adaptively controlled single integrator network is seen to track the reference velocity profile asymptotically.

The evolution of the reference velocity of the second agent in Fig. 7(a) resembles the evolution of temperature at the midpoint of the rod in Fig. 1. This is to be expected since Eq. (36) for the reference model, with reference input provided only to agents 1 and 3, resembles the discretized version of Eq. (9).

5.2. MRAC of a double integrator network: Hyperbolic evolution

In this section, the similarity between the evolution of the reference model and that of a hyperbolic equation is shown, and the satisfactory performance of the MRAC law for a three-agent double integrator network under decentralized position profile assignment is demonstrated. Consider a three-agent network moving in a plane with the task of tracking the reference position profile. Let the network be governed by

$$\ddot{p}_{u1} = f_1, \qquad \ddot{p}_{u2} = 2f_2, \qquad \ddot{p}_{u3} = 4f_3, \quad (42)$$

where $p_{ui} = [p_{uix}(t) p_{uiy}(t)] \in \mathbb{R}^2$ is the two-component position of the ith agent, and have the same topology as that of the system considered in Section 5.1. The reference model (20) in this case takes the form

$$\begin{aligned} \ddot{p}_{r1} &= p_{r2} - p_{r1} - \dot{p}_{r1} + r_1, \\ \ddot{p}_{r2} &= p_{r3} - 2p_{r2} + p_{r1} - \dot{p}_{r2} + r_2, \\ \ddot{p}_{r3} &= p_{r2} - p_{r3} - \dot{p}_{r3} + r_3. \end{aligned} \quad (43)$$

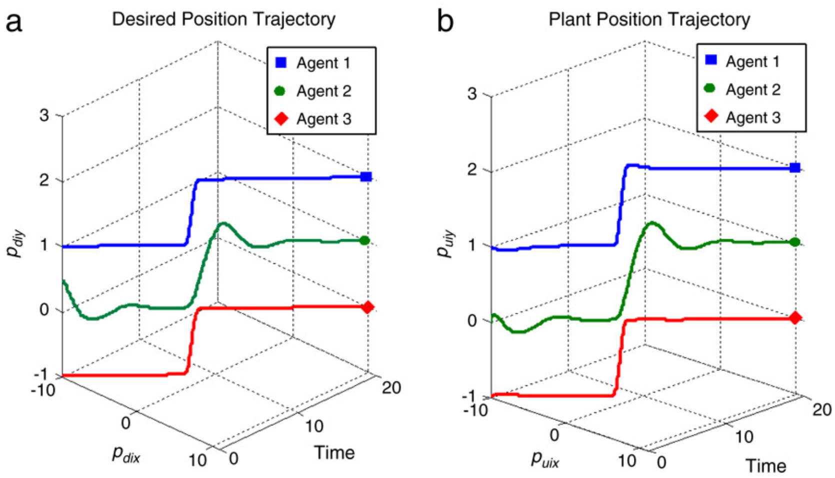

Fig. 8. The reference position profile (graph (a)) and the corresponding position profile of the double integrator plant (graph (b)) attained under the adaptive control law.

Let the desired position profile be given by

$$\begin{bmatrix} p_{d1x} \\ p_{d1y} \end{bmatrix} = \begin{bmatrix} \{-10+t, \quad t \geq 0,\} \\ \left\{ \begin{matrix} 1, & 0 \leq t < 8, \\ 6\,(t-8)^5 - 15\,(t-8)^4 + 10\,(t-8)^3 + 1, & 8 \leq t < 9, \\ 2, & t \geq 9, \end{matrix} \right\} \end{bmatrix},$$

$$\begin{bmatrix} p_{d3x} \\ p_{d3y} \end{bmatrix} = \begin{bmatrix} \{-10+t, \quad t \geq 0,\} \\ \left\{ \begin{matrix} -1, & 0 \leq t < 8, \\ 6\,(t-8)^5 - 15\,(t-8)^4 + 10\,(t-8)^3 - 1, & 8 \leq t < 9, \\ 0, & t \geq 9, \end{matrix} \right\} \end{bmatrix},$$

where $p_{di} = [p_{dix}(t)p_{diy}(t)] \in \mathbb{R}^2$ represents the two-component desired position of the ith agent. The corresponding reference input computed using (21) is given by

$$r_1 = \begin{bmatrix} \ddot{p}_{d1x} + \dot{p}_{d1x} + p_{d1x} - p_{r2x} \\ \ddot{p}_{d1y} + \dot{p}_{d1y} + p_{d1y} - p_{r2y} \end{bmatrix}, \qquad r_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$
$$r_3 = \begin{bmatrix} \ddot{p}_{d3x} + \dot{p}_{d3x} + p_{d3x} - p_{r2x} \\ \ddot{p}_{d3y} + \dot{p}_{d3y} + p_{d3y} - p_{r2y} \end{bmatrix}. \tag{44}$$

Substituting these r_1 and r_3 into (43) yields reference model of form (22) with decentralized position profile assignment given by

$$\begin{aligned} \ddot{p}_{r1} &= -p_{r1} - \dot{p}_{r1} + \ddot{p}_{d1} + \dot{p}_{d1} + p_{d1}, \\ \ddot{p}_{r2} &= p_{r3} - 2p_{r2} + p_{r1} - \dot{p}_{r2}, \\ \ddot{p}_{r3} &= -p_{r3} - \dot{p}_{r3} + \ddot{p}_{d3} + \dot{p}_{d3} + p_{d3}. \end{aligned} \tag{45}$$

Setting $v_{ri}(t=0) = v_{di}(t=0), \dot{v}_{ri}(t=0) = \dot{v}_{di}(t=0), i = 1, 3$, ensures that $v_{ri}(t) = v_{di}(t), i = 1, 3$. Note that for the second agent no desired position profile is specified and the reference input is 0. Its position is determined solely by the choice of communication structure used in Eq. (43). Fig. 8(a) depicts the resulting network reference position profile. The reference inputs r_1 and r_3 in (44) used in the control law (31)–(33) are calculated using p_{r2} determined from solving (45). Omitting the terms involving r_2 as a factor, the adaptive control law defined in Proposition 2 takes the form

$$\begin{aligned} f_1 &= (p_{u2} - p_{u1}) - \dot{p}_{u1} + k_1(p_{d1} - p_{u1}) + k_2(\dot{p}_{d1} - \dot{p}_{u1}) + \eta_{w,1} \circ (p_{r2} - p_{d1}) - \eta_{w,1} \circ \dot{p}_{d1} + \eta_{r,1} \circ r_1, \\ f_2 &= (p_{u3} - p_{u2}) + (p_{u1} - p_{u2}) - \dot{p}_{u2} + k_1(p_{r2} - p_{u2}) \\ &\quad + k_2(\dot{p}_{r2} - \dot{p}_{u2}) + \eta_{w,2} \circ ((p_{d3} - p_{r2}) + (p_{d1} - p_{r2})) - \eta_{w,2} \circ \dot{p}_{r2}, \\ f_3 &= (p_{u2} - p_{u3}) - \dot{p}_{u3} + k_1(p_{d3} - p_{u3}) + k_2(\dot{p}_{d3} - \dot{p}_{u3}) + \eta_{w,3} \circ (p_{r2} - p_{d3}) - \eta_{w,3} \circ \dot{p}_{d3} + \eta_{r,3} \circ r_3, \end{aligned} \tag{46}$$

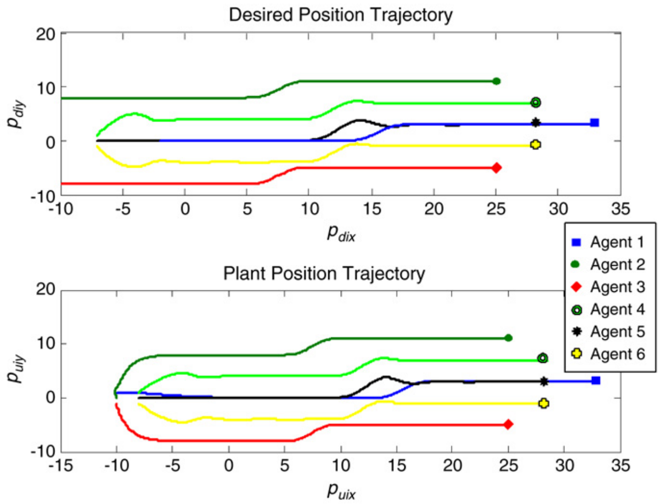

Fig. 9. Coordinated motion of six agents in which three of the agents herd the other three ones.

with

$$\begin{aligned}
\dot{\eta}_{w,1} &= \varepsilon_w((p_{d1} - p_{u1}) + (\dot{p}_{d1} - \dot{p}_{u1})) \circ (p_{r2} - p_{d1} - \dot{p}_{d1}),\\
\dot{\eta}_{w,2} &= \varepsilon_w((p_{r2} - p_{u2}) + (\dot{p}_{r2} - \dot{p}_{u2})) \circ ((p_{d3} - p_{r2}) + (p_{d1} - p_{r2}) - \dot{p}_{r2}),\\
\dot{\eta}_{w,3} &= \varepsilon_w((p_{d3} - p_{u3}) + (\dot{p}_{d3} - \dot{p}_{u3})) \circ (p_{r2} - p_{d3} - \dot{p}_{d3}),\\
\dot{\eta}_{r,1} &= \varepsilon_r((p_{d1} - p_{u1}) + (\dot{p}_{d1} - \dot{p}_{u1})) \circ r_1,\\
\dot{\eta}_{r,3} &= \varepsilon_r((p_{d3} - p_{u3}) + (\dot{p}_{d3} - \dot{p}_{u3})) \circ r_3.
\end{aligned} \tag{47}$$

Fig. 8(b) depicts the position profiles of the three agents comprising the controlled network in the p_u plane, with initial conditions

$$p_{u1}(0) = \begin{bmatrix} -10 \\ 1 \end{bmatrix}, \qquad p_{u2}(0) = \begin{bmatrix} -10 \\ 0 \end{bmatrix}, \qquad p_{u3}(0) = \begin{bmatrix} -10 \\ -1 \end{bmatrix}, \qquad \dot{p}_{ui}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad i = 1, 2, 3$$

subject to the control (46) and (47) with zero initial values for all control parameters. It is seen that the algorithm developed provides successful tracking under agent parameter uncertainty.

The evolution of the desired position for the second agent in Fig. 8(a) resembles the evolution of displacement at the midpoint of the string in Fig. 3. This is to be expected since the Eq. (43) for the reference model, with reference input provided only to agents 1 and 3, resembles the discretized version of Eq. (19) [11]. Hence a suitable communication structure can give rise to hyperbolic-type behavior.

5.3. Coordination in a double integrator network

To demonstrate generalization of the example above, consider a setting where several agents with the assigned trajectories herd intermediate agents. Note that since the motions in the *x* and *y* directions are independent, we can enforce parabolic- or hyperbolic-type behavior in both of these directions. Fig. 9 is a simulation of a double integrator six-agent network where agents 1, 2 and 3 have prescribed desired trajectories and form an isosceles triangle. Agent 4 communicates with agents 1 and 2, agent 6 communicates with agents 1 and 3 and agent 5 communicates with agents 4 and 6, neither of which has an assigned trajectory. Thus, we have 3 agents moving in a triangle and herding the other 3 agents in a systematic fashion through a particular communication structure. The latter governs the transients in the reference trajectories of the agents that are herded. For example, if agents 1 and 2 make sharp movements, then transients are introduced into the reference trajectory of agent 4. The magnitude of these transients can be reduced by tuning the reference dynamics – scaling up Δw and $\dot{w}$ that occur in the reference models – and making the corresponding changes in the control law. This is equivalent to increasing the stiffness of the string.

Clearly, if *n* agents have assigned trajectories, they can form a variety of convex structures and herd the rest of the agents within these structures in a coordinated fashion. Furthermore, varying graph topologies and the corresponding physical structures can be simply attained by switching in real time the communication structures of the agents being herded.

In the simulations the agents are assumed to move in a plane, but the developed MRAC laws are not limited to planar motion. Moreover, it is not required that the communication structure between the various agents be identical. Hence the developed MRAC laws very well apply to the case where the interaction between some of the agents is made stronger or weaker by means of attaching weights to the non-adaptive part of the control laws and making suitable changes in the

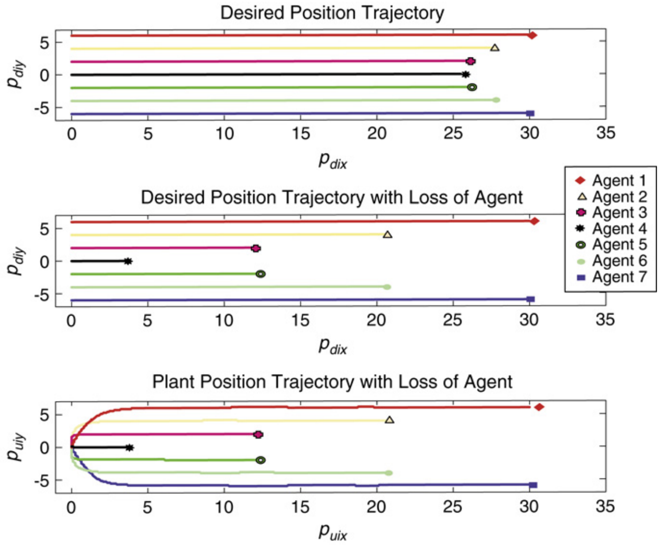

Fig. 10. Behavior of double integrator network under loss of agent.

reference model. Also note that, the definition of the Laplacian does not place any restriction on the number of neighbors that an agent can have and hence scenarios of one agent being herded by more than 2 agents are feasible.

5.4. Switched topology case — loss of agent in double integrator network

The behavior of the double integrator network under the sudden loss of agent is presented next. By loss of agent, we imply that the agent ceases to communicate with its neighbors and the neighbors use the last set of data that they received from the lost agent for all future computations. This scenario is depicted in Fig. 10 where agent 4 is presumed to be lost after a certain time and its position becomes constant. The reference trajectory is provided only to agents 1 and 7. The desired trajectory for the remaining agents is induced by the communication structure imposed on the reference model. The trajectory in the case of no agent loss is given by the upper graph in Fig. 10. But due to the localized nature of the computation (Fig. 5), the loss of an agent affects the calculation of both the desired position trajectory and the plant position trajectory. The new desired trajectory and the plant trajectory under the loss of an agent are shown in Fig. 10. In the present scenario the plant trajectory follows the new desired trajectory since the error in tracking for agent 4 was close to zero when the agent was lost. If this were not the case, the new desired position and plant position trajectory will deviate.

6. Summary

A well-posed globally stable PDE-based model reference adaptive control setting of [9] is extended to control of uncertain heterogeneous multiagent networks described by partial difference equations on graphs. The reference models with the decentralized motion assignment are introduced. The algorithms developed are augmented with the capability of tracking the desired profiles of the variables of interest. The setting proposed paves the way for considering control of uncertain multiagent networks under disturbances and switching topologies.

Acknowledgements

The present work was supported by the NSF grants CMS-0324630, ECS-0501407, and CMS-0449319. Shu Zhang is gratefully acknowledged for his help with the figures.

References

[1] A. Bensoussan, J.-L. Menaldi, Difference equations on weighted graphs, Journal of Convex Analysis 12 (1) (2005) 13–44. (Special issue in honor of Claude Lemaréchal).
[2] P.-A. Bliman, G. Ferrari-Trecate, Average consensus problems in networks of agents with delayed communications, in: Proc. 44th IEEE Conf. Decis. Contr., Seville, Spain, Dec. 10–13, 2005, pp. 7066–7071.
[3] G. Cao, G. Kesidis, T. LaPorta, B. Yao, Purposeful mobility in tactical sensor networks, in: S. Phoha, T. LaPorta, C. Griffin (Eds.), Sensor Network Operations, IEEE Press/Wiley-Interscience, Piscataway, NJ, 2006, pp. 113–126.
[4] J.B. Conway, A Course in Functional Analysis, second ed., Springer, New York, 1990.
[5] L.C. Evans, Partial Differential Equations, Am. Math. Soc., Providence, RI, 1998.

[6] G. Ferrari-Trecate, A. Buffa, M. Gati, Analysis of coordination in multiple agents formations through partial difference equations, N.5-PV, Instituto di Matematica Applicata e Tecnologie Informatiche, C.N.R., Pavia, Italy. http://www-rocq.inria.fr/who/Giancarlo.Ferrari-Trecate/FTBG04.htmlTech. Rep., 2004.
[7] L. Galbusera, M.P.E. Marciandi, P. Bolzern, G. Ferrari-Trecate, Control schemes based on the wave equation for consensus in multi-agent systems with double-integrator dynamics, in: Proc. 46th IEEE Conf. Decis. Contr., New Orleans, USA, Dec. 12–14, 2007, pp. 1498–1503.
[8] H.K. Khalil, Nonlinear Systems, 3rd ed., Prentice-Hall, Upper Saddle River, NJ, 2002.
[9] J.Y. Kim, J. Bentsman, Robust model reference adaptive control of parabolic and hyperbolic systems with spatially-varying parameters, in: Proc. 44th IEEE Conf. Decis. Contr., Seville, Spain, Dec. 10–13, 2005, pp. 1503–1508.
[10] O.A. Ladyzhenskaya, The Boundary Value Problems of Mathematical Physics, Springer-Verlag, New York, NY, 1985.
[11] K.W. Morton, D.F. Mayers, Numerical Solution of Partial Differential Equations, Cambridge University Press, 1994.
[12] C. Reynolds, Flocks, herds, and schools: A distributed behavioral model, Computer Graphics 21 (4) (1987) 25–34.
[13] S. Roy, A. Saberi, K. Herlugson, Formation and alignment of distributed sensing agents with double-integrator dynamics and actuator saturation, in: S. Phoha, T. LaPorta, C. Griffin (Eds.), Sensor Network Operations, IEEE Press/Wiley-Interscience, Piscataway, NJ, 2006, pp. 126–157.
[14] J.Y. Kim, Adaptive control of distributed parameter system, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 2006.