Self-Organizing Control Mechanism Based on Collective Decision-Making for Information Uncertainty

NAOMI KUZE and DAICHI KOMINAMI, Osaka University
KENJI KASHIMA, Kyoto University
TOMOAKI HASHIMOTO, Osaka Institute of Technology
MASAYUKI MURATA, Osaka University

Because of the rapid growth in the scale and complexity of information networks, self-organizing systems are increasingly being used to realize novel network control systems that are highly scalable, adaptable, and robust. However, the uncertainty of information (with regard to incompleteness, vagueness, and dynamics) in self-organizing systems makes it difficult for them to work appropriately in accordance with the network state. In this study, we apply a model of the collective decision-making of animal groups to enable self-organizing control mechanisms to adapt to information uncertainty. Specifically, we apply a mathematical model of collective decision-making that is known as the effective leadership model (ELM). In the ELM, informed individuals (those who are experienced or well-informed) take the role of leading the others. In contrast, uninformed individuals (those who perceive only local information) follow neighboring individuals. As a result of the collective behavior of informed/uninformed individuals, the animal group achieves consensus. We consider a self-organizing control mechanism using potential-based routing with an optimal control, and propose a mechanism for determining a data-packet forwarding scheme based on the ELM. Through evaluation by simulation, we show that, in a situation in which the perceived information is incomplete and dynamic, nodes can forward data packets in accordance with the network state by applying the ELM.

Categories and Subject Descriptors: C.2.2 [**Computer-communication Networks**]: Network Protocols—*Routing Protocol*

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Self-organization, collective decision-making, potential-based routing, information uncertainty

ACM Reference format:
Naomi Kuze, Daichi Kominami, Kenji Kashima, Tomoaki Hashimoto, and Masayuki Murata. 2018. Self-Organizing Control Mechanism Based On Collective Decision-Making for Information Uncertainty. *ACM Trans. Auton. Adapt. Syst.* 13, 1, Article 7 (April 2018), 21 pages.
https://doi.org/10.1145/3183340

This research was supported by Grant-in-Aid for Young Scientists (Start-up) No. 16H06915 from the Japan Society for the Promotion of Science (JSPS), and was partially supported by Grant-in-Aid for Scientific Research (B) No. 26289130, and also from the JSPS.

Authors' addresses: N. Kuze and M. Murata, Graduate School of Information Science and Technology, Osaka University; emails: n-kuze@ist.osaka-u.ac.jp, murata@ist.osaka-u.ac.jp; D. Kominami, Graduate School of Economics, Osaka University; email: d-kominami@econ.osaka-u.ac.jp; K. Kashima, Graduate School of Informatics, Kyoto University; email: kashima@amp.i.kyoto-u.ac.jp; T. Hashimoto, Faculty of Engineering, Osaka Institute of Technology; email: tomoaki.hashimoto@oit.ac.jp.

https://doi.org/10.1145/3183340

1 INTRODUCTION

Self-organization is a promising approach for controlling complicated, large-scale networks. The components of a self-organizing system behave automatically and autonomously based on simple rules and local interactions among components. This leads to the emergence of global patterns or behavior at a macroscopic level (Dressler 2008; Müller-Schloer et al. 2011; Prokopenko 2014). The bottom-up mechanism leads to low communication and computational costs for such emergence. However, in practice there are some challenges to using self-organizing control systems in industrial and business systems.

In most self-organizing systems, components have access to only local information. Although this feature certainly lowers communication and computational costs, it also sometimes leads the system to a solution that is only locally optimal. Moreover, the information that is available to components tends to be uncertain (i.e., incomplete, vague, or dynamic) because of effects such as noise and fluctuation and because systems tend to change dynamically, so information that components have already collected can become outdated.

To address the issue of information uncertainty, we apply the *collective decision-making* of swarms (Zhang et al. 2008; Conradt 2011, 2013) to self-organizing control systems. In swarms of animals such as birds, fish, and insects, the ability and energy of an individual is limited; a single member of a swarm perceives only itself and its surrounding environment. However, convergence to a state in which all individuals make the same, correct decision is achieved through local interactions among individuals. In this study, we apply the *effective leadership model* (ELM) (Couzin et al. 2005; Conradt et al. 2009), in which there are two types of individuals: those who are *informed* and those who are *uninformed*. Informed individuals take leadership roles; using their superior knowledge and experience, they can make correct decisions in accordance with the states of the swarm and the surrounding environment. In contrast, uninformed individuals perceive only the states of neighboring individuals and duly follow them. Consequently, uninformed individuals follow informed individuals, and this leads all individuals to make the same, correct decision. With the ELM, it has been demonstrated that the fraction of leaders required for an identically correct decision to be made diminishes with the size of the swarm (Couzin et al. 2005, thereby indicating that the ELM is highly scalable. Moreover, it is worth noting that decision-making is achieved without individuals knowing which individuals are the informed ones.

We apply the ELM to an optimal control mechanism for self-organizing systems that we proposed in previous work (Kuze et al. 2016). In self-organizing control systems with this optimal control mechanism, an external controller monitors the system state via partial nodes known as *controlled nodes*, and provides control feedback to them for faster convergence of self-organization. In other words, controlled nodes collect a larger amount of information about the system than do the other nodes, and are informed by the external controller as to how they should behave, thereby accelerating convergence. We view the controlled nodes as *leader nodes* (corresponding to informed individuals in the ELM) that lead all nodes in the system to make the same, correct decision.

In this study, we consider potential-based routing with the optimal control mechanism for wireless sensor networks (WSNs). Potential-based routing is a self-organizing routing mechanism in which a gradient field (known as a potential field) is used to determine the forwarding in a self-organizing manner. The next-hop nodes of data packets are determined stochastically in accordance with the potential field. However, which data-packet forwarding scheme is appropriate depends on the state of the system. For example, data packets should not be forwarded based on the potential field if that field is changing, because data packets could be forwarded to incorrect nodes. To address this problem, we propose a mechanism for determining the data-packet

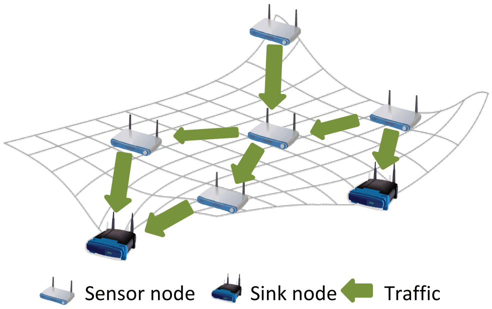


Fig. 1. Potential-based routing.

forwarding scheme based on the ELM. Note that although potential-based routing and WSNs are good examples of the application of the ELM, it can in fact be applied to many other mechanisms, situations, and environments.

The remainder of this article is organized as follows. In Section 2, we briefly explain potential-based routing and how it can be improved by using an optimal control mechanism. In Section 3, we propose and explain data-packet forwarding based on collective decision-making to deal with information uncertainty. We then conduct simulation experiments to demonstrate the advantages and properties of our proposed scheme. Finally, we give our conclusions and mention possible future work.

2 POTENTIAL-BASED ROUTING WITH AN OPTIMAL CONTROL MECHANISM

Potential-based routing is a self-organizing routing mechanism in which each node chooses a route by means of a hop-by-hop forwarding rule. Such mechanisms are actively used in the fields of WSNs, mobile ad-hoc networks, and information-centric networks (Kominami et al. 2013; Basu et al. 2003; Jung et al. 2009; Wu et al. 2008; Sheikhattar and Kalantari 2014; Eum et al. 2015; Lee et al. 2015). Here, we assume that potential-based routing is used in a WSN in which information gathering is infrequent and the capacity of each node is strictly limited.

In potential-based routing, each node has a scalar value called its *potential*, and data packets are forwarded to a neighbor whose potential is lower than that of the forwarder. In WSNs, data packets are generally sent to a sink node, and a smaller number of hops to the sink node is reflected in a lower potential value. The simple forwarding rule to “forward data to a neighboring node with a lower potential” can therefore result in data packets gathering at sink nodes, as illustrated in Figure 1. Potential-based routing is highly scalable because each node uses only local information to calculate potentials and uses a local rule to forward data. In Sections 2.1 and 2.3, we describe a method for constructing a potential field and show how to use it to select the next hop node.

2.1 Potential-Field Construction

Sheikhattar and Kalantari (2014) focused on the convergence of potential-based routing and enhanced the potential convergence speed. They proposed a potential-calculation method based on not only current potentials but also prior potentials to accelerate the convergence. The potential $\theta_n(t)$ of node n at time t is given by

$$\theta_n(t+1) = \theta_n(t) + \alpha(\theta_n(t) - \theta_n(t-1)) + \beta\sigma_n \left(\sum_{k \in \mathcal{N}(n)} \{\theta_k(t) - \theta_n(t)\} + f_n(t) \right). \tag{1}$$

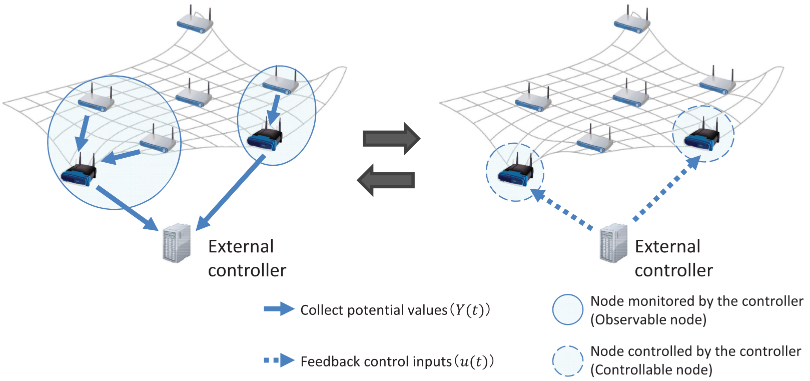

External controller
External controller
Collect potential values $(Y(t))$
Feedback control inputs $(u(t))$
Node monitored by the controller (Observable node)
Node controlled by the controller (Controllable node)

Fig. 2. Potential-based routing with controller feedback in which an external controller collects potential values from observable nodes and periodically provides control inputs to controllable nodes.

Here, $\mathcal{N}(n)$ is the set of the neighbors of node n, and α is a weighting parameter for the increase in potential from time $t-1$ to t when calculating the potential at $t+1$. Larger values of α mean that the amount by which the potential changes is more important and, therefore, the system becomes less subject to current noise but it converges more slowly. The parameter β determines the amount of influence exerted by the potentials of neighboring nodes. The node-dependent weighting σ_n is defined as $\sigma_0/|\mathcal{N}(n)|$ (where σ_0 is a parameter), and $f_n(t)$ corresponds to the flow rate of node n at time t. For sensor nodes, $f_n(t)$ is a negative value that indicates the data-generation rate; this rate is generally application dependent.

In contrast, for sink nodes, $f_n(t)$ is a positive value that determines the rate at which data packets are delivered to the node. The network manager can set the data-packet delivery rate to an arbitrary value. If the flow-conservation constraint is satisfied (i.e., $\sum_{n\in\{1,\ldots,N\}} f_n(t) = 0$), then a potential field is constructed such that the actual rates at which data packets are delivered to nodes satisfy the given flow rates (i.e., all gradients). Specifically, the potential differences between next-hop nodes correspond to the appropriate flow rates.

2.2 Potential-Field Construction with Optimal Control

We now describe our construction of a potential field with an associated optimal control mechanism using a method that we proposed in previous work (Kuze et al. 2016). The convergence of potentials based on Equation (1) is faster than that of simple Jacobi iterations (as used in our previous work (Kominami et al. 2013)), but it still takes a long time to converge because the calculation is based on local information only. We therefore introduce a *controller* to observe and estimate the network state (potential values) and to regulate the potentials of a partial set of nodes to achieve faster convergence.

The controller monitors network information, in particular the potential values of a partial set of nodes, which we call the *observable nodes*. The controller then returns suitable control inputs to a partial set of nodes, which we call the *controllable nodes*, to accelerate the convergence of the potential distribution toward the target potential distribution. We assume that the controller has direct connections with the controlled nodes to regulate their potentials. The controller collects the potential information of observable nodes via controlled nodes, as illustrated in Figure 2. The controller cannot directly access node potentials of non-observable nodes, but it can estimate them by utilizing a model of the potential dynamics, which describes potential changes based on local node interactions.

Note that information about the network topology and the flow rates of nodes is needed to design a controller and to calculate target potential values. Such information is difficult to estimate, and is reported to the controller only when it changes because we assume that the intervals over which the network topology and flow rates change are lower than the convergence time of the potentials. This assumption is plausible because potential convergence is generally achieved as a result of iterative behavior (nodes' potential updates and the controller's feedback) in potential-based routing with optimal feedback. This requires the frequencies of potential updates and controls to be much higher than those of changes in the network topology and flow rates.

2.2.1 Network Dynamics. Let the dynamics of potentials be given by a deterministic discrete-time model. Nodes interact locally with each other to update their potentials. With our proposed mechanism, the controller sends feedback inputs $\boldsymbol{u}(t) = [\eta_1(t) \ \cdots \ \eta_{N_{ctrl}}(t)]^T$ to N_{ctrl} controllable nodes to facilitate potential convergence. In this study, the update rule of each potential is the same as that in Sheikhattar and Kalantari (2014), except for the controllable nodes. Node n updates its potential at time t by

$$\theta_n(t+1) = (\alpha+1)\theta_n(t) - \alpha\theta_n(t-1) + \beta\sigma_n\left(\sum_{k\in\mathcal{N}_b(n)} \{\theta_k(t) - \theta_n(t)\} + \bar{f}_n\right) + \eta_n(t). \tag{2}$$

If node n is not controllable, then $\eta_n(t) = 0$. We set σ_n to a constant value σ $(0 < \sigma < 1)$ for all n $(\in \{1, \ldots, N\})$ because the original value of σ_n $(\sigma_0/|\mathcal{N}(n)|)$ proposed in Sheikhattar and Kalantari (2014) leads to oscillation of potentials in some situations.

Next, we describe the potential dynamics of the network. The potential values of N nodes in the network are described as a vector $\boldsymbol{\Theta}(t) = [\boldsymbol{\theta}_1(t) \ \cdots \ \boldsymbol{\theta}_N(t)]^T$ using $\boldsymbol{\theta}_n(t) = [\theta_n(t) \ \theta_n(t+1)]$. The potential dynamics of the network are given by Equation (3) using the flow matrix $\boldsymbol{F}^i$ and control inputs $\boldsymbol{u}^i$:

$$\boldsymbol{\Theta}(t+1) = \boldsymbol{A}\boldsymbol{\Theta}(t) + (\boldsymbol{E}\boldsymbol{u}(t) + \beta\sigma\boldsymbol{F}) \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{3}$$

where

$$\boldsymbol{A} = \boldsymbol{I}_{N\times N} \otimes \begin{bmatrix} 0 & 1 \\ -\alpha & \alpha+1 \end{bmatrix} - \boldsymbol{\Gamma} \otimes \begin{bmatrix} 0 & 0 \\ 0 & \beta\sigma \end{bmatrix}. \tag{4}$$

Matrix $\boldsymbol{I}_{N\times N}$ is the $N \times N$ identity matrix, and $\boldsymbol{\Gamma}$ corresponds to the graph Laplacian, which represents the network topology. The $(N \times N_{ctrl})$-matrix $\boldsymbol{E}$ specifies the controllable nodes: that is, element $e_{ij} \in \{0, 1\}$ of $\boldsymbol{E}$ is 1 if and only if node i receives the jth element of $\boldsymbol{u}(t)$ as control input $\eta_i(t)$.

Under these dynamics, the target potential distribution is given by a solution of

$$(\boldsymbol{I}_{2N\times 2N} - \boldsymbol{A})\bar{\boldsymbol{\Theta}} = \beta\sigma\boldsymbol{F} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{5}$$

The controller calculates the target potential distribution of the network to calculate the control inputs.

2.2.2 Optimal Controller Design. We now explain how the controller calculates the control feedback given to the controlled nodes. To calculate the control inputs $\boldsymbol{u}$, the controller monitors the potentials $\boldsymbol{Y}$ of the observable nodes in the network via the controllable nodes. The $(2N_{obs} \times 1)$-vector $\boldsymbol{Y}(t)$ is given by $\boldsymbol{Y}(t) = \boldsymbol{H}\boldsymbol{X}(t)$ using $\boldsymbol{X}(t) = \bar{\boldsymbol{\Theta}} - \boldsymbol{\Theta}(t)$, the gap between the current and target potential values. N_{obs} is the number of observable nodes in the network, and the $(2N_{obs} \times 2N)$-matrix $\boldsymbol{H}$ determines the observable nodes. The element $h(2n, 2m)$, $h(2n+1, 2m+1) \in \{0, 1\}$ of $\boldsymbol{H}$ is 1 if and only if the controller monitors the potential value of node m as the nth element of $\boldsymbol{Y}$.

The controller estimates $\boldsymbol{X}(t)$ from the observable information $\boldsymbol{Y}$, and then calculates the control inputs $\boldsymbol{u}$. The $(2N \times 1)$-vector $\tilde{\boldsymbol{X}}(t)$ is the estimation of $\boldsymbol{X}(t)$ by the controller, where $\tilde{\boldsymbol{X}}(t)$ and $\boldsymbol{u}(t)$ are given by

$$\tilde{\boldsymbol{X}}(t+1) = \boldsymbol{A}_c\tilde{\boldsymbol{X}}(t) + \boldsymbol{B}_c\boldsymbol{Y}(t), \tag{6}$$

$$\boldsymbol{u}(t) = \boldsymbol{C}_c\tilde{\boldsymbol{X}}(t) + \boldsymbol{D}_c\boldsymbol{Y}(t). \tag{7}$$

If $\tilde{\boldsymbol{X}}^i(t)$ is close to zero, then the potentials are estimated to be close to their target values. $\boldsymbol{A}_c$, $\boldsymbol{B}_c$, $\boldsymbol{C}_c$, and $\boldsymbol{D}_c$ are design parameters.

Concerning the performance criteria, let us define

$$\phi(k) = \boldsymbol{X}(k)^T\boldsymbol{X}(k) + r\boldsymbol{u}(k)^T\boldsymbol{u}(k)$$

as the stage cost, where r specifies the tradeoff between convergence speed and input energy. With a larger r, control inputs become smaller and the stability of the system is enhanced. Specifically, potentials change more gently; however, the convergence of the potentials is slower. Our design objective is then to minimize the worst-case error

$$\sup_{\boldsymbol{d}} \frac{\sum_{k=0}^{\infty} \phi(k)}{\sum_{k=0}^{\infty} \boldsymbol{d}(k)^T\boldsymbol{d}(k)}.$$

This min–max-type problem is called H^∞ optimization (Zhou et al. 1995).

With the estimation model described by Equations (6) and (7), which has $2N$ state variables, the optimal feedback $\boldsymbol{u}(t)$ is calculated with computational cost $O(N^2)$. To reduce the computational cost, the controller uses reduced-order models that have h $(<2N)$ state variables (Zhou et al. 1995; Antoulas et al. 2006) for which the computational cost is $O(h^2)$. The details are explained in previous work (Kuze et al. 2016).

2.3 Routing

A node with a data packet forwards it according to the potential values of itself and its neighbors. In our potential-based routing, when a sensor node generates or receives a data packet, it probabilistically selects a subsequent node that has a lower potential value than its own, and the packet eventually arrives at a sink node in this way. Specifically, a next-hop node is selected with a probability that is proportional to the difference of potential values: the probability $p_{n\rightarrow j}(t)$ that sensor node n selects neighbor node j as the next-hop node for a data packet at time t is given by

$$p_{n\rightarrow i}(t) = \begin{cases} \frac{\theta_n(t)-\theta_i(t)}{\sum_{k\in\mathcal{N}_l(n)}\{\theta_n(t)-\theta_k(t)\}}, & \text{if } i \in \mathcal{N}_l(n) \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

where $\mathcal{N}_l(n)$ is the neighbor node set of node n that are assigned lower potential values than node n. In other words, $\theta_n(t) - \theta_i(t) > 0$ for all i $(\in \mathcal{N}_l(n))$. If node n has no neighbor node with lower potential, that is, $|\mathcal{N}_l(n)| = 0$, then the data packet is not sent to any node and is dropped; however, this generally occurs only in transient cases, such as node failures or changes of potential values at the sink node.

3 DATA PACKET FORWARDING BASED ON COLLECTIVE DECISION-MAKING TO DEAL WITH INFORMATION UNCERTAINTY

We now propose a mechanism for determining a scheme for data-packet forwarding based on the ELM.

3.1 Overview

The ELM (Couzin et al. 2005; Conradt et al. 2009) is a mathematical model that describes collective decision-making in swarms. There are two types of individuals: *informed* ones and *uninformed* ones. Informed individuals that have a preferred direction lead uninformed individuals in that direction. In contrast, uninformed individuals perceive only the positions and velocities of their neighbors in order to follow them. As a result, all individuals in the group go in an accurate direction.

For potential-based routing with optimal control, we introduce the concept of *leader nodes* and *follower nodes*, corresponding to informed individuals and uninformed individuals, respectively. We consider controlled nodes to be leader nodes, and the others to be follower nodes. This is because the controller collects information about the system via controlled nodes so that controlled nodes have a larger amount of information than the others, which allows controlled nodes to determine which forwarding scheme should be used for data-packet forwarding. Leader nodes, whose role is to guide follower nodes, use the collected information to determine which forwarding scheme is preferred, and make their decision accordingly. In contrast, follower nodes decide which forwarding scheme to use in accordance with the decisions of their neighbors.

We explain the ELM briefly in Section 3.2. We then describe our scheme in Section 3.3.

3.2 Effective Leadership Model

Given a group of N individuals, individual n has position vector $\boldsymbol{c}_n(t)$ and velocity vector $\boldsymbol{v}_n(t)$ at time t. Individuals move at a distance from each other to avoid collisions. If there are individuals within distance α, individual n changes its direction to be farther from them. The desired direction $\boldsymbol{d}_n(t)$ of individual n at time t is updated by

$$\boldsymbol{d}_n(t+\Delta t) = -\sum_{i \in \mathcal{N}_b(n,\alpha)} \frac{\boldsymbol{c}_i(t) - \boldsymbol{c}_n(t)}{|\boldsymbol{c}_i(t) - \boldsymbol{c}_n(t)|}, \tag{9}$$

where $\mathcal{N}_b(n,\alpha)$ is the set of individuals within distance α from individual n. Otherwise, uninformed individual n determines its direction by following neighboring individuals, and updates its desired direction by

$$\boldsymbol{d}_n(t+\Delta t) = \sum_{i \in \mathcal{N}_b(n,\rho)} \frac{\boldsymbol{c}_i(t) - \boldsymbol{c}_n(t)}{|\boldsymbol{c}_i(t) - \boldsymbol{c}_n(t)|} + \sum_{i \in \mathcal{N}_b(n,\rho)} \frac{\boldsymbol{v}_n(t)}{|\boldsymbol{v}_n(t)|}, \tag{10}$$

where ρ corresponds to the range that individuals can perceive.

In contrast, informed individual n determines its desired direction $\boldsymbol{d}'_n(t)$ based not only on the local coordination but also on the preferred direction $\boldsymbol{g}_n$. Its desired direction is calculated by

$$\boldsymbol{d}'_n(t+\Delta t) = \frac{\hat{\boldsymbol{d}}_n(t+\Delta t) + \omega \boldsymbol{g}_n}{|\hat{\boldsymbol{d}}_n(t+\Delta t) + \omega \boldsymbol{g}_n|}, \tag{11}$$

where $\hat{\boldsymbol{d}}_n(t+\Delta t) = \boldsymbol{d}_n(t+\Delta t)/|\boldsymbol{d}_n(t+\Delta t)|$. The parameter ω (≥ 0) determines the weight of the preferred direction to the desired direction. The larger the value of ω, the more an informed individual attempts to go in its preferred direction. On the contrary, the smaller the value of ω, the more the individual is influenced by local coordination. In the original context, ω was regarded as the degree of *assertiveness* (Conradt et al. 2009).

3.3 Data-Packet Forwarding Mechanism Based on the Effective Leadership Model

We propose a mechanism for determining a data-packet forwarding scheme based on the ELM. In our proposal, a node decides based on the ELM which forwarding scheme it uses when transmitting data packets. Note that a node does not move in itself and we do not consider the mobility of nodes in this article.

A node n with a data packet selects forwarding scheme r_i from the set $\mathcal{R} = \{r_1, \ldots, r_M\}$ of forwarding schemes and uses it to forward the data packet. Node n has a decision vector $\boldsymbol{c}_n(t) = [c_n^1(t), \ldots, c_n^M(t)]$, and stochastically selects a forwarding scheme in accordance with $\boldsymbol{c}_n$. Element c_n^i is a real value, and its lower and upper bounds are c_{min}^i and c_{max}^i (i.e., $c_n^i \in [c_{min}^i, c_{max}^i]$ for all nodes n). The probability that node n selects scheme i at time t is given by

$$P_n^i(t) = \frac{c_n^i(t)}{\sum_{j \in \{1, \ldots, M\}} c_n^j(t)}. \tag{12}$$

The larger the value of c_n^i, the more likely node n is to select scheme i.

Decision vector $\boldsymbol{c}$ is updated based on the ELM. Follower node n updates its decision vector $\boldsymbol{c}_n$ with local coordination by

$$\boldsymbol{c}_n(t+1) = \sum_{n' \in \mathcal{N}_b(n)} \frac{\boldsymbol{c}_{n'}(t)}{|\mathcal{N}_b(n)|} + \boldsymbol{\delta}_n(t) \sum_{n' \in \mathcal{N}_b(n)} \frac{\boldsymbol{c}_{n'}(t) - \boldsymbol{c}_{n'}(t-1)}{|\mathcal{N}_b(n)|}, \tag{13}$$

where $\mathcal{N}_b(n)$ is the set of neighboring nodes of node n. Equation (13) corresponds to the direction update of uninformed individuals, as described in Equation (10). Vector $\boldsymbol{\delta}_n = \{\delta_n^1, \ldots, \delta_n^M\}$ is a parameter vector that determines the weight given to the change of $\boldsymbol{c}$ of the neighboring nodes. Element δ_n^i is given by

$$\delta_n^i(t) = \begin{cases} c_{max}^i - \sum_{n' \in \mathcal{N}_b(n)} \frac{c_{n'}^i(t)}{|\mathcal{N}_b(n)|}, \text{ if } \sum_{n' \in \mathcal{N}_b(n)} \frac{c_{n'}^i(t)}{|\mathcal{N}_b(n)|} \geq c_{max}^i/2 \\ \sum_{n' \in \mathcal{N}_b(n)} \frac{c_{n'}^i(t)}{|\mathcal{N}_b(n)|} - c_{min}^i, \text{ otherwise} \end{cases}. \tag{14}$$

In contrast, leader node n updates its decision vector $\boldsymbol{c}_n$ using both local coordination and its preferred decision vector $\boldsymbol{g}_n$ by

$$\boldsymbol{c}_n'(t) = (1 - \omega)\boldsymbol{c}_n(t) + \omega \boldsymbol{g}_n(t), \tag{15}$$

where $\omega \in [0, 1]$ is a parameter that determines the weight given to the preferred decision vector $\boldsymbol{g}$. The larger the value of ω, the more leader node n is influenced by its preferred decision vector.

4 PERFORMANCE EVALUATION

4.1 Overview

We conducted a computer simulation to demonstrate the advantages and properties of our proposed scheme. We first show in Section 4.3 that, in our proposed scheme based on ELM, nodes can select the proper forwarding scheme according to the network condition. Then, we investigate in detail the properties of our proposed scheme in Sections 4.4 and 4.5.

For the network simulator, we use an event-driven packet-level simulator written by us in Visual C++ that calls MATLAB functions *dlqr* to design an optimal central controller and sub-controllers with PBR-h-opt and PBR-h-opt-mr, *dhinflmi* to design an optimal external controller with PBR-opt-mr, and *balred* to obtain a reduced-order model with PBR-opt-mr and PBR-h-opt-mr on a 64-bit PC with a 2.70-GHz Intel Xeon CPU and 64.0GB of memory. In the MAC layer, each node sends information about its own potential to its neighbors for their potential updates using intermittent receiver-driven data transmission (IRDT) (Kominami et al. 2013), which is an asynchronous receiver-driven data transmission protocol. We use a disk model as a physical layer model in which

Table 1. Network Settings

Parameter	Value
Buffer size	1
Interval of ID packet emissions	0.5s
Potential update interval	50s
Control feedback interval	50s

data packets drop with 100% probability if they collide with each other. Because the capacity of each sensor node is limited in a WSN, we set the queue size of each sensor node to 1.

In the simulator, nodes are not synchronized. Nodes do not match their timing to receive feedback from the controller or to update their potentials. We set the interval of the control feedback by the controller, and that of potential updates in nodes, to be equal so that the controller can estimate the dynamics of the network with small errors.

4.2 Simulation Settings

In this evaluation, nodes select a scheme for data-packet forwarding from two types of scheme: potential-based forwarding and hop-based forwarding. Leader nodes choose a preferred scheme in accordance with the state of the potential field. If the potential value changes by only a small amount, the leader nodes assume that the potential field has already converged to the target potential distribution: that is, potential-based forwarding is preferred. Otherwise, if the potentials are changing by large amounts, the leader nodes assume that the potential field has not yet converged, in which case hop-base forwarding is preferred. As for the follower nodes, they select a scheme in accordance with local coordination.

The detailed implementation is as follows. The decision vector $\boldsymbol{c}$ of nodes is given by $[c_{potential}, c_{hop}]^T$. The proportion of nodes that select potential-based forwarding is $\frac{c_{potential}}{c_{potential}+c_{hop}}$, and the proportion of nodes that select hop-based forwarding is $\frac{c_{hop}}{c_{potential}+c_{hop}}$. If the amount by which the potential changes is lower than a specified threshold, τ, the leader nodes set their preferred decision vector $\boldsymbol{g}$ to $[1\ 0]^T$; otherwise, they set $\boldsymbol{g}$ to $[0\ 1]^T$. Nodes that select potential-based forwarding choose the next-hop nodes of their data packets stochastically in accordance with the potential field, that is, by using Equation (8). In contrast, nodes that select hop-based forwarding send their data packets to nodes that are close to a sink node. The assertiveness ω of the leader nodes is set to 1. In other words, the decision vector $\boldsymbol{c}$ of a leader node is always equal to its own preferred decision vector $\boldsymbol{g}$. Note that $c_{potential}$ and c_{hop} are initialized with random values.

We use the network with 100 nodes that is depicted in Figure 3. This network includes four sink nodes (red dots) and 96 sensor nodes (black dots). The 96 sensor nodes are randomly placed in a field of 550 m × 550 m; the four sink nodes are placed at the points (137.5 m, 137.5 m), (137.5 m, 412.5 m), (412.5 m, 137.5 m), and (412.5 m, 412.5 m). The communication range of all nodes is set to 100 m. In this evaluation, we assume that the controller can access the potential information of all nodes with no delay.

In the MAC layer, nodes send data packets to their neighbors using IRDT (Kominami et al. 2013). If a node has no data packets to send, it intermittently broadcasts ID packets to its neighbors at a specified interval to inform them that it is ready to receive a data packet. Note that such a node also informs its neighbors of its own potential value with an ID packet so that its neighbors can update their own potentials. A node that has a data packet to send to a neighbor node does so when it receives an ID packet from a neighbor node. The network settings are summarized in Table 1. The values of parameters $(\alpha, \beta, \sigma, r)$ for optimal control are set to $(0.4, 0.2, 0.1, 10)$.

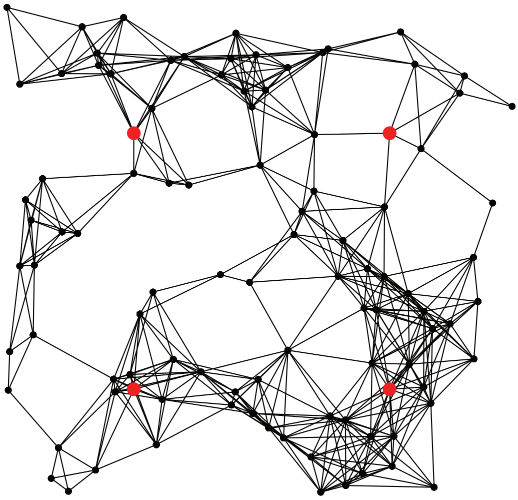


Fig. 3. Network topology.

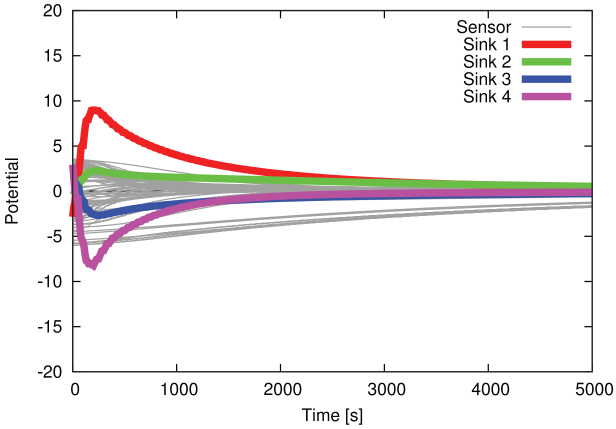


Fig. 4. Potential changes.

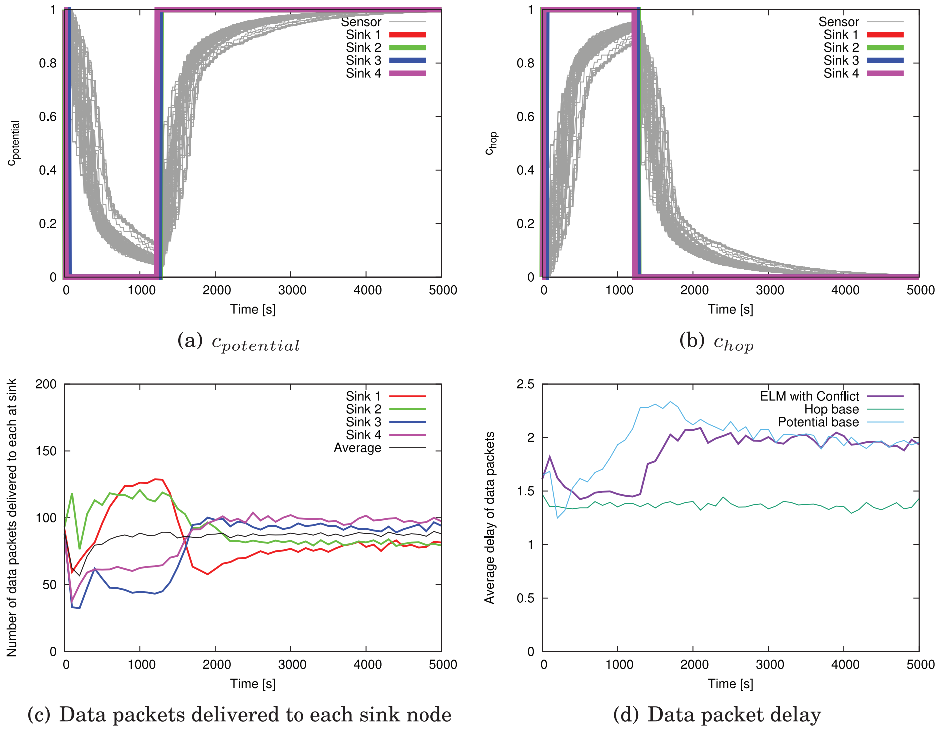
4.3 Adaptation to Environmental Changes Based on Our ELM-based Mechanism

To evaluate and demonstrate the advantages and properties of ELM-based data-packet forwarding we consider four scenarios. First, we compare the following two cases to evaluate the impact of selecting the forwarding scheme.

— *Without data-packet-forwarding selection.* Nodes forward data packets with an identical forwarding scheme (potential-based or hop-based forwarding).

— *Data-packet forwarding based on ELM with leaders sharing their preferred decision vectors.* There are leaders and followers. Leaders update their decision vectors with the same preferred decision vector. Specifically, each leader shares information about its own preferred decision vector and uses the one that is preferred by most leaders to update its decision vector. Followers update their decision vectors using local coordination.

At the beginning of the simulation, the potential values of all nodes are initialized to zero. During the first 1,000s, each node exchanges its potential value with neighbor nodes and updates



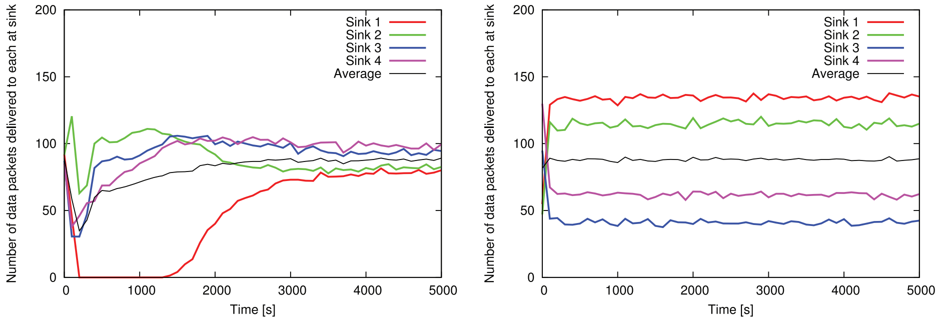
(a) $c_{potential}$

(b) c_{hop}

(c) Data packets delivered to each sink node

(d) Data packet delay

Fig. 5. Results for data-packet forwarding based on the effective leadership model (ELM).



(a) Potential-based forwarding

(b) Hop-based forwarding

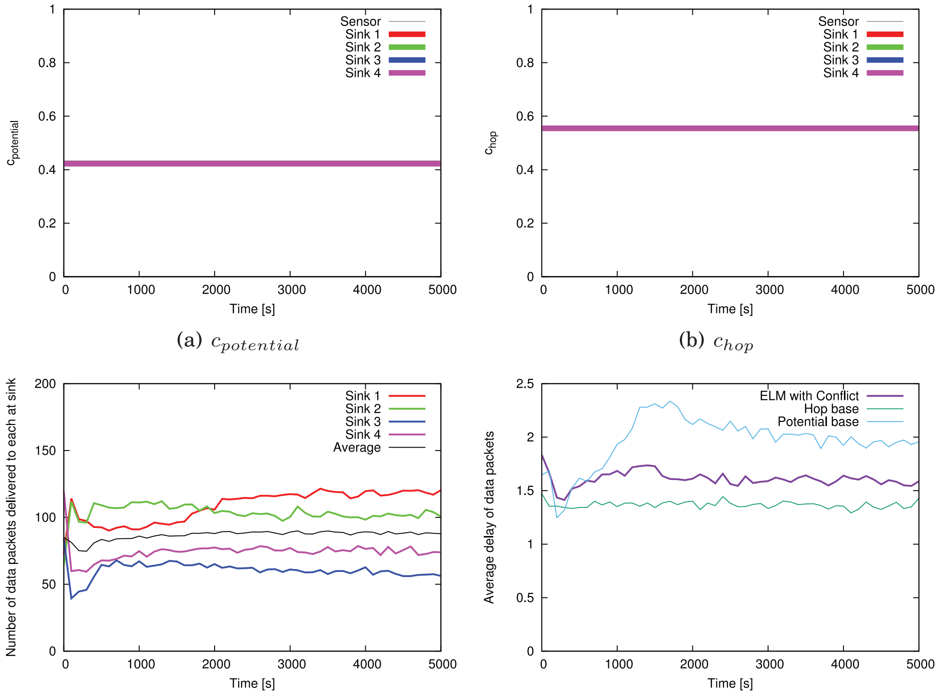
Fig. 6. Data packets received by each sink node when all nodes use the same mechanism for data-packet forwarding.

its potential value so that the potential values are stabilized. At 1,000s, data packets begin to be generated at sensor nodes according to the Poisson process for their flow rates. At 10,000s after the start of the simulation, the data-generation rates at the nodes are changed. We evaluate the changes in the decision vectors, data packets delivered to each sink node, and data-packet delays after traffic changes.

The data-generation rates are set initially to 0.02 packets/s for the sensor nodes in the left-hand half of the network depicted in Figure 3, and to 0.06 packets/s for the remaining sensor nodes. After the traffic change at 10,000s, the data-generation rates are increased to 0.06 packets/s for the

Table 2. Data-packet Drop Rates During the 5,000s After the Traffic Changes

Data-packet forwarding	Drop rate [%]
Based on ELM	3.53
Potential-based	8.86
Hop-based	1.41



(a) $c_{potential}$

(b) c_{hop}

(c) Data packets delivered to each sink node

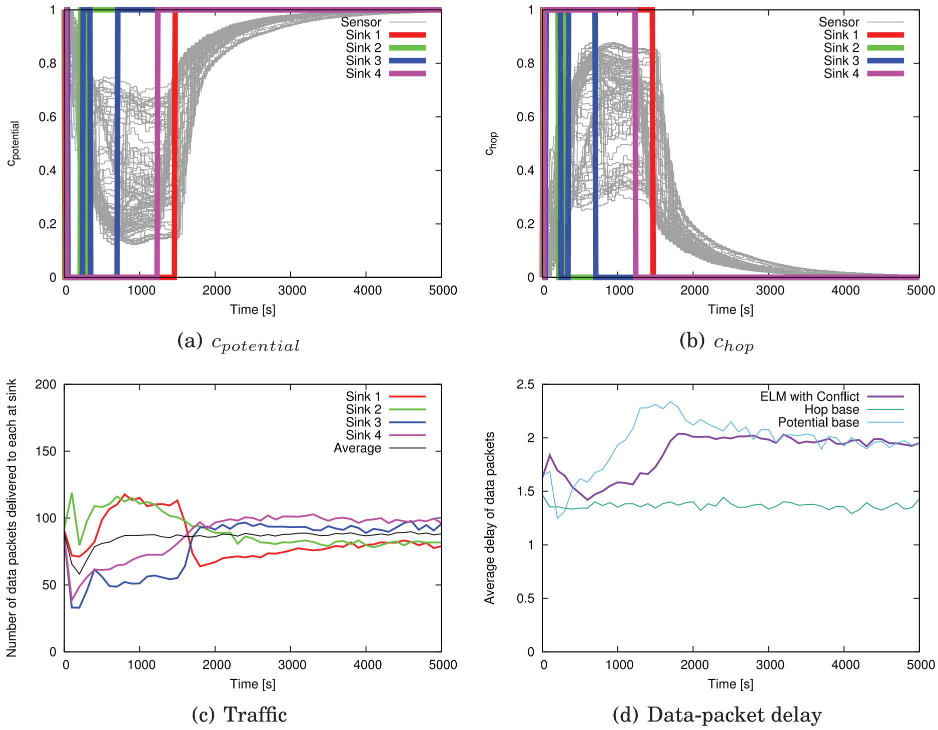
(d) Data-packet delay

Fig. 7. Results for the case in which data-packet forwarding is based on only local interactions.

sensor nodes in the left-hand half, whereas those for the remaining sensor nodes are decreased to 0.02 packets/s. Note that we construct the potential fields such that all sink nodes can receive data packets at equal rates because load balancing is known to be a challenging task for a WSN. We set the four sink nodes to be controlled nodes that receive control feedback from the controller. The four sink nodes are also set to be leader nodes that guide the other nodes to their preferred states.

Figure 4 shows potential changes against time. The horizontal axis represents the elapsed time after traffic changes, and the vertical axis represents $X(t) = \bar{\Theta}(t) - \Theta(t)$, the difference between each potential and its target value. The colored lines correspond to sink nodes, whereas the gray lines correspond to sensor nodes. Note that the potential changes do not depend on changes in the decision vectors.

Figure 5 shows the results for the case in which data-packet forwarding is based on effective-leadership nodes. In that figure, the horizontal axes represent the elapsed time after traffic changes. The vertical axes in Figure 5(a) and (b) represent the values of $c_{potential}$ and c_{hop}, respectively. Figure 5(c) shows the numbers of data packets delivered to each sink node in the previous 100s and their average values. Figure 5(d) shows data-packet delays before arrival at sink nodes, averaged



(a) $c_{potential}$

(b) c_{hop}

(c) Traffic

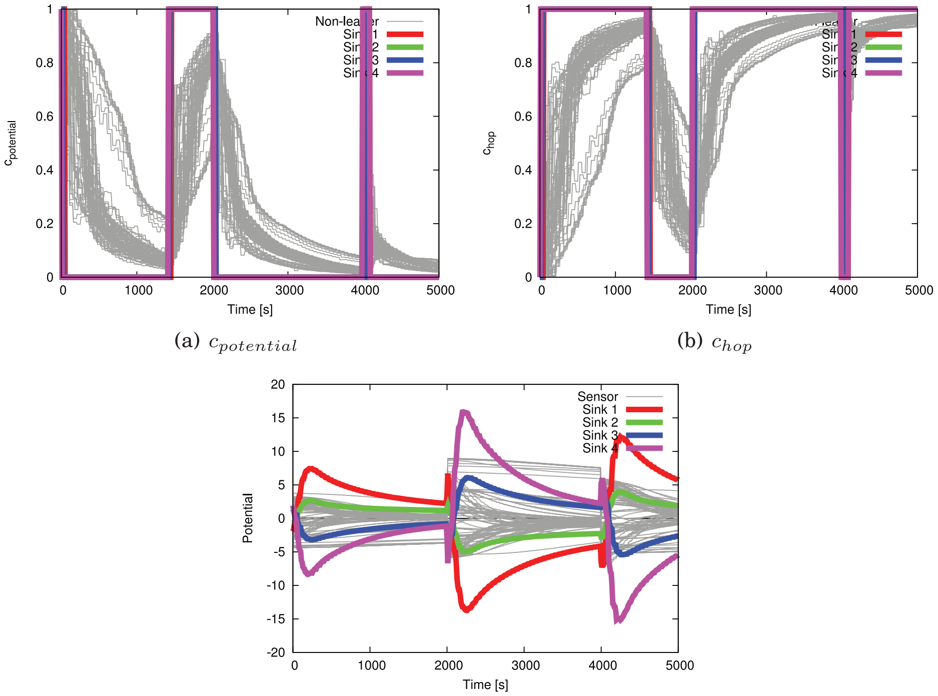
(d) Data-packet delay

Fig. 8. Results in the case in which data-packet forwarding is based on ELM (but leaders do not share their preferred decision vectors).

over the previous 100s. In this case, leader nodes change their decision vectors $[c_{potential}\ c_{hop}]^T$ to $[0\ 1]^T$ just after traffic changes, as shown in Figure 5(a) and (b). This is because leader nodes perceive changes of potentials according to traffic changes, and set their preferred decision vectors to $[0\ 1]^T$. Follower nodes follow leader nodes through local coordination, and their decision vectors approach $[0\ 1]^T$. When the potential changes become small, the leader nodes change their preferred decision vectors to $[1\ 0]^T$, and follower nodes follow them. Under this situation, data packets are forwarded in accordance with the state of the network, specifically the state of potentials, in this evaluation.

For comparison, in Figure 6 we show changes in the data packets delivered to each sink node in the case in which nodes all use the same data-packet forwarding scheme—either potential-based forwarding or hop-based forwarding. In this evaluation, the potential field is constructed so that the numbers of data packets delivered to each sink node are equal for load balancing. With potential-based forwarding, the numbers of data packets delivered to each sink node become different after traffic changes, as shown in Figure 6(a). This is because the potential field is reconstructed according to the traffic changes. Roughly 3,000s after the traffic changes, the potentials come close to converging to their target values, and therefore the numbers of data packets delivered to each sink node become approximately equal.

One problem in the case with potential-based forwarding is that, during the potential-field reconstruction, the next-hop nodes of the data packets are not selected correctly. Consequently, the average number of data packets reaching the sink nodes is reduced. This is because some sink nodes temporarily have the largest potential values within their communication ranges according



(a) $c_{potential}$

(b) c_{hop}

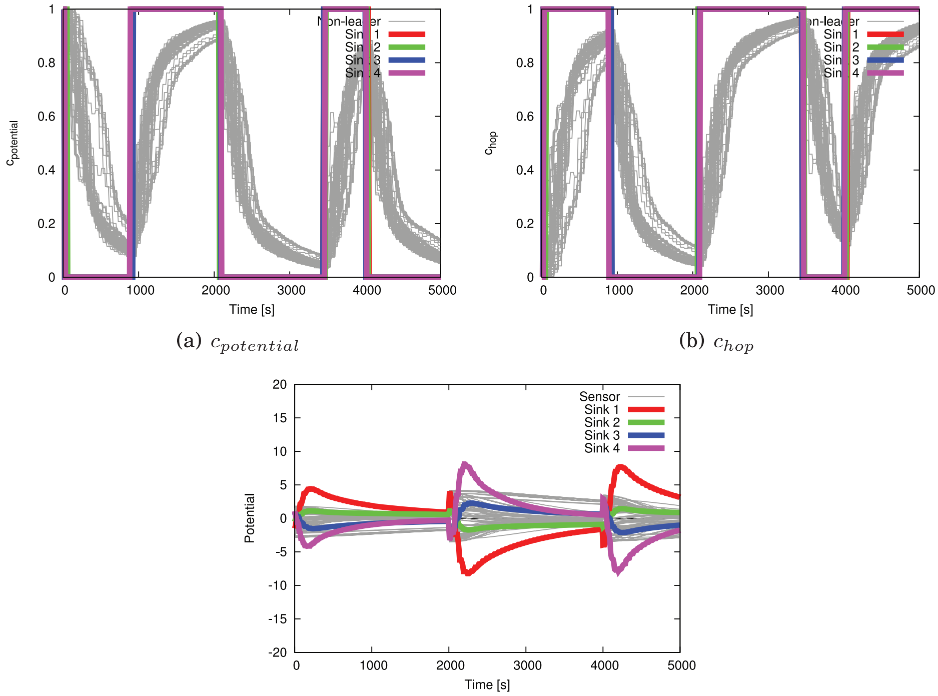
(c) Potential value

Fig. 9. Results for the case when the communication range is 90m.

to the control inputs, so data packets cannot arrive at sink nodes and are sometimes dropped. In contrast, with hop-based forwarding, data packets can be delivered to sink nodes with a small delay regardless of environmental changes because each node forwards data packets to neighbor nodes that are closer to a sink node. One problem with hop-based forwarding is that load balancing is not considered. The numbers of data packets delivered to each sink node are different all the time, as shown in Figure 6(b), which indicates that traffic is concentrated in a part of the network. Such concentration of traffic can shorten the network lifetime. The appropriate forwarding scheme depends on the network state.

With data-packet forwarding based on the ELM, nodes select the data-packet forwarding scheme stochastically in accordance with the state of the potentials. For roughly 1,800s after the traffic changes, the numbers of data packets delivered to each sink node are different, as shown in Figure 5(c), as they are in the hop-based forwarding case shown in Figure 6(b). This is because, according to the potential changes, the decision vectors of nodes approach $[0\ 1]^T$ through leaders' preference and local coordination. The number of data packets delivered to sink nodes is also reduced just after the traffic changes, when the decision vectors are still close to $[1\ 0]^T$ and the nodes are likely to select potential-based forwarding. However, the decision vectors of the nodes approach $[0\ 1]^T$ soon after and, as a result, data packets arrive at the sink nodes.

Table 2 gives the data drop rates during the 5,000s after the traffic changes. Compared with potential-based forwarding, the data drop rate is lower for data-packet forwarding based on the ELM. From roughly 1,800s after the traffic changes, the numbers of data packets delivered to each sink node are approximately equal, as shown in Figure 5(c), as they are in the potential-based



(a) $c_{potential}$ (b) c_{hop}

(c) Potential value

Fig. 10. Results for the case when the communication range is 100m.

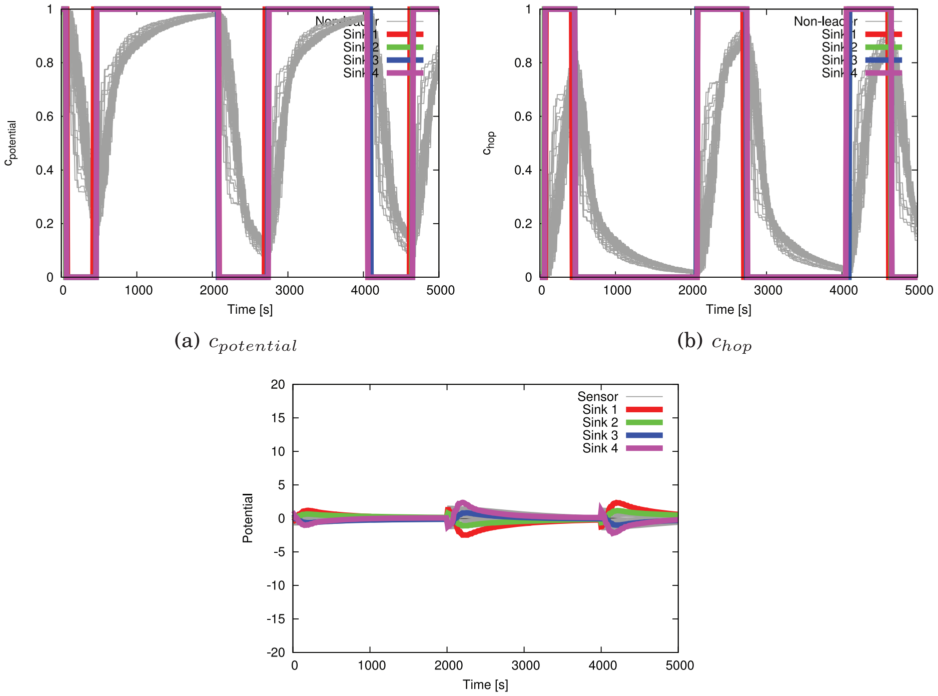
forwarding case shown in Figure 6(a). This is because potentials come close to converging to their target values, and the decision vectors of the nodes approach $[0\ 1]^T$.

It is worth mentioning that nodes select the data-packet forwarding scheme in accordance with the network state, although each follower node perceives only information about its neighbors; that is, the information held by each follower node is incomplete. Moreover, follower nodes do not know which nodes are leaders, and simply follow their neighbors through local interactions. Consequently, with data-packet forwarding based on the ELM, nodes can forward data packets in accordance with the network state even when the perceived information is incomplete and dynamic.

Next, we evaluate the following case to demonstrate the role of the leader nodes.

— *Data-packet forwarding based on only local coordination.* There are no leader nodes. Each node updates its own decision vector using local coordination, and decides stochastically which forwarding scheme to use in accordance with its decision vector.

Figure 7 shows the results for the case in which data-packet forwarding is based on only local coordination. In that case, both $c_{potential}$ and c_{hop} converge to roughly 0.5 and do not change, as shown in Figure 7(a) and (b), despite the changes in potential after the traffic changes, as described for Figure 4. Without leader nodes, all nodes update their decision vectors with local coordination. For this reason, the decision vector of each node converges to the average values of all initial decision vectors. In this evaluation, $c_{potential}$ and c_{hop} are initialized with random values so that the average values of the initial decision vectors are approximately 0.5. Therefore, $c_{potential}$ and c_{hop} finally converge to roughly 0.5. Moreover, because there are no leaders to update the decision



(a) $c_{potential}$

(b) c_{hop}

(c) Potential value

Fig. 11. Results for the case when the communication range is 120m.

vectors with a preferred decision vector determined in accordance with the state of the potentials, the decision vectors do not change even when the potentials are changing.

In that situation, the proportions of nodes that select potential-based forwarding and that select hop-based forwarding are approximately equal. As a result, both the load balancing of potential-based forwarding and the small communication delay of hop-based forwarding are lost, as shown in Figure 7(c) and (d). This indicates that leader nodes play an important role in selecting the means of data-packet forwarding according to the network state.

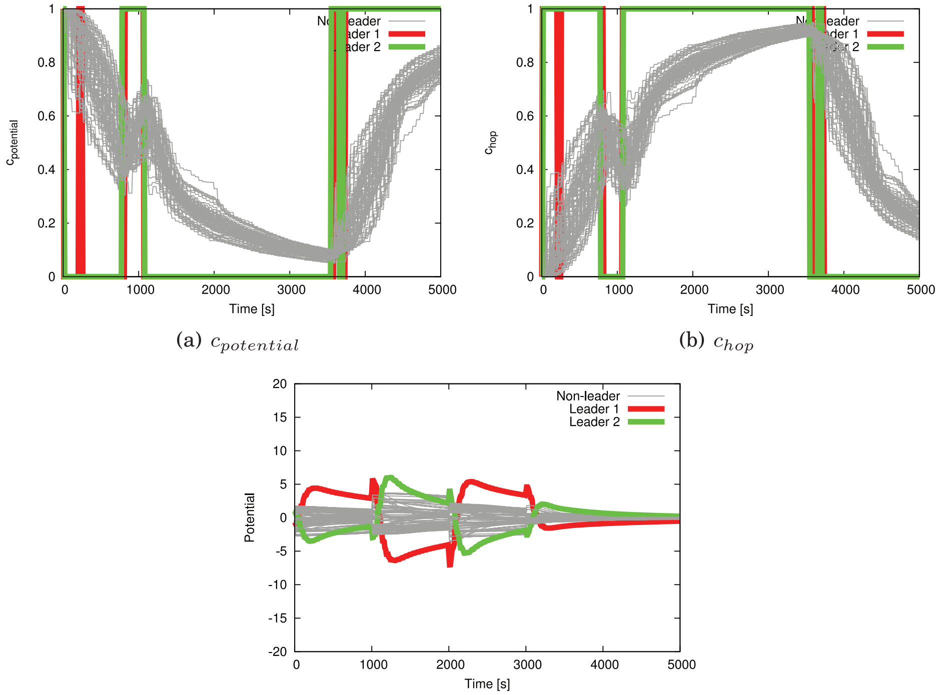
Finally, we evaluate the following case.

— *Data-packet forwarding based on the ELM with leaders not sharing their preferred decision vectors.* There are leaders and followers. Leaders update their decision vectors with their own preferred decision vectors, whereas followers update their decision vectors using local coordination.

Figure 8 shows the results for data-packet forwarding based on the ELM with leaders not sharing their preferred decision vectors. In this case, leader nodes change their decision vectors $[c_{potential}\ c_{hop}]^T$ to $[0\ 1]^T$ just after the traffic changes, as shown in Figure 8(a) and (b), as in the case with data-packet forwarding based on the ELM with leaders sharing their preferred decision vectors, which is shown in Figure 5(a) and (b). However, some leader nodes change their preferred decision vectors to [1 0] faster than other leader nodes. This is because the information that can be perceived is different among leader nodes. As a result, the decision vectors of follower nodes do not approach either [0 1] or [1 0].

Table 3. The Diameter and the Minimum/Average/Maximum Distance Between a Node and Its Nearest Leader Node in the Networks Used in the Evaluation of Section 4.4

Communication range	Diameter	Distance to the leader node		
		Min	Avg	Max
90	12	1	1.81	4
100	10	1	1.62	3
120	7	1	1.36	2



(a) $c_{potential}$

(b) c_{hop}

(c) Potential value

Fig. 12. Results in the case with two leader nodes.

In that situation, either the potential-based or hop-based forwarding scheme can be selected for data-packet forwarding. The difference from the case in which data-packet forwarding is based on only local coordination is that each leader node can adapt to environmental changes and follower nodes are affected more by closer leader nodes. This indicates that nodes can adapt to local environmental changes even if leader nodes are in conflict with each other.

4.4 Influence of the Network Density on Adaptability

Intuitively, given a certain number of leader nodes, the speed of adaptation of follower nodes is faster in a denser network. To investigate the influence of network density on adaptability, we use the network shown in Figure 3, and set the communication range of nodes to 90, 100, or 120 m. We set the four sink nodes to be controlled nodes that receive control feedback from the controller.

The four sink nodes are also set to be leader nodes that guide the other nodes to their preferred states.

At the beginning of the simulation, the potential values of all nodes are initialized to zero. During the first 1,000s, each node exchanges its potential value with neighbor nodes and updates its potential value so that the potential values are stabilized. At 1,000s, data packets begin to be generated at sensor nodes according to the Poisson process for their flow rates. At 10,000s after the start of the simulation, the data-generation rates at the nodes are changed. At 11,000s and 12,000s, the generation rates at nodes are changed again. Finally, at 13,000s, the generation rates at nodes are changed back to the initial values. We evaluate the adaptation speed after traffic changes (from 10,000s to the ending of the simulation). Note that the networks whose communication ranges are 90, 100, and 120m have different topologies which affects data-packet flow and delays as well as potential changes. Therefore, in this evaluation, we focus on the adaptation speed of the decision vectors of the follower nodes.

The data-generation rates are set initially to 0.04 packets/s for all sensor nodes. During 10,000–11,000s and 12,000–13,000s, the data-generation rates are set to 0.06 packets/s for the sensor nodes in the left-hand half of the network depicted in Figure 3, and to 0.02 packets/s for the remaining sensor nodes. During 11,000–12,000s, the data generation rates are set to 0.02 packets/s for the sensor nodes in the left-hand half, whereas those for the remaining sensor nodes are set to 0.06 packets/s. After 13,000s, the data generation rates are set to the initial values, that is, 0.04 packets/s for all nodes. In this simulation, we also construct the potential fields such that all sink nodes can receive data packets at equal rates.

We show the results of our proposal when the communication range is set to 90, 100, and 120m in Figures 9, 10, and 11, respectively. These figures plot changes of $c_{potential}$, c_{hop}, and potential values after traffic changes (from 10,000s to the ending of the simulation).

Figures 9, 10, and 11 indicate that the adaptation of decision vectors to changes of network condition is faster with a larger communication range. The larger the communication range, the shorter the distances between follower nodes and leader nodes. The influence of the leader nodes' preferred values of decision vectors is propagated hop by hop so that the speed of adaptation of the decision vectors of followers is faster when these distances are shorter. In Table 3, we show the graph diameter (the length of the longest shortest path between any two nodes) and the distance between a node and its nearest leader node in these networks.

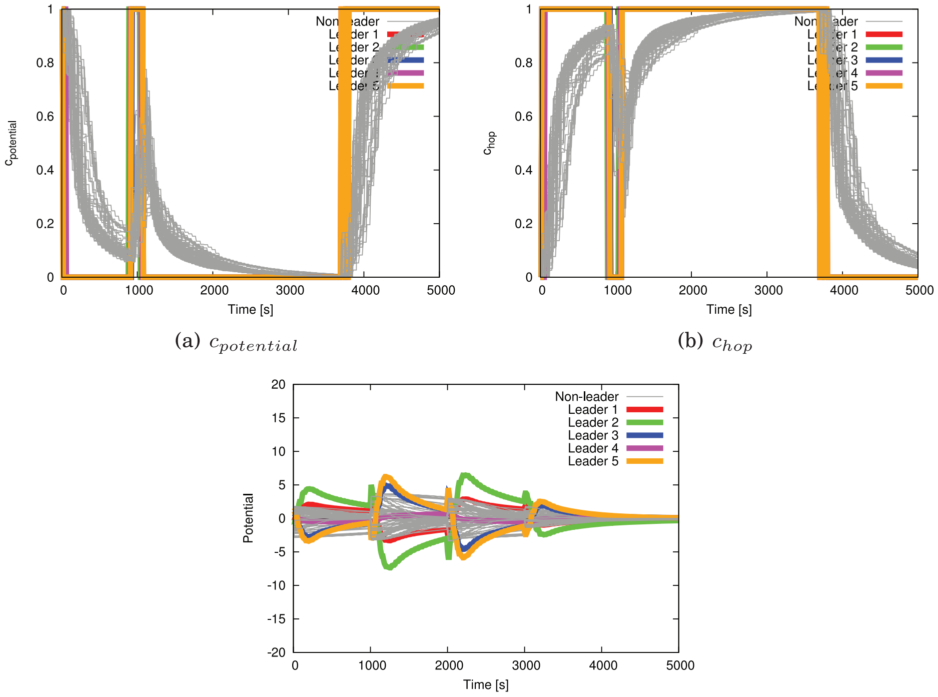
4.5 Influence of the Number of Leader Nodes on Adaptability

To investigate the influence of the number of leader nodes on adaptability we compare networks with 2, 5, and 10 leader nodes.

To shorten the distance between each node and a leader node, we select leader nodes as follows. When we select l leader nodes, we first divide the network into l sub-networks according to coordinates of nodes using k-means clustering. We then select a node in each sub-network as a leader node so that the maximum distance between the leader node and nodes within the corresponding sub-network is minimized. Note that leader nodes are also set to be the controlled nodes. The settings of traffic changes are the same as in Section 4.4. Since the number of leader nodes affects data-packet flow and delays as well as potential changes, in this simulation we also focus on the speed of adaptation of decision vectors of follower nodes.

We show the results for 2, 5, and 10 leader nodes in Figures 12, 13, and 14, respectively. These figures plot changes of $c_{potential}$, c_{hop}, and potential values after traffic changes (from 10,000s to the ending of the simulation).

Figures 12, 13, and 14 indicate that the adaptation of the decision vectors to changes of network condition is faster with a larger number of leader nodes. With two leader nodes, as



(a) $c_{potential}$

(b) c_{hop}

(c) Potential value

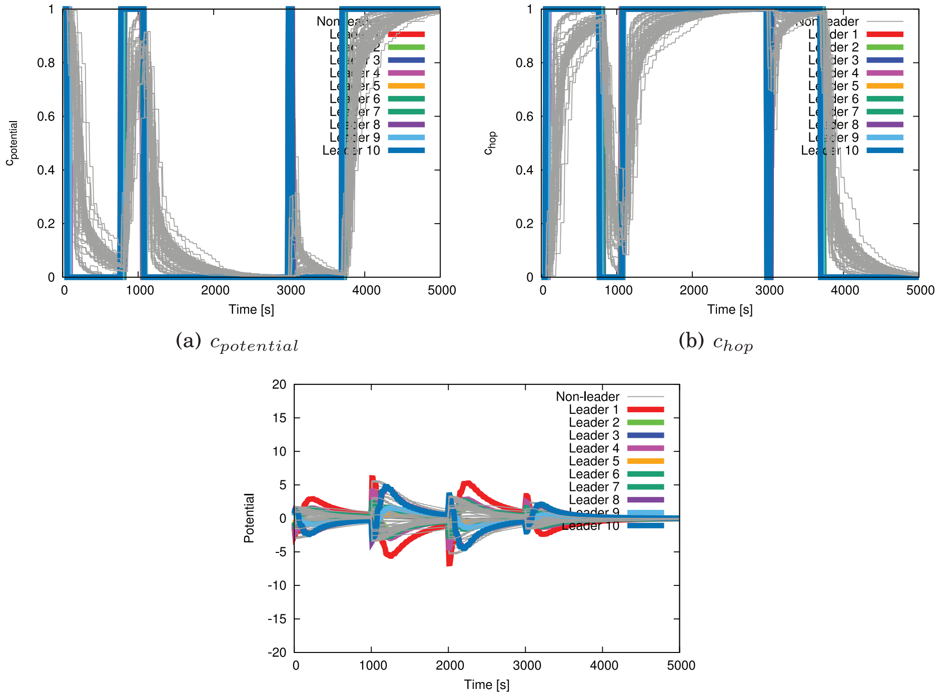
Fig. 13. Results in the case with five leader nodes.

shown in Figure 12, the values of $c_{potential}$ and c_{hop} of follower nodes reach at most 0.05 and 0.95, respectively, although the potential values are changing and leader nodes prefer to select the hop-based forwarding scheme. In contrast, with 5 or 10 leader nodes, as shown in Figures 13 and 14, the values of $c_{potential}$ and c_{hop} of followers reach nearly 0 and 1 in 2,000s (from 1,000s to 3,000s). Comparing the case with 5 leader nodes with that with 10 leader nodes, the speeds of adaptation of the decision vectors are not so different, although the number of leader nodes increases twofold.

In conclusion, if the number of leader nodes is too small, the speed of adaptation of decision vectors is slow, which leads slow adaptation to environmental changes. However, if the number of leader nodes is larger than a certain value (5 in the evaluation of this section), decision vectors can adapt quickly to environmental changes. This indicates that fast adaptation can be achieved with a comparatively small number of leader nodes, which is an advantage of our proposed scheme because the deployment cost of leader nodes is higher than that of follower nodes.

5 CONCLUSION AND FUTURE WORK

Uncertainty of information is a significant problem for the practical use of self-organizing control systems. To develop a system that can overcome this problem, we applied a mathematical model based on the collective decision-making of animal groups. The model used is the effective leadership model (ELM). This enables the system to achieve consensus even under adverse conditions. In this study, we considered a network using potential-based routing with optimal control and supplemented it with a variant of the ELM applied to the selection of a packet-forwarding scheme by network nodes. Through computer simulations, we demonstrated the advantages and properties of our proposed scheme.



(a) $c_{potential}$

(b) c_{hop}

(c) Potential value

Fig. 14. Results in the case with 10 leader nodes.

In future work, we will investigate the relationships among network size, leader node proportion, and performance in other information networks. In the original ELM, it has been demonstrated that the fraction of informed individuals required to achieve consensus decreases as the group becomes larger (Couzin et al. 2005). Moreover, we intend to investigate the influence of information ambiguity.

REFERENCES

A. C. Antoulas, D. C. Sorensen, and S. Gugercin. 2006. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics* 280 (Oct. 2006), 193–219.

Anindya Basu, Alvin Lin, and Sharad Ramanathan. 2003. Routing using potentials: A dynamic traffic-aware routing algorithm. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. ACM, 37–48.

Larissa Conradt. 2011. Models in animal collective decision-making: Information uncertainty and conflicting preferences. *Interface Focus* (Dec. 2011), 1–5.

Larissa Conradt. 2013. Collective animal decisions: Preference conflict and decision accuracy. *Interface Focus* 3, 6 (Oct. 2013), 1–12.

L. Conradt, J. Krause, I. D. Couzin, and T. J. Roper. 2009. Leading according to need in self-organizing groups. *The American Naturalist* 173, 3 (March 2009), 304–312.

Iain D. Couzin, Jens Krause, Nigel R. Franks, and Simon A. Levin. 2005. Effective leadership and decision-making in animal groups on the move. *Nature* 433, 7025 (Nov. 2005), 513–516.

Falko. Dressler. 2008. *Self-organization in Sensor and Actor Networks*. Wiley.

Suyong Eum, Yozo Shoji, Masayuki Murata, and Nozomu Nishinaga. 2015. Design and implementation of ICN-enabled IEEE 802.11 access points as nano data centers. *Journal of Network and Computer Applications* 50 (April 2015), 159–167.

Sangsu Jung, Malaz Kserawi, Dujeong Lee, and J.-K. K. Rhee. 2009. Distributed potential field based routing and autonomous load balancing for wireless mesh networks. *IEEE Communications Letters* 13, 6 (June 2009), 429–431.

Daichi Kominami, Masashi Sugano, Masayuki Murata, and Takaaki Hatauchi. 2013. Controlled and self-organized routing for large-scale wireless sensor networks. *ACM Transactions on Sensor Networks* 10, 1 (Nov. 2013), 13:1–13:27.

Naomi Kuze, Daichi Kominami, Kenji Kashima, Tomoaki Hashimoto, and Masayuki Murata. 2016. Controlling large-scale self-organized networks with lightweight cost for fast adaptation to changing environments. *ACM Transactions on Autonomous and Adaptive Systems* 11, 2 (June 2016), 9:1–9:25.

Munyoung Lee, Junghwan Song, Kideok Cho, Sangheon Pack, Jussi Kangasharju, Yanghee Choi, and Ted Taekyoung Kwon. 2015. SCAN: Content discovery for information-centric networking. *Computer Networks* 83 (June 2015), 1–14.

Christian Müller-Schloer, Hartmut Schmeck, and Theo Ungerer. 2011. *Organic Computing-a Paradigm Shift for Complex Systems*. Birkhaeuser, Berlin.

Mikhail Prokopenko. 2014. *Guided Self-organization: Inception*. Springer, Berlin.

Alireza Sheikhattar and Mehdi Kalantari. 2014. Distributed load balancing using alternating direction method of multipliers. In *Proceedings of 2014 IEEE Global Communications Conference (GLOBECOM'14)*. IEEE, 392–398.

Chengjie Wu, Ruixi Yuan, and Hongchao Zhou. 2008. A novel load balanced and lifetime maximization routing protocol in wireless sensor networks. In *Proceedings of the 67th IEEE Vehicular Technology Conference*. IEEE, 113–117.

H. T. Zhang, M. Z. Chen, G. B. Stan, T. Zhou, and J. M. Maciejowski. 2008. Collective behavior coordination with predictive mechanisms. *IEEE Circuits and Systems Magazine* 8, 3 (Aug. 2008), 67–85.

Kemin Zhou, John Comstock Doyle, Keith Glover, and others. 1995. *Robust and Optimal Control*. Prentice Hall, New Jersey.

Received April 2017; revised December 2017; accepted January 2018