Controlling large-scale self-organized networks with lightweight cost for fast adaptation to changing environments

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

Self-organization has potential for high scalability, adaptability, flexibility, and robustness, which are vital features for realizing future networks. Convergence of self-organizing control, however, is slow in some practical applications compared to control with conventional deterministic systems using global information. It is therefore important to facilitate convergence of self-organizing controls. In *controlled self-organization*, which introduces an external controller into self-organizing systems, the network is controlled to guide systems to a desired state. Although existing controlled self-organization schemes could achieve this feature, convergence speed for reaching an optimal or semi-optimal solution is still a challenging task. We perform potential-based self-organizing routing, and propose an optimal feedback method using a reduced-order model for faster convergence at low cost. Simulation results show that the proposed mechanism improves the convergence speed of potential-field construction (i.e., route construction) by at most 22.6 times with low computational and communication cost.

1. INTRODUCTION

Self-organization, where components behave individually and autonomously, is a natural phenomenon in natural distributed systems [Pintea 2014; Yang et al. 2013]. In a self-organizing system, each component follows simple rules using locally available information. Through direct or indirect interactions among components, a global behavior or pattern emerges on a macroscopic level without a central control entity. In a self-organizing system, up-to-date information regarding the entire system or many

other components is unnecessary, which considerably reduces computational cost and communication overhead for collecting global information. This localized control leads to a capability of handling local failures and small environmental changes by interaction of local components. Thus, self-organizing systems are expected to automatically recover from failures and adapt to environmental changes, without involving centralized control. These are reasons why a variety of self-organization-based models have been applied to information networking such as routing, synchronization, and task assignment [Zhang et al. 2013; Zheng and Sicker 2013]. In future large-scale, complex networks we can expect features such as scalability, adaptability, and robustness to be improved to an extent not possible by conventional network control methods [Balasubramaniam et al. 2011].

Although self-organization control without global knowledge of the current network state has various benefits, such control has critical disadvantages that complicate practical use in industrial and business systems [Dressler 2008]. It may take a long time for global patterns to emerge in large-scale systems, because they appear as a consequence of interactions between autonomous components. This property also leads to slow adaptation to large environmental changes, which is difficult to solve solely by local interaction in self-organizing systems. Also, self-organizing systems that use only local information sometimes fall into local optima. On the other hand, in conventional centralized systems global information can reach an optimal solution, though the required computational cost to do so often leads to unrealistic solutions.

Such disadvantages raised from real applications brought about the idea of controlled (guided, managed) self-organization, where the self-organizing system is controlled through some constraints [Branke et al. 2006; Schmeck et al. 2010; Prokopenko 2014]. For example, the authors of [Arakawa et al. 2011; Kominami et al. 2013] use the concept of controlled self-organization, where an external observer/controller guides self-organizing optical network [Arakawa et al. 2011] and sensor network [Kominami et al. 2013] systems through a feedback mechanism that leads them to a desired state. The self-organizing system can then be controlled through fully observed information. However, enhancing the convergence speed to reach an optimal or semi-optimal solution remains as an outstanding task.

We previously introduced an external controller having an optimal feedback mechanism to self-organizing systems in [Kuze et al. 2014]. The external controller collects information regarding the network such as node states and network topology via a partial set of nodes directly monitored by the controller, and estimates system dynamics using a mathematical model that describes the network dynamics, then determines optimal control inputs based on robust control theory [Zhou et al. 1995] for facilitating the convergence. Optimal feedback mechanisms for controlling dynamical systems have been researched for long years in the field of control theory [Kirk 2012]. A controller monitors a system and provides an optimal control feedback that minimizes the cost function [Kirk 2012] based on a mathematical model of the system. Note that there are various errors such as modeling errors and unexpected disturbances so that the optimal control feedback is not necessarily optimal at any time in real systems. Simulation results in [Kuze et al. 2014] showed that the mechanism improves the convergence speed of self-organization, however, in large-scale networks especially, it is generally difficult for the controller to collect detailed network information and to estimate the network dynamics [Mulvey et al. 1995], because doing so requires considerable costs and, even worse, loses the advantage of scalability of self-organizing systems.

The contribution of this paper is to converge self-organizing network systems with lightweight cost while retaining a high performance (such as high convergence and adaptability in our proposal) of robust control. To that end, we first regulate the

#node = N $\bigcirc$: Node $\bullet$: Observable/ controllable node $\leftrightarrow$: Local interaction $\leftrightarrow$: Controller's I/O (from/to $\bullet$)	No control scheme	Robust control scheme	Robust control scheme with model reduction
		External Controller	External Controller
Model size of controller	None	pN $(p \in \mathbb{N}^+)$	h $(\ll N)$
Computation cost at controller	None	High $(O(N^3))$	Low $(O(h^3))$
	Potentials X / convergence / Time	Potentials X / convergence / Time	Potentials X / convergence / Time
Convergence speed	Slow	Fast	Fast (perturbation may occur due to modeling error)

Fig. 1. Advantages of optimal feedback with model reduction ($\mathbb{N}^+$ is the set of all non-zero natural numbers)

area in which the external controller collects node states to reduce the cost for collecting information (Figure 1). If the controller collects all node states, the communication overhead is extremely large and traffic congestion would occur. Therefore, we limit the area from which the controller can collect information to nodes that can be reached within several hops from the monitored nodes. The controller can estimate the states of all nodes from only a part of them given the network topology, because the state of each node is determined in accordance with local interactions based on prescribed rules [Zhou et al. 1995]. That is, the controller acquires network information with lower communication overhead, then calculates optimal feedback inputs. In our scheme, a controller has an internal model estimating the actual network system. The order (N_{dim}) of that model is proportional to that of the actual network model, namely the number of nodes. Note that we need $O(N_{dim}^2)$ for controller state updates at each time instance, and also $O(N_{dim}^3)$ computation for controller re-design when the network topology changes. The computational cost is thus extremely large in large-scale networks when we use the original model, whose order is proportional to the number of nodes. We therefore reduce the order of the network model (consequently that of N_{dim}) to decrease the computational cost via a model reduction technique [Antoulas et al. 2006]. As shown in Figure 1, model reduction refers to reducing the number of state variables in dynamical systems, while retaining suitable input/output characteristics. Note that there is a trade-off between the optimization of control input and the number of state variables, i.e., between the convergence speed and the computational cost. Intuitively, a reduced-order model leads to the loss of the original model while de-

creasing the computational cost. This trade-off relation requires us to further examine the number h of state variables of the reduced-order model.

The effectiveness of our proposal is evaluated through computer simulation studies where we consider potential-based routing—a self-organizing routing mechanism for wireless sensor networks— with optimal feedback, and we evaluate the convergence speed after environmental changes. Wireless sensor networks represent one suitable application of our proposal. This is because our proposal requires the expensive computation of control input for an external powerful controller and not for each component of a self-organizing system. Therefore, the performance of the whole system can be improved even though each component in the systems does not need much computational power and energy consumption. Optimality of our feedback mechanism is analytically guaranteed in synchronous systems, but not in asynchronous systems. Then, we first assume a wireless sensor network where nodes behave asynchronously and the controller can observe the network state via the monitored nodes experiencing communication delay. To evaluate the robustness of our proposal against information loss in the controller, we next simulate an environment where several nodes fail and the controller cannot immediately detect node failures. Through these evaluations, we will show that optimal feedback using a reduced-order model can enhance the convergence speed of self-organizing systems at fairly low cost, even if the network is large. Moreover, optimal feedback does not have a negative impact on the robustness that is a notable property of self-organizing systems. Note that potential-based routing for wireless sensor networks is just an example of application for our proposal.

The remainder of this paper is organized as follows. Firstly, we briefly explain potential-based routing in Section 2. We propose and explain potential-based routing with optimal feedback in Section 3, and explain a reduced-order model with which the controller estimates the network dynamics in Section 4. We then show fast adaptation to an environmental change of the proposed method through simulation and give a discussion of our proposal in Section 5. Finally, in Section 6 we present our conclusions and suggest areas for future work.

2. POTENTIAL-BASED ROUTING

Potential-based routing is a self-organizing routing mechanism being active in the fields of wireless sensor networks, 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 2013; Eum et al. 2014; Lee et al. 2014]. Here we assume that potential-based routing is used in wireless sensor networks. In potential-based routing, each node has a scalar value called its *potential*, and data packets are forwarded to a neighbor whose potential is smaller than the forwarder's. In wireless sensor networks, data packets are generally sent to a sink node, and the fewer hops from the sink node a node is, the lower the potential value assigned to the node. The simple forwarding rule "forward data to a neighboring node with lower potential" can therefore result in data packet collection toward sink nodes, as illustrated in Figure 2. Potential-based routing has high scalability because each node uses only local information for calculating potentials and a local rule for forwarding data. Furthermore, it can achieve load balancing and consequently network lifetime improvement by calculating potentials using information such as flow rates, queue lengths, or remaining energy [Wu et al. 2008]. In Subsections 2.1 and 2.2, we describe a potential field construction method and show how to select a next hop node using the gradient of the field.
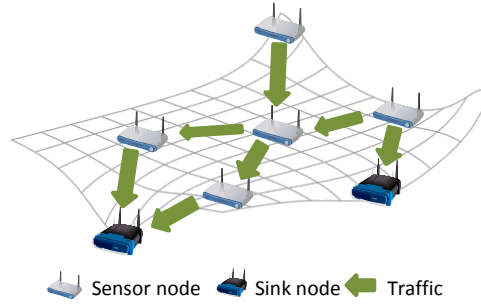
Sensor node   Sink node   Traffic

Fig. 2.   Potential-based routing

2.1. Potential Field Construction

Sheikhattar and Kalantari [Sheikhattar and Kalantari 2013] focused on the convergence of potential-based routing and achieved enhancement of the potential convergence speed. They proposed a potential calculation method based not only on current potentials but also on last potentials to accelerate potential convergence. Node n's potential at time t, $\theta_n(t)$, is calculated by Equation (1).

$$
\begin{aligned}
\theta_n(t+1) =&(\alpha+1)\theta_n(t) - \alpha\theta_n(t-1) \\
&+ \beta\sigma_n \left(\sum_{k \in \mathcal{N}_b(n)} \{\theta_k(t) - \theta_n(t)\} + f_n(t) \right),
\end{aligned}
\tag{1}
$$

where $\mathcal{N}_b(n)$ is the set of neighbors of node n, and α is a parameter that determines the weights of the current and the last potential values when calculating the next potential. Larger α means that the weight of the last potential value is larger and therefore the system becomes less subject to current noise, though the convergence speed is slower. Parameter β determines the influence amount of neighbor node potentials. σ_n is defined as $\sigma_0/|\mathcal{N}_b(n)|$ (σ_0 is a parameter), and $f_n(t)$ corresponds to the flow rate of node n at time t. If $f_n(t)$ is a positive value it means the data generation rate of node n, whereas if $f_n(t)$ is negative it means the rate of data packets delivered to node n. For sink node n, $f_n(t)$ corresponds to targeted flow rates which are given by the network manager. If the flow conservation constraint is upheld, that is, $\sum_{n \in \{1, \cdots, N\}} f_n(t) = 0$, a potential field is constructed so that actual rates of data packets delivered to nodes satisfy given flow rates, i.e., all gradients, which are potential differences between next hop nodes, correspond to the appropriate flow rates between next hop nodes.

The convergence speed based on Equation (1) is faster than simple Jacobi iterations (such as our previous work [Kominami et al. 2013]), but still takes a long time to converge due to its calculation being based only on local information (see Section 5 for an example convergence). In both cases, potentials of nodes converge as a result of the iterative behavior (i.e., potential updates of nodes), and therefore, the convergence speed is faster with a shorter interval of potential updates. Instead of relying on only local interactions among nodes, we introduce into potential-based routing an external controller that observes network states (potential values), estimates its future state, and regulates potentials of a partial set of nodes for faster convergence.
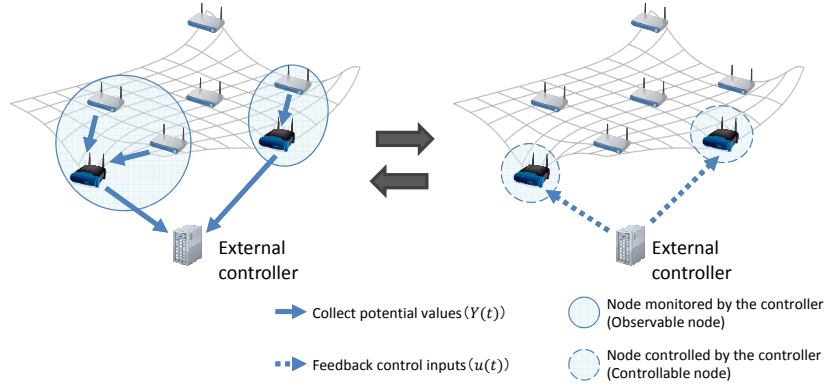
Fig. 3. Potential-based routing with a controller's feedback, where an external controller collects potential values from observable nodes and introduces control inputs to controllable nodes periodically.

2.2. Routing

If a node has a data packet, it 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 next node that is assigned a lower potential value than itself, and the packet eventually arrives at a sink node. Specifically, a next-hop node is selected proportionally with potential values, that is, the probability $p_{i \to j}(t)$ that sensor node i selects a neighbor node j as the next-hop node for a data packet at time t is given by

$$p_{i \to j}(t) = \begin{cases} \frac{\theta_i(t) - \theta_j(t)}{\sum_{k \in \mathcal{N}_l(i)} \{\theta_i(t) - \theta_k(t)\}}, & \text{if } j \in \mathcal{N}_l(i) \\ 0, & \text{otherwise} \end{cases} ,$$

where $\mathcal{N}_l(i)$ is the set of node i's neighbor nodes that are assigned lower potential values than node i. If node i has no neighbor node with lower potential, that is, $|\mathcal{N}_l(i)| = 0$, the data packet is not sent to any node and is dropped, but such cases generally happen only in transient cases, such as node failures or changes of potential values at the sink node.

3. POTENTIAL-BASED ROUTING WITH OPTIMAL FEEDBACK

3.1. Overview

In this section, we describe a model of network dynamics and explain our optimal control scheme. A *controller* monitors network information, in particular the potential values of a partial set of nodes, which we call *observable nodes*. The controller then returns suitable control inputs to a partial set of nodes, which we call *controllable nodes*), for accelerating convergence of the potential distribution toward the target potential distribution. In this paper, target potentials are estimates of converged potential values derived from current information, specified in Subsection 3.2. We assume that the controller and sink nodes are power-supplied so that these sink nodes can have direct reliable connections to sensor nodes and the controller at all times. Therefore, in our proposal, the controller monitors network information and provides control inputs via them, as illustrated in Figure 3.

In our proposal, the area over which the controller monitors potential values is limited to several hops from sink nodes for reducing communication overhead. Of

course, the controller cannot directly get node potentials outside the area, but it can estimate them by utilizing the potential dynamics model, which describes potential changes based on local node interactions. When receiving control inputs from the controller, sink nodes diffuse the information, changing potential amounts over the entire network through local interactions of sensor nodes (Equation (2) in Subsection 3.2). This estimation of the controller requires high computational cost and the reduced-order model can reduce this computational cost for estimating potential values of non-observable nodes, as will be explained in Section 4. Note that information of the network topology is needed for designing a controller, as is flow rates of nodes for calculating target potential values. Such information is difficult to estimate, but is reported to the controller only when it changes, because we assume that changing intervals of the network topology and flow rates are lower than the convergence time of potentials. This assumption is feasible because the potential convergence is generally achieved as a result of the iterative behavior (nodes' potential updates and the controller's feedback) in potential-based routing with an optimal feedback so that frequencies of potential updates and controls need to be much higher than those of changes in the network topology and flow rates.

3.2. Network Dynamics

Let the dynamics of potentials be given by a deterministic discrete-time model. In our proposal we represent $f_n(t) = \bar{f}_n + d_n(t)$, where $\bar{f}_n$ is the stationary flow rate and d_n is non-stationary fluctuation (white Gaussian noise) at node n. Then, as in [Sheikhattar and Kalantari 2013], each potential is updated by

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

where η_n represents feedback input received from the controller. If node n does not receive any feedback input directly from the controller, then $\eta_n(t) = 0$. In our proposal, the controller collects and estimates the node potentials, and provides to the network feedback inputs $\boldsymbol{u}(t) = [\eta_1(t)\ \eta_2(t)\ \cdots\ \eta_{N_{ctrl}}(t)]^T$ (where N_{ctrl} denotes the number of nodes that receive feedback from the controller), as described later. Because of feedback inputs, we consider that node potentials can converge faster than in the non-control scheme (Equation (1)) where each node updates its potential based only on local interactions with neighbors. In [Sheikhattar and Kalantari 2013], σ_n is set to $\sigma_0/|\mathcal{N}_b(n)|$, but this value may lead to oscillation of potentials in some situations since Equation (9) (after-mentioned) has no solution. In this paper we therefore set σ_n to the constant value σ for all n ($n \in \{1, 2, \cdots, N\}$). Now we define $\boldsymbol{\theta}_i(t)$ as $[\theta_i(t)\ \theta_i(t+1)]$, and Equation (2) is rewritten by

$$
\boldsymbol{\theta}_n(t+1) = \boldsymbol{A}_1\boldsymbol{\theta}_n(t) + \boldsymbol{A}_0 \left(\sum_{k \in \mathcal{N}_b(n)} \{\boldsymbol{\theta}_k(t) - \boldsymbol{\theta}_n(t)\} + \boldsymbol{M}\left(\bar{f}_n + d_n(t)\right) \right) + \boldsymbol{M}\eta_n(t), \tag{3}
$$

where

$$
\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 \\ -\alpha & \alpha+1 \end{bmatrix}, \boldsymbol{A}_0 = \begin{bmatrix} 0 & 0 \\ 0 & \beta\sigma \end{bmatrix}, \boldsymbol{M} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.
$$

In our proposal, the controller estimates potentials of unobservable nodes and future potentials, and determines control inputs with H^∞ optimization (described below),
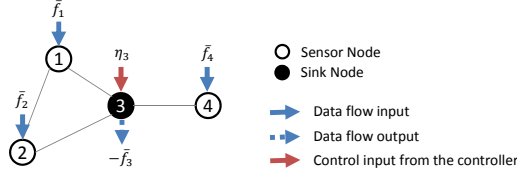
Fig. 4. An example of a network with 3 sensor nodes and a sink node. $\bar{f}_1$, $\bar{f}_2$, $\bar{f}_3$ and $\bar{f}_4$ are stationary flow rates for each node. η_3 is the control input to node 3 from the controller (the other nodes are not directly controlled by the controller, and η_1, η_2 and η_4 is 0 at all times). d_1, d_2, d_3 and d_4 are non-stationary fluctuations at each node although they are not described in this figure.

allowing us to describe the system dynamics in the form of the state space model. We first formulate the potential dynamics of all nodes. For example, we consider the case of a network of 4 nodes as shown in Figure 4. From Equation (3), the potential dynamics of node 1 is given by

$$\boldsymbol{\theta}_1(t+1) = \boldsymbol{A}_1\boldsymbol{\theta}_1(t) + \boldsymbol{A}_0\left(\sum_{k\in\{2,3\}}\{\boldsymbol{\theta}_k(t) - \boldsymbol{\theta}_1(t)\} + \boldsymbol{M}\left(\bar{f}_1 + d_1(t)\right)\right). \qquad (4)$$

Similarly to Equation (4), the potential dynamics of node 2, 3 and 4 are given, respectively, by:

$$\boldsymbol{\theta}_2(t+1) = \boldsymbol{A}_1\boldsymbol{\theta}_2(t) + \boldsymbol{A}_0\left(\sum_{k\in\{1,3\}}\{\boldsymbol{\theta}_k(t) - \boldsymbol{\theta}_2(t)\} + \boldsymbol{M}\left(\bar{f}_2 + d_2(t)\right)\right), \qquad (5)$$

$$\boldsymbol{\theta}_3(t+1) = \boldsymbol{A}_1\boldsymbol{\theta}_3(t) + \boldsymbol{A}_0\left(\sum_{k\in\{1,2,4\}}\{\boldsymbol{\theta}_k(t) - \boldsymbol{\theta}_3(t)\} + \boldsymbol{M}\left(\bar{f}_3 + d_3(t)\right)\right) + \boldsymbol{M}\eta_3(t), \ (6)$$

$$\boldsymbol{\theta}_4(t+1) = \boldsymbol{A}_1\boldsymbol{\theta}_4(t) + \boldsymbol{A}_0\left(\sum_{k\in\{3\}}\{\boldsymbol{\theta}_k(t) - \boldsymbol{\theta}_4(t)\} + \boldsymbol{M}\left(\bar{f}_4 + d_4(t)\right)\right). \qquad (7)$$

From Equations (4)–(7), the potential dynamics of the network shown in Figure 4 is given by

$$\begin{bmatrix}\boldsymbol{\theta}_1(t+1)\\ \boldsymbol{\theta}_2(t+1)\\ \boldsymbol{\theta}_3(t+1)\\ \boldsymbol{\theta}_4(t+1)\end{bmatrix} = \left(\boldsymbol{I}_{4\times4}\otimes\boldsymbol{A}_1 - \begin{bmatrix}2 & -1 & -1 & 0\\ -1 & 2 & -1 & 0\\ -1 & -1 & 3 & -1\\ 0 & 0 & -1 & 1\end{bmatrix}\otimes\boldsymbol{A}_0\right)\begin{bmatrix}\boldsymbol{\theta}_1(t)\\ \boldsymbol{\theta}_2(t)\\ \boldsymbol{\theta}_3(t)\\ \boldsymbol{\theta}_4(t)\end{bmatrix}$$

$$+ \left(\beta\sigma\left(\begin{bmatrix}f_1\\ f_2\\ f_3\\ f_4\end{bmatrix} + \begin{bmatrix}d_1\\ d_2\\ d_3\\ d_4\end{bmatrix}\right) + \begin{bmatrix}0\\ 0\\ \eta_3\\ 0\end{bmatrix}\right)\otimes\boldsymbol{M}, \qquad (8)$$

where $\boldsymbol{I}_{N\times N}$ is the identity matrix of $N\times N$. The matrix $\begin{bmatrix}2 & -1 & -1 & 0\\ -1 & 2 & -1 & 0\\ -1 & -1 & 3 & -1\\ 0 & 0 & -1 & 1\end{bmatrix}$ corresponds to the graph Laplacian, which indicates the network topology. We refer to the graph Laplacian as $\boldsymbol{\Gamma}$ in this paper. The element l_{ij} of graph Laplacian $\boldsymbol{\Gamma}$ is given by

$$l_{ij} = \begin{cases} \deg(i), & \text{if } i = j \\ -1, & \text{if } i \neq j \wedge \text{node } i \text{ is adjacent to node } j \\ 0, & \text{otherwise} \end{cases},$$

where $\deg(i)$ corresponds to the degree of node i. The operation $\otimes$ that is called the Kronecker product is an operation on two matrices. Given an $m \times n$ matrix $\boldsymbol{J}$ and a $p \times q$ matrix $\boldsymbol{K}$, the Kronecker product $\boldsymbol{J} \otimes \boldsymbol{K}$ is given by

$$\boldsymbol{J} \otimes \boldsymbol{K} = \begin{bmatrix} j_{11}\boldsymbol{K} & \cdots & j_{1n}\boldsymbol{K} \\ \vdots & & \vdots \\ j_{m1}\boldsymbol{K} & \cdots & j_{mn}\boldsymbol{K} \end{bmatrix}.$$

Then, we define a vector $\boldsymbol{\Theta}(t)$ that shows potential values for all nodes as

$$\boldsymbol{\Theta}(t) := [\boldsymbol{\theta}_1(t) \, \boldsymbol{\theta}_2(t) \, \cdots \, \boldsymbol{\theta}_N(t)]^T.$$

As in Equation (8), the potential dynamics of all nodes is described with $\boldsymbol{\Theta}(t)$ as

$$\boldsymbol{\Theta}(t+1) = \boldsymbol{A}\boldsymbol{\Theta}(t) + (\beta\sigma\,(\boldsymbol{F} + \boldsymbol{D}(t)) + \boldsymbol{E}\boldsymbol{u}(t)) \otimes \boldsymbol{M},$$

where

$$\boldsymbol{A} = \boldsymbol{I}_{N \times N} \otimes \boldsymbol{A}_1 - \boldsymbol{\Gamma} \otimes \boldsymbol{A}_0.$$

The flow rate vector $\boldsymbol{F}$ is defined as $\begin{bmatrix} \bar{f}_1 & \cdots & \bar{f}_N \end{bmatrix}^T$, and the fluctuation rate vector $\boldsymbol{D}(t)$ is defined as $[d_1(t) \, \cdots \, d_N(t)]^T$. Note that the $(N \times N_{ctrl})$-matrix $\boldsymbol{E}$ specifies the controllable node, that is, the element $e_{ij} \in \{0,1\}$ of $\boldsymbol{E}$ is 1 if and only if node i receives the j-th element of $\boldsymbol{u}(t)$ as control input $\eta_i(t)$ (in the case with the network shown in Figure 4, $\boldsymbol{E}$ is set to $[0\ 0\ 1\ 0]^T$). With the larger number of controllable nodes, i.e., with denser $\boldsymbol{E}$, the influence of the optimal control is larger so that the convergence speed is faster. On the contrary, significant properties originating from self-organization such as scalability and adaptability could be lost with too many controllable nodes.

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

$$(\boldsymbol{I}_{2N \times 2N} - \boldsymbol{A})\bar{\boldsymbol{\Theta}} = \beta\sigma\boldsymbol{F} \otimes \boldsymbol{M}. \tag{9}$$

Then, we define $\boldsymbol{X}(t)$ as the regulation error $\bar{\boldsymbol{\Theta}} - \boldsymbol{\Theta}(t)$, which shows the differences between the convergence values and the current values of potentials for each node. The potential dynamics can be rewritten using $\boldsymbol{X}(t)$ in the form of the state space representation by Equation (10), with which the controller can estimate potentials of unobservable nodes and future potentials.

$$\boldsymbol{X}(t+1) = \boldsymbol{A}\boldsymbol{X}(t) + \boldsymbol{B}_1\boldsymbol{d}(t) + \boldsymbol{B}_2\boldsymbol{u}(t). \tag{10}$$

Here, the N_{ctrl}-dim vector $\boldsymbol{u}$ (resp. N-dim vector $\boldsymbol{d}$) concatenates $\eta_n(t)$ for controllable nodes (resp. d_n). $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ characterizes the effect of noises and control inputs, respectively, given by

$$\boldsymbol{B}_1 = \boldsymbol{B}_1' \otimes \begin{bmatrix} 0 \\ \beta\sigma \end{bmatrix}, \boldsymbol{B}_2 = \boldsymbol{E} \otimes \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

$\boldsymbol{B}_1'$ characterizes the variance/co-variance of the noise distribution injected at each node. For simplicity, we choose $\boldsymbol{B}_1' = \boldsymbol{I}_{N \times N}$ in this paper, which implies each node has independent noise with unit variance. Note that the dynamics (Equation (10)) depends on the graph Laplacian $\boldsymbol{\Gamma}$, but not on the flow rate $\bar{f}$.

3.3. Optimal Controller Design

We next explain the controller dynamics. We consider the case where the controller can observe

$$\boldsymbol{Y}(t) = (\boldsymbol{H}^T \otimes \boldsymbol{I}_{2\times2})\boldsymbol{X}(t)$$
$$= (\boldsymbol{H}^T \otimes \boldsymbol{I}_{2\times2})(\bar{\boldsymbol{\Theta}} - \boldsymbol{\Theta}(t)), \tag{11}$$

where $\boldsymbol{Y}(t)$ is a $2N_{obs}$-dim vector, with N_{obs} being the number of observable nodes, and an $(N \times N_{obs})$ matrix $\boldsymbol{H}$ determines observable nodes in the same manner as $\boldsymbol{E}$ (if the controller can only observe node 3 in the network shown in Figure 4, $\boldsymbol{H}$ is set to $[0\ 0\ 1\ 0]^T$). Note that the controller can estimate the potential values of all nodes based on the information of only a partial set of nodes, and therefore, not all elements of $\boldsymbol{H}$ need to be set to 1. If more nodes are monitored by the controller, it can estimate potential values of all nodes more precisely while the communication overhead for collecting information of observable nodes becomes much larger. The future node potentials can be estimated by Equation (10) using observable information expressed with Equation (11). Considering the future potentials, the controller calculates a $\boldsymbol{u}(t)$ that accelerates the convergence speed of potentials.

Then, the control input is calculated according to

$$\tilde{\boldsymbol{X}}(t+1) = \boldsymbol{A}_c\tilde{\boldsymbol{X}}(t) + \boldsymbol{B}_c\boldsymbol{Y}(t), \tag{12}$$
$$\boldsymbol{u}(t) = \boldsymbol{C}_c\tilde{\boldsymbol{X}}(t) + \boldsymbol{D}_c\boldsymbol{Y}(t), \tag{13}$$

where $2N$-dim vector $\tilde{\boldsymbol{X}}$ is an internal model for the controller and the controller estimates potentials of all nodes with the model. $\boldsymbol{A}_c$, $\boldsymbol{B}_c$, $\boldsymbol{C}_c$, 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 trade-off between convergence speed and input energy. With a larger r, control inputs become smaller and the stability of the system is enhanced. Namely, potentials change more gently, whereas the convergence speed of potentials becomes 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]. Given $\boldsymbol{A}$, $\boldsymbol{B}_1$, $\boldsymbol{B}_2$ and $\boldsymbol{H}$, it is known that the optimal $\boldsymbol{A}_c$, $\boldsymbol{B}_c$, $\boldsymbol{C}_c$, $\boldsymbol{D}_c$ can be obtained based on semi-definite programming (see also Section 5). In general, the degree of the optimal controller, that is the size of $\tilde{\boldsymbol{X}}$, is the same as that of the system to be controlled, which is $2N$ in this case. Note that we need to solve this optimization problem only when the connection topology changes. In other words, we only have to update $\tilde{\boldsymbol{X}}$ in Equation (12) and compute Equation (13) so long as the connection topology does not change.

In [Kuze et al. 2014], we previously proposed potential-based routing with an optimal control that is explained in this section, but we only evaluate the situation where the controller observes all nodes, i.e., $N_{obs} = N$ and $\boldsymbol{H} = \boldsymbol{I}_{N\times N}$. The upper bound of the order of the communication cost for observation is $O(NP_{avg})$ where P_{avg} corresponds to the average path length of the network. Note that the average path length of the network generally depends on the number N of nodes. For example, the average path length of Erdös-Rényi network (a random network) [Erdős and Rényi 1961] and

Barabási-Albert model (a scale-free network) [Albert and Barabási 2002] is $O(\log N)$, and that of Watts-Strogatz network (a small-world network) [Watts and Strogatz 1998] is $O(N)$ if $N < N^*$ and $O(\log N)$ if $N > N^{*1}$. Therefore, in this paper, we show that the controller can estimate potentials of non-observable nodes based on only a part of nodes' potentials with simulation evaluations in Section 5. Note that, if the average path length is quite small against the network size as in the case with WWW, the external controller can observe a large number of nodes with small communication cost, which allows the external controller to estimate the potential dynamics more correctly.

Moreover, as the number of nodes increases, that is, the size of vector $X(t)$ becomes large, it becomes unfeasible to control the system using the model expressed in Equation (10) and Equation (11), owing to the considerable computational cost. The design parameters (A_c, B_c, C_c, and D_c) are calculated with $O(N^3)$ because the computational cost is proportion to the cube of the size of X [Gahinet and Apkarian 1994]. Control inputs ($u(t)$) are calculated with $O(N^2)$ because they are calculated by Equations (12) and (13). We therefore approximate the model with a reduced-order model for reducing the computational cost.

4. MODEL REDUCTION

In our proposal, the controller uses a reduced-order model for estimating network dynamics to reduce the computational cost. In this section, we explain the model reduction of the network dynamics model described in Section 3.

In the reduced-order model, the dynamics of the system is expressed as an $(h \times 1)$-vector $X_r(t)$ whose elements are linear transformations of the original model $X(t)$, and the reduced-order model is given by Equations (14) and (15).

$$X_r(t+1) = A_r X_r(t) + B_{r1} d(t) + B_{r2} u_r(t), \tag{14}$$
$$Y_r(t) = C_r X_r(t) + D_r d(t). \tag{15}$$

Here, X_r is an h-dim vector with h ($< 2N$) being the order of the approximate model, and $u_r(t)$ corresponds to control inputs provided by the controller with the reduced-order model. We need to choose matrices A_r, B_{r1}, B_{r2}, C_r, and D_r of compatible dimensions such that $Y_r(t) \approx Y(t)$ for all input $u_r(t)$, $d_r(t)$. Many researchers have studied a variety of methods to approximate a model with a reduced-order model to control large, complex systems [Antoulas et al. 2006]. In our proposal, we approximate the original model based on a '*balanced realization*' that is highly compatible with the model expressed in the state space representation [Zhou et al. 1995; Antoulas et al. 2006]. In model reduction, a reduced-order model needs to have the same response characteristics as the original model for the accurate estimation of potentials. Here, response characteristics mean the effectiveness of inputs to the system.

The control input is then calculated in the same manner as $u_r(t)$ according to

$$\tilde{X}_r(t+1) = A_{cr} \tilde{X}_r(t) + B_{cr} Y_r(t), \tag{16}$$
$$u_r(t) = C_{cr} \tilde{X}_r(t) + D_{cr} Y_r(t), \tag{17}$$

where A_{cr}, B_{cr}, C_{cr}, and D_{cr} are design parameters. Now, we can use $Y(t)$ for $Y_r(t)$ in Equation (17) under the well-designed C_r, and D_r because of the relation of $Y_r(t) \approx Y(t)$. We similarly obtain the H^∞ optimal controller of order h for the reduced system. These parameters determine how potential values observed by the controller affect feedback inputs.

---

[1] N^* is a length which depends on the randomness p of a small-world network. See [Albert and Barabási 2002] for details.

A reduced-order model can describe the system dynamics with a constant number h of state variables and the computational cost can be reduced using it. With a reduced-order model, the design parameters (A_{cr}, B_{cr}, C_{cr}, and D_{cr}) are calculated with $O(h^3)$, and control inputs ($u_r(t)$) are calculated with $O(h^2)$. In general, a model that has more state variables lets the controller estimate more correctly, but the computational cost is larger. In contrast, the computational cost is smaller but the estimation error can be larger in a model that has fewer state variables. Therefore, h needs to be properly determined in accordance with the requirements or system properties (see also Section 5).

In Section 5, we conduct a network simulation to reveal the packet-level dynamics and to show that Equations (2)–(17) assuming synchrony of nodes can be adopted to asynchronous network systems.

5. PERFORMANCE EVALUATION

In this section, we evaluate our proposal to show that the convergence speed is enhanced by introducing optimal feedback using the reduced-order model. We conduct computer simulation and evaluate the convergence speed comparing our proposal (potential-based routing with optimal feedback (PBR-opt) and one using the model reduction (PBR-opt-mr)) with the non-control scheme proposed in [Sheikhattar and Kalantari 2013]. First, in Subsection 5.2, we evaluate the potential convergence speed after traffic changes to show that our proposed methods (PBR-opt and PBR-opt-mr) enhance the convergence speed of potentials. Moreover, we show that PBR-opt-mr can be adapted to large-scale networks in Subsection 5.3. Next, in Subsection 5.4 for demonstrating the robustness of our proposal, we consider a changing environment where several nodes fail but the controller does not collect network topology information, and thus cannot detect these failures.

In conducting simulation experiments, for the network simulator we use an event-driven packet-level simulator written in Visual C++ that calls MATLAB functions *dhinflmi*[2] to design the optimal controller and *balred*[3] to obtain a reduced-order model on a 64 bit operating system with Intel(R) Xeon(R) CPU with 2.70 GHz, and 64.0 GB memory. The simulator is of our own making and, 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], an asynchronous receive-drive data transmission protocol, in the MAC layer. Note that this setting does not mean that our proposal depends on specific MAC layer protocols such as IRDT. In the physical layer, we use a simple disk model for wireless communication, where the packet reception between two nodes is successful if two nodes are within the pre-defined communication range. Also we use a simple packet collision model where a receiver node always drops a packet if other packets arrives during the reception of the packet. In the simulator, the asynchrony of components, i.e., the controller and all nodes, is implemented as follows. The controller and all nodes do not match their timing to provide feedback and to update their potentials. The asynchronous behavior of all nodes is implemented on a single thread with an event queue in our simulator. Note that we set the interval of the control feedback by the controller

---

[2]*dhinflmi* is a function for designing the H^∞ optimal controller [Gahinet and Apkarian 1994] of a discrete system. Given parameter matrices A, B_1, B_2, C, D_1 and D_2 for the system dynamics $X(t+1) = AX(t) + B_1 D(t) + B_2 u(t)$ and the information $Y(t) = CX(t) + D_1 D(t) + D_2 u(t)$ that is observed by the controller, *dhinflmi* computes design parameters A_c, B_c, C_c and D_c that are referred in Equations (12), (13.) The matrix $D(t)$ shows non-stability fluctuations, $u(t)$ are control inputs, A, B_1, B_2, C, D_1, D_2

[3]*balred* is a function for reducing the model order [Varga 1991]. Given design parameter matrices A, B_1, B_2, C, D_1 for the original model and the degree h of a reduced-order model, *balred* computes a hth-order approximation of the original model.

and the intervals of potential updates in each node to be equal in accordance with Equations (10) and (14) so that the controller can estimate potentials of nodes with small errors.

5.1. Simulation Settings

We evaluate changes of potentials in potential-based routing and the number of data packets delivered to each sink node after traffic changes or node failures. Network models used in evaluations consist of an external controller and sensor/sink nodes. Only sink nodes are directly connected to the controller, so the controller monitors network states via sink nodes and sends suitable control inputs to sink nodes. The controller provides feedback to each sink node at interval T_f, while at interval T_p, each node updates its next potential value. Typically, $T_f = T_p$ for matching with the potential dynamics described by Equation (2). In real situations, it is difficult to monitor up-to-date potential values of all nodes in the network because of the overhead for collecting potential values, especially when the number of nodes increases. Therefore, here we consider that the controller can directly monitor only nodes within p hops from the sink nodes. That is, nodes within p hops from the sink nodes send a control message to the sink nodes at interval T_i to notify the controller of their potential values. Otherwise, the controller does not monitor node potentials outside p hops from sink nodes, and only estimates them using the model (Equation (14)), which describes potential dynamics through local interactions among nodes. The interval of control message emission T_i is set to 50 s, p is set to 2, and the i-th element h_i of $\boldsymbol{H}$ is set to

$$h_i = \begin{cases} 1, & \text{if node } i \text{ is within } p \text{ hops from sink nodes} \\ 0, & \text{otherwise} \end{cases}.$$

At the beginning of the simulation, potential values of all nodes, including sensor nodes and sink nodes, are initialized to 0. During the first $1,000$ s, each node exchanges its potential values with neighbor nodes and updates its potential value at interval T_p according to Equation (2) so that the potential values are stabilized. For this, we do not generate data packets during that time duration. At $1,000$ s data packets begin to be generated at sensor nodes according to the Poisson process with their flow rate.

We evaluate the convergence speed of potentials and data packets delivered to each sink node after traffic changes. To measure the convergence speed of potentials, we define the degree of the potential convergence $\epsilon_n(t)$ for each node that is given by

$$\epsilon_n(t) = \frac{|\bar{\theta}_n - \theta_n(t)|}{|\bar{\theta}_n|},$$

where $\bar{\theta}_n$ corresponds to the target potential value of node n. We consider convergence to be achieved when $\epsilon_n(t)$ for all nodes becomes sufficiently small. Convergence time is defined as the minimum time taken by all sensor and sink nodes to satisfy the condition

$$\epsilon_n(t) < c, \tag{18}$$

where c (≥ 0) is a constant value. In an ideal situation where all nodes are completely synchronized and there is no noise, all sensor and sink nodes satisfy Equation (18) in the end even if $c = 0$, but in an actual situation, not all nodes can satisfy Equation (18). The larger the value of c is, the shorter the time needed for the achievement of the potential convergence is. Therefore, c needs to be set carefully according to the purpose of the evaluation. In this paper, we set c to 0.2 in order to evaluate the convergence speed strictly.

Table I. Parameter settings

Parameter	Value
α	0.4
β	0.2
σ	0.1
r	10
T_f	50 s
T_p	50 s

Energy efficiency is a significant challenging task in wireless sensor networks. With respect to our proposal, the computational cost of each sensor node is fairly low because each sensor node behaves based only on local information and simple rules. This is an inherent characteristic of self-organizing systems. Moreover, our proposal can achieve load balancing by setting flow matrix F so that each sink node receives data packets equally, which we explain in Subsection 5.2. Finally, our proposal does not depend on specific MAC layer protocols. Therefore, we can reduce energy for data transmission between adjacent nodes by introducing energy-efficient MAC layer protocols such as IRDT that we use in this evaluation.

Parameters are summarized in Table I. Optimal parameters $(\alpha, \beta\sigma)$ for PBR-no-ctrl are given in [Sheikhattar and Kalantari 2013] [4], but potentials of nodes diverge to infinity and do not converge. Therefore, we use original settings shown in Table I even for PBR-no-ctrl. All results presented below are averaged over 40 simulation runs for each parameter setting.

5.2. Performance evaluation of reduced order controller

Here, we evaluate the convergence speed considering constraints in wireless sensor networks.

Figure 5 shows the network model with 104 nodes (including four sink nodes) used for this evaluation. Sink nodes are illustrated with squares and sensor nodes with dots in that figure. In this network model described in Figure 5, every sensor node is within 3 hops from a sink node. The number of sensor nodes within 2 hops from a sink node is 86, that is, the external controller observes potentials of 86 sensor nodes in this network. In this evaluation, at 10,000 s from the beginning of the simulation, data packet generation rates of sensor nodes are changed to examine the convergence speed of our method. Data packet generation rates are initially set to be 0.0005 packets/s for sensor nodes in the left half of Figure 5 (illustrated with black dots), and 0.0015 packets/s for sensor nodes in the right half (illustrated with gray dots). After traffic changes at 10,000 s, data packet generation rates are increased to 0.0015 packets/s for the left half sensor nodes and decreased to 0.0005 packets/s for the right half nodes. The average data generation rate of a node of 0.001 packets/s corresponds to $\bar{f}_n = 1$, so before traffic changes the flow rate vector $F = \begin{bmatrix} \bar{f}_1 & \cdots & \bar{f}_N \end{bmatrix}^T$ is given by

$$\bar{f}_i = \begin{cases} 0.5, & \text{if } i \in \mathcal{N}_{sen_l} \\ 1.5, & \text{if } i \in \mathcal{N}_{sen_r} \\ -\frac{|\mathcal{N}_{sen}|}{|\mathcal{N}_{sin}|}, & \text{if } i \in \mathcal{N}_{sin} \end{cases}, \tag{19}$$

[4]Optimal α and $\beta\sigma$ are respectively given by $\frac{1-\xi}{1+\xi}$ and $\frac{\alpha+1}{2}$ for the fastest convergence rate of potentials, where $\xi = \sqrt{1-\nu_1^2}$. ν_1 is the spectrum radius of the matrix $S = I_{N \times N} - G^{-1}\Gamma$ where G is the degree matrix and Γ is the graph Laplacian of the network. The spectrum radius of the matrix S is defined as the max value among absolute values of eigenvalues of S. See [Sheikhattar and Kalantari 2013] for more details.
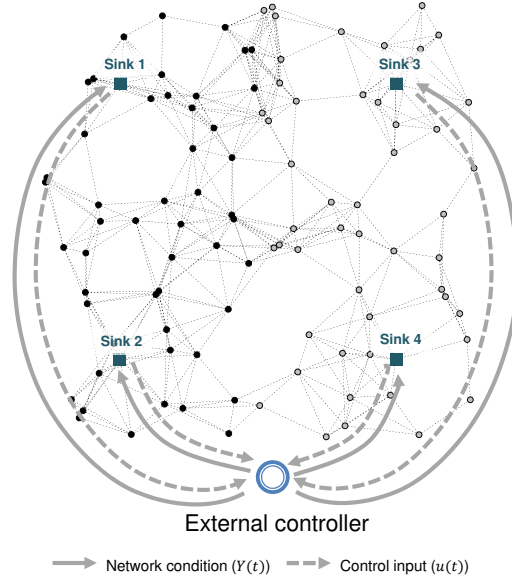
Fig. 5.   Network topology ($N = 104$)

where $\mathcal{N}_{sen}$ corresponds to the set of sensor nodes, and $\mathcal{N}_{sin}$ corresponds to that of sink nodes. $\mathcal{N}_{sen_l}$ is the set of sensor nodes in the left half of Figure 5 and $\mathcal{N}_{sen_r}$ is that of sensor nodes in the right half. Note that we construct the potential fields such that all sink nodes can receive data packets equally because load balancing is a challenging task in wireless sensor networks. Thus, the flow rate at each sink node is ideally given by $|\mathcal{N}_{sen}|/|\mathcal{N}_{sin}|\ (= 25)$. Similarly, after the traffic changes, the i-th element of $\boldsymbol{F}$ is given by

$$\bar{f}_i = \begin{cases} 1.5, & \text{if } i \in \mathcal{N}_{sen_l} \\ 0.5, & \text{if } i \in \mathcal{N}_{sen_r} \\ -\frac{|\mathcal{N}_{sen}|}{|\mathcal{N}_{sin}|}, & \text{if } i \in \mathcal{N}_{sin} \end{cases}. \tag{20}$$

In this evaluation, we conduct simulation in the case where flows of data packets massively change the network as mentioned above in order to show that our proposal can adapt to massive environmental changes. A system that can adapt to massive environmental changes is considered to be able to adapt to minor fluctuations. Moreover, we assume that the data generation rate does not change until $10,000$ s from the beginning of the simulation in order to compare our proposal and the non-control scheme under the same conditions.

In the evaluation of PBR-opt-mr, degree h of the reduced order model is set to 3 and 100. As explained in Section 4, with lower h, the computational cost is lower but the estimation error is larger, which results in the slow convergence speed of potentials. Therefore, we evaluate not only in the case with low $h\ (= 3)$ but also in the case with high $h\ (= 100)$ for clarifying how the degree of the reduced order model effects potential convergence.

In the evaluation of PBR-opt, the design parameters ($\boldsymbol{A}_c, \boldsymbol{B}_c, \boldsymbol{C}_c, \boldsymbol{D}_c$) cannot be calculated with the *dhinflmi* function when $N = 104$, i.e., the size of the model is 208, due to the considerable computational cost. This motivates us to reduce the computational burden upon controller parameter design and controller state updates. For compari-
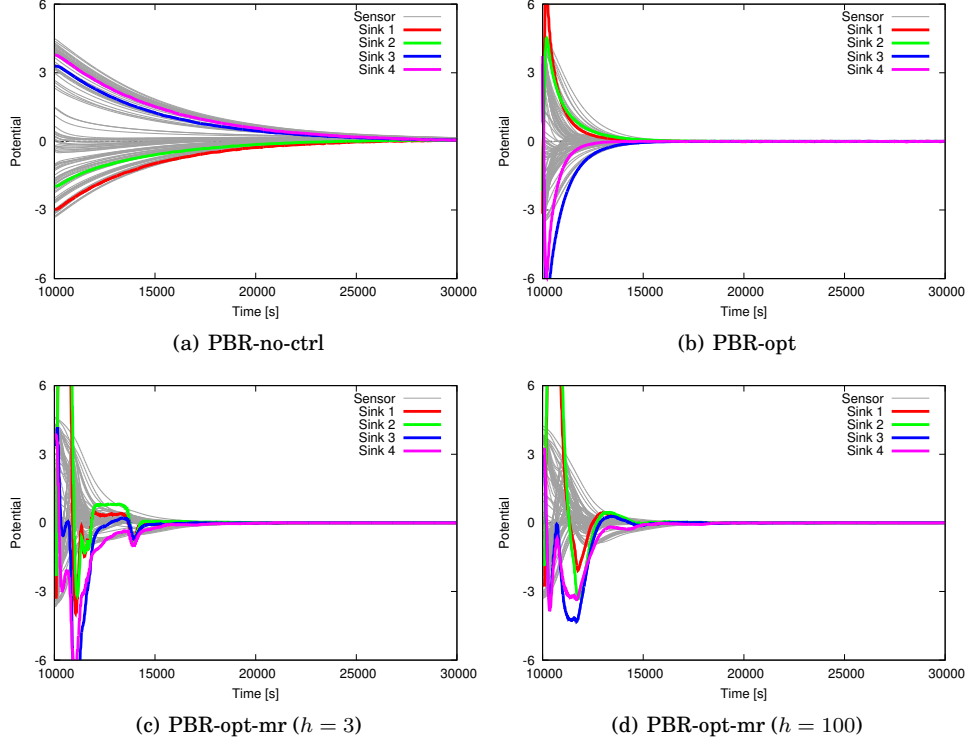
(a) PBR-no-ctrl

(b) PBR-opt

(c) PBR-opt-mr ($h = 3$)

(d) PBR-opt-mr ($h = 100$)

Fig. 6.   Potential convergence in case of traffic changes

Table II. Potential convergence time

Scheme	Convergence time [s]
PBR-no-ctrl	70,417
PBR-opt	9,174
PBR-opt-mr ($h = 3$)	11,765
PBR-opt-mr ($h = 100$)	9,313

son, we utilize an optimal static controller designed by the *dlqr* function[5], where the controller is assumed able to monitor the latest conditions of all nodes with no communication delay or control overhead (i.e., $h_i = 1$ for all nodes). It is known that static controllers can achieve satisfactory performance with such state feedback settings [Zhou et al. 1995].

Figure 6 shows changes of potential values of the non-control scheme, PBR-opt, and PBR-opt-mr with $h = 3$, and 100. More exactly, Figure 6 plots $\boldsymbol{X}(t) = \bar{\boldsymbol{\Theta}} - \boldsymbol{\Theta}(t)$ against time t, and the potential convergence is achieved when each element of $\boldsymbol{X}(t)$ is sufficiently close to 0, that is, Equation (18) is satisfied. In the figure, thick lines correspond to potential changes of the two sink nodes, and thin lines correspond to those of other

---

[5]The function *dlqr* is linear-quadratic (LQ) state-feedback regulator for discrete-time state space system. Given parameter matrices $\boldsymbol{A}$, $\boldsymbol{B}$, q and r for the system dynamics $\boldsymbol{X}(t+1) = \boldsymbol{X}(t) + \boldsymbol{B}\boldsymbol{u}(t)$, *dlqr* computes $\boldsymbol{K}$, $\boldsymbol{S}$ and e. Then, control inputs $\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{X}(t)$ using the optimal gain matrix $\boldsymbol{K}$ minimizes the quandratic cost function $J(u) = \sum_{n=1}^{\infty} (q||\boldsymbol{X}(n)|| + r||\boldsymbol{u}(n)||)$ for the system. Note that the controller needs to observe all nodes for calculating control inputs by $\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{X}(t)$ although the computational cost for calculating $\boldsymbol{K}$ with *dlqr* function is much lower than that for calculating $\boldsymbol{A}_c$, $\boldsymbol{B}_c$, $\boldsymbol{C}_c$ and $\boldsymbol{D}_c$ with *dhinflmi* function.

(a) PBR-no-ctrl

(b) PBR-opt

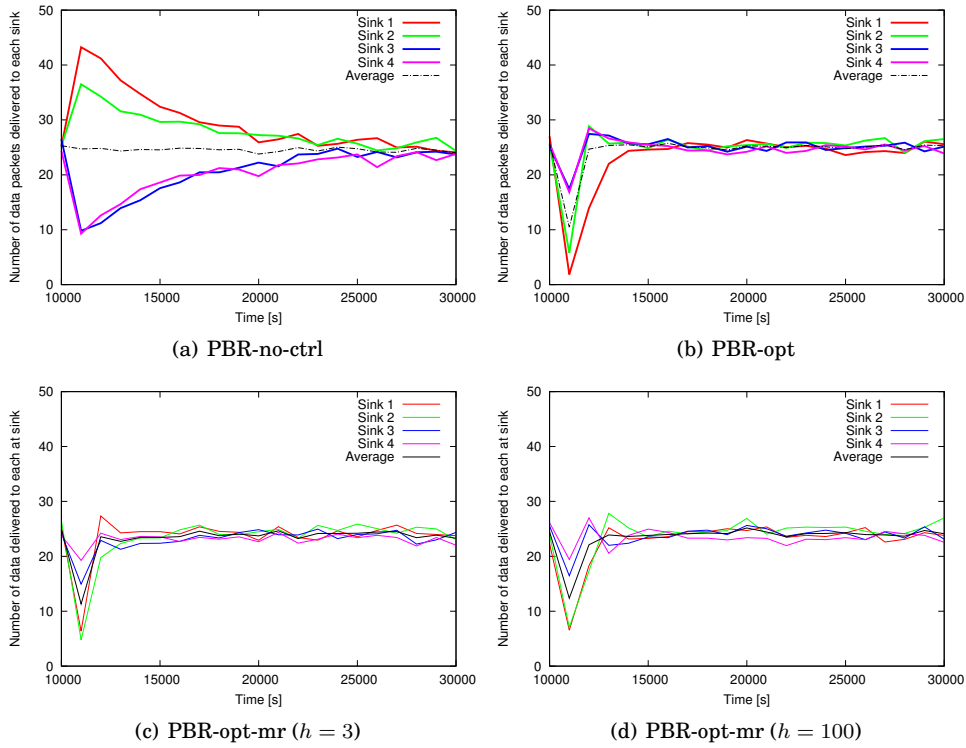(c) PBR-opt-mr ($h = 3$)

(d) PBR-opt-mr ($h = 100$)

Fig. 7.   Data packets delivered to each sink node in case of traffic changes

sensor nodes. Sink node potentials change more than those of sensor nodes, because sink nodes receive feedback inputs u directly from the controller. Feedback inputs provided to each sink node are different, so the potential change of each sink node varies. On the other hand, sensor nodes are indirectly affected by feedback inputs via sink nodes through potential updates by Equation (2). As shown in the figure, if the potential value of a sink node increases (resp. decreases) due to feedback inputs from the controller, potential values of sensor nodes near the sink node also increase (resp. decrease).

As shown in Figure 6, our proposal can enhance the convergence speed of potentials, as compared with the case of no control scheme, and Table II shows the time needed from traffic changes until the potential convergence is achieved when $c = 0.2$. Figure 6 shows potential changes only within $10,000$–$30,000$ s but it takes $70,417$ s for the potential convergence, i.e., satisfying Equation (18), with PBR-no-ctrl as shown in Table II. As a result, potential convergence is accelerated by about 5.99 times and 7.56 times due to PBR-opt-mr with $h = 3$ and 100, respectively. Compared with PBR-opt, potential convergence speed is 0.780 times and 0.985 times using PBR-opt-mr with $h = 3$ and 100, respectively. These results indicate that if h is properly chosen, the convergence speed of our proposal can be approximately equal to that of PBR-opt (about 7.68 times as fast as the original). The convergence speed of PBR-opt-mr with $h = 3$ is a little bit slower than that of PBR-opt-mr with $h = 100$, because the smaller h is the larger the approximation error is. Specifically, if h is smaller than 3, potentials diverge and cannot converge to targeted potential values although we do not show the result in this paper. However, from the point of view of computational cost, when h is 3, PBR-

opt-mr can enhance the convergence speed with much lower computational cost than PBR-opt, because the cost of the design parameter calculation is reduced from $O(N^3)$ to $O(h^3)$ and the cost of the control input calculation is reduced from $O(N^2)$ to $O(h^2)$ by the model reduction. Note that the computational cost for designing the controller of PBR-opt using *dhinflmi* is too large to compute in a practical time as explained above. Therefore, PBR-opt cannot be adapt to large-scale networks. It is worth mentioning that the controller of PBR-opt-mr observes only node potentials within two hops of sink nodes (i.e., $p = 2$). The controller estimates potentials of non-observable and future potentials using Equation (14), then determines the optimal feedback inputs. This indicates that the controller does not need to collect information of the entire network to enhance the convergence speed of potentials. In general, it becomes more difficult for the controller to estimate potentials correctly as the number of nodes observed by the controller becomes smaller. We will investigate the trade-off between the number of observable nodes and the accuracy of the potential estimation as future work.

At about $14,000$ s in Figure 6(c), potentials of nodes rapidly fluctuate. This is because the controller cannot receive control messages, which collect potential information and inform the controller about them, due to packet drops so that the controller temporarily provides irrelevant control inputs. However, such fluctuations are temporary and potentials immediately converge again. Therefore, these fluctuations have little effect on transmission of data packets as shown in Figure 7(c) which is mentioned below. Moreover, in the case where the controller collects potential information via sensor nodes (i.e., not sink nodes), traffic congestion around sink nodes is reduced, and therefore, these fluctuations caused by packet drops will be reduced.

Estimation errors of potentials cannot be completely avoided in real networks, because nodes are not synchronized unless the potential dynamics described by Equation (2), and the potential values the controller collects are not always correct due to communication delays, dropped data, or interference. In this evaluation, traffic congestion occurs around sink nodes because the controller collects the network information via sink nodes, and 1.32%, 1.33% of the control messages for collecting potential values are dropped when using PBR-opt-mr with $h = 3, 100$ respectively, leading to estimation errors of potentials. Moreover, the asynchrony of the controller and nodes is also a cause of estimation errors because PBR-opt-mr (and also PBR-opt) inherently assumes synchronous systems. Nevertheless, our proposal can achieve fast convergence of potentials despite such errors, which clearly shows that our proposal works well in an asynchronous environment where noise or disturbance exists.

Figure 7 shows the average number of data packets delivered to each sink node every $1,000$ s. In each case, the number of data packets delivered to each sink node becomes disproportionate after the traffic changes at $10,000$ s. Then sink nodes gradually become able to receive data packets equally, because potentials are updated to adapt to the current packet rate. We can observe that the traffic convergence is also accelerated by optimal feedback. This is because the potential convergence speed is enhanced by the optimal feedback mechanism.

One problem we can find is that our proposal reduces the average number of data packets delivered to each sink node immediately after traffic changes. This is because some sink nodes temporarily have the largest potential values within their communication ranges according to the control inputs, so data packets cannot arrive at sink nodes. Therefore, a partial set of data packets would drop when the controller make large changes to the potentials, which contributes to the faster convergence speed of potentials. However, the data packet drops are immediately reduced and the traffic finally converges faster than the non-control scheme because of the faster potential convergence. Note that in an actual situation data packets may be retransmitted instantly. Here, we evaluate only the case where data packets are never retransmitted,

because the main purpose of this paper is to reveal the upper limit of convergence speed of self-organizing systems. Moreover, Figures 7(c) and 7(d) show worst case scenarios for temporary packet drops, because the controller changes sink node potentials (in other words, data packet destinations), so many data packets are dropped when a sink node temporarily gets the highest potential within its communication range due to control inputs. If the controller provides an optimal feedback to several sensor nodes where only some data packets arrive, the number of data packet drops will be smaller. With a lower r sink nodes are more likely to be assigned higher potential values, since the controller can make large changes to the potentials, whereas the recovery speed of data packets delivered to each sink node becomes faster. This indicates that there is a trade-off between the convergence speed of potentials and potential fluctuations.

It is also worth mentioning that the traffic convergence when using PBR-opt-mr with $h = 3$ is as fast as that with $h = 100$ as shown in Figures 7(c) and 7(d), although potentials oscillate more frequently until the potential convergence is achieved with $h = 3$ compared to the case of $h = 100$ (Figures 6(c) and 6(d)). This indicates that temporary oscillations of potential values because of approximation errors have little effect on data packet transmissions.

In this subsection, we have shown that an optimal control by the external controller is effective in wireless sensor networks where the capacity and the energy of each node are limited. Moreover, PBR-opt-mr enhanced the convergence speed of potentials with much lower computational cost than PBR-opt. That indicates that a reduced-order model reflects the dominant characteristics of the original model. It is also worth mentioning that even when some amount of approximation error exists, fast convergence of potentials can be achieved. In this evaluation, however, we have assumed that the controller has up-to-date topology information. Therefore, in Subsection 5.4 we conduct a simulation experiment where several nodes fail and the controller cannot detect the failures (i.e., the controller provides feedback inputs which lead potentials to converge to values that are not adequate to the network).

5.3. Scalability of reduced order controller

Next, we conduct a simulation on a network with 309 nodes including 9 sink nodes in order to reveal the scalability of our proposal in large-scale networks. Specifically, we prove that our proposal can facilitate convergence of a self-organizing system with light-weight cost even if the network size is large. Figure 8 shows the network model that is used in this evaluation. In this network model described in Figure 8, every sensor node is within 3 hops from a sink node. The number of sensor nodes within 2 hops from a sink node is 271, that is, the external controller observes potentials of 271 sensor nodes in this network. In this evaluation, at $10,000$ s from the beginning of the simulation, data packet generation rates of sensor nodes are changed to examine the convergence speed of our method similarly to the scenario of the evaluation in Subsection 5.2. Data packet generation rates are initially set to be 0.0005 packets/s for sensor nodes in the left half of Figure 8 (illustrated with black dots), and 0.0015 packets/s for sensor nodes in the right half (illustrated with gray dots). After traffic changes at $10,000$ s, data packet generation rates are increased to 0.0015 packets/s for the left half sensor nodes and decreased to 0.0005 packets/s for the right half nodes. The average data generation rate of a node of 0.001 packets/s corresponds to $\bar{f}_n = 1$, so before traffic changes the flow rate vector $\boldsymbol{F} = \begin{bmatrix} \bar{f}_1 & \cdots & \bar{f}_N \end{bmatrix}^T$ is given by Equation (19) and after the traffic changes, $\boldsymbol{F}$ is given by Equation (20).

Figure 9 shows changes of potential values of the non-control scheme, PBR-opt, and PBR-opt-mr with $h = 20$, and 100, and Table III shows the time needed from traffic changes until the potential convergence is achieved when $c = 0.2$. Figure 6 shows po-
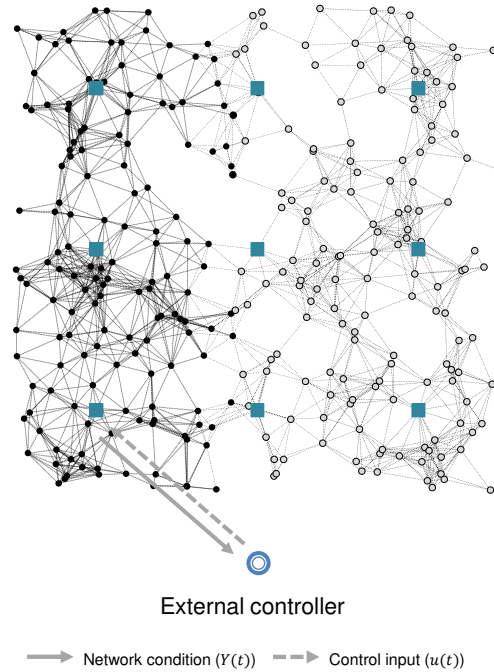
Fig. 8. Network topology ($N = 309$). Sink nodes are illustrated with squares and sensor nodes with dots in this figure. In this figure, only one sink node seems to be connected to the external controller, but all sink nodes actually are wired with the external controller.

tential changes only within $10,000$–$30,000$ s but it takes more than $20,000$ s for the potential convergence with PBR-no-ctrl and PBR-opt as shown in Table III. As a result, the potential convergence speed is accelerated by about 7.03 times, 22.4 times and 22.6 times due to PBR-opt, PBR-opt-mr with $h = 20$ and 100, respectively, compared to the case with PBR-no-ctrl. Table III indicates that potentials converge faster with the larger number of state variables. This characteristic is valid because the controller is generally able to estimate potentials of nodes more correctly with the larger number of state variables. Table III also shows that the convergence speed of potentials with PBR-opt-mr is faster than that with PBR-opt. Feedback inputs provided by the external controller with PBR-opt are optimal in an ideal situation where the external controller and all nodes are synchronized and the external controller can obtain potential information without any communication delays. However, in a real situation, the optimality is not guaranteed because there are estimation errors due to the asynchrony of the system, data packet drops and communication delays. Therefore, the convergence speed with PBR-opt is not always faster than that with PBR-opt-mr. On the contrary, the convergence speed of potentials with PBR-opt seems not to be so different from that with PBR-opt-mr in Figure 9. This indicates that PBR-opt works as well as PBR-opt-mr in the real setting of wireless sensor networks.

Compared with the evaluation of the smaller scale network in Subsection 5.2, degree h of a reduced-order model needs to be larger. This is because a reduced-order model needs a larger number of state variables in order to keep the dominant characteristics of the original model with a larger number of nodes. When h is set to 3 as similar to the evaluation in Subsection 5.2, the controller cannot estimate correctly po-
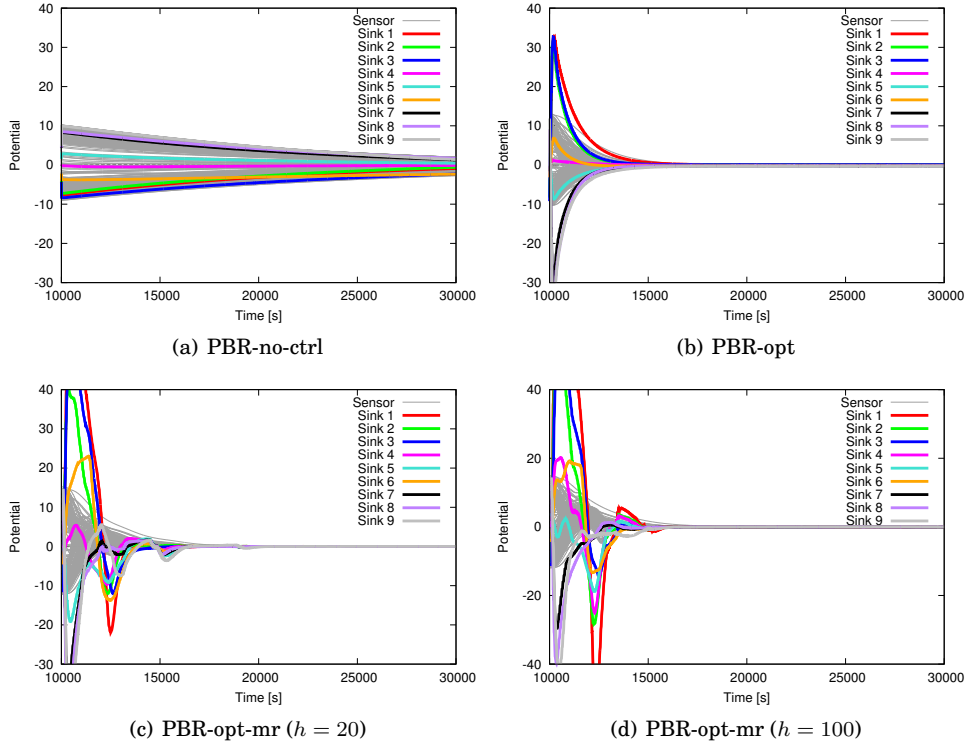
(a) PBR-no-ctrl

(b) PBR-opt

(c) PBR-opt-mr ($h = 20$)

(d) PBR-opt-mr ($h = 100$)

Fig. 9. Potential convergence in the large-scale network

Table III. Potential convergence time in the large-scale network

Scheme	Convergence time [s]
PBR-no-ctrl	363,675
PBR-opt	51,712
PBR-opt-mr ($h = 20$)	16,234
PBR-opt-mr ($h = 100$)	16,064

tentials of nodes and the potential convergence speed is quite slow although we do not show the result in Figure 9. In this paper, we only show the result of evaluations using PBR-opt-mr with $h = 20, 100$, but our proposal can achieve the potential convergence even if $h = 1$, that is, the lower bound of degree h for the network shown in Figure 8 is 1, which is smaller than the lower bound of h for the network of 104 nodes shown in Figure 5 ($= 3$). The lower bound of h deeply depends on the network topology, and so we will investigate the scalability of the reduced order model in the future. In PBR-opt-mr, the computational cost for designing the controller is $O(h^3)$ and that for calculating control inputs is $O(h^2)$. Therefore, the computational cost of PBR-opt-mr with $h = 100$ in Subsection 5.2 and that in Subsection 5.2 is approximately the same, although the number of nodes is 104 and 309, respectively. Moreover, the number of controllable and observable nodes increases compared with the evaluation of the smaller network. That is because the communication overhead for collecting the information of nodes far from observable nodes is large, and these nodes also take a long time to receive the effect of the optimal feedback that is provided to sink nodes.
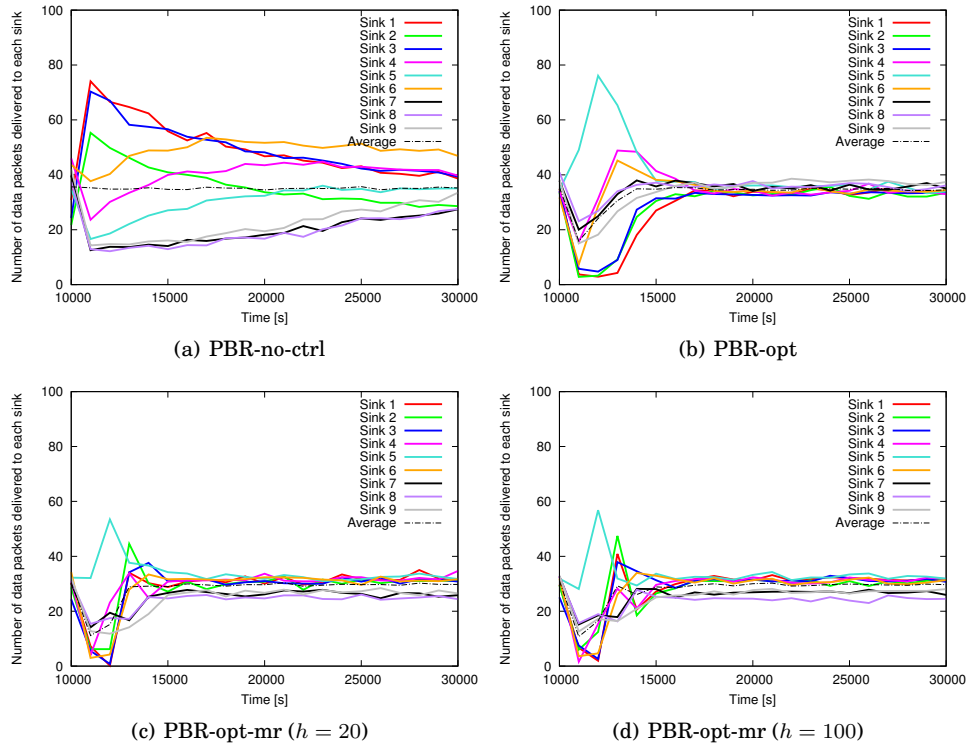
Fig. 10.   Data packets delivered to each sink node in the large-scale network

Figure 10 shows the number of data packets delivered to each sink node every $1,000$ s. As shown in this figure, the traffic convergence speed is accelerated because the potential convergence speed is improved by an optimal feedback. As shown in Figures 9(c) and 9(d), many sink nodes have potential values higher than their target potential values according to control inputs provided by the controller. Therefore, the decrease rate of data packet arrivals in sink nodes is higher than that in evaluation of 104 nodes. However, this is not always true because control inputs deeply depend on the network topology and flow rates. From Figure 10, the average number of data packets delivered to each sink node with PBR-opt-mr is a bit smaller than that with PBR-no-ctrl and PBR-opt. This is because a partial set of data packets and control messages drops near sink nodes because a lot of data packets and control messages are delivered to sink nodes. The number of data packet and control messages dropped will be reduced if the controller collects potential information via several sensor nodes.

In conclusion, our proposal can accelerate the convergence speed even in large-scale networks with the lower computational cost. However, the lower bound degree of a reduced-order model deeply depends on the network topology.

5.4. Robustness against Node Failures

We finally investigate the robustness of PBR-opt-mr. The network model of 104 nodes is used in this evaluation. In this evaluation, several nodes fail at $10,000$ s from the start of the simulation, and we examine changes of potentials and data packets delivered to each sink node after node failures. At $10,000$ s from the beginning of the simulation, q nodes fail. Failing nodes are randomly selected from nodes more than

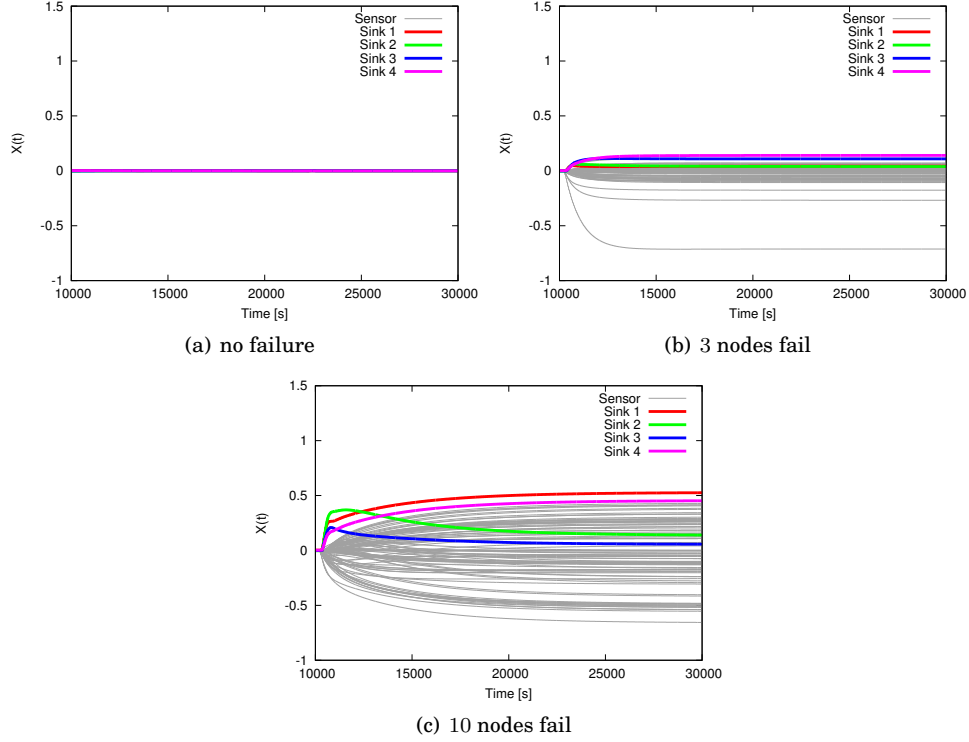(a) no failure

(b) 3 nodes fail

(c) 10 nodes fail

Fig. 11.   Potential convergence in case of node failures (PBR-opt-mr with $h = 3$)

p hops from the sink nodes, so the controller cannot detect them. Instead, each sensor node constantly detect their current neighbors in a self-organizing manner. To avoid the influence of traffic changes, data packet generation rates of all sensor nodes are fixed at 0.001 packets/s, so the i-th element f_i of $\boldsymbol{F}$ is given by

$$\bar{f}_i = \begin{cases} 1, & \text{if } i \in \mathcal{N}_{sen} \\ -\frac{|\mathcal{N}_{sen}|}{|\mathcal{N}_{sin}|}, & \text{if } i \in \mathcal{N}_{sin} \end{cases}.$$

The controller monitors only nodes within $p \ (= 2)$ hops from the sink nodes, as in Subsection 5.2. That is, nodes within two hops from the sink nodes send a control message to the sink nodes at intervals of T_i (50 s), and transfer their potentials to the controller. The degree h of a reduced-order model is set to 3. Other simulation settings are the same as in Subsection 5.2.

Figure 11 shows the potential changes with $q = 0$, 3, and 10. With $q = 0$, no nodes fail. The figure plots $\boldsymbol{X}(t) = \bar{\Theta} - \Theta(t)$ against time t. Note that $\bar{\Theta}$ of each case is the same as $\bar{\Theta}$ in the case where no node fails, because the controller cannot detect the node failures. The reason why the convergence speed is faster than Figure 6 in Subsection 5.2 is that data generation rates of all sensor nodes are fixed in this evaluation. As the figure shows, potentials converge immediately after the node failures, but potential values at the time when they converge are different from $\bar{\Theta}$. This is because $\bar{\Theta}$ changes due to the topology change, but the controller cannot detect the change. However, the potential convergence is achieved soon after node failures and these failures only slightly affect the data packets delivered to each sink node, as explained below.
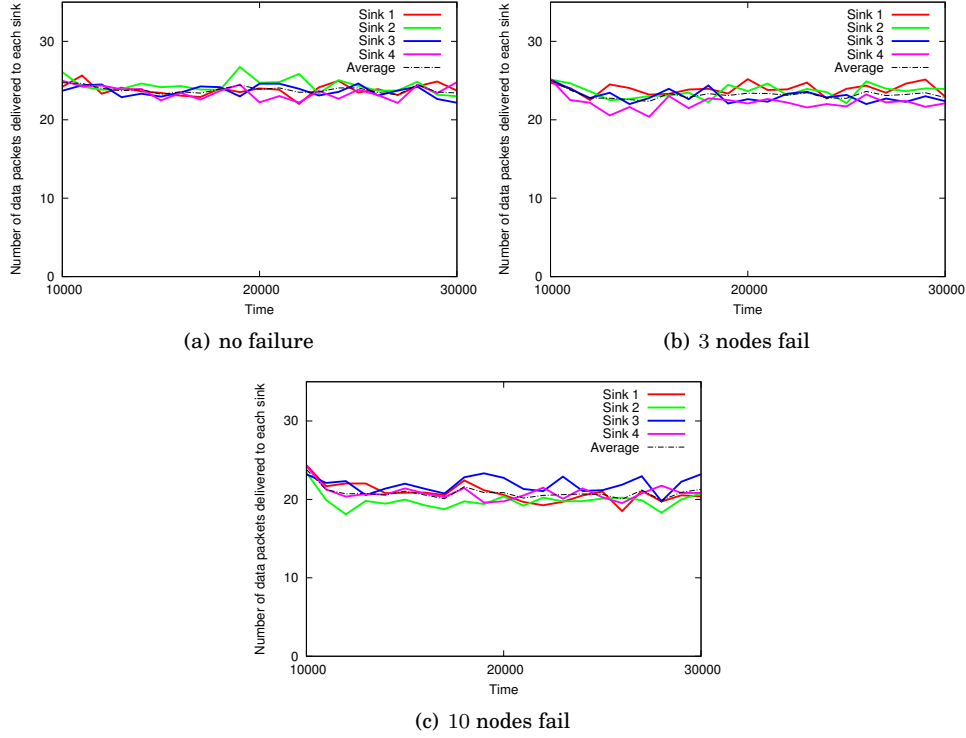
(a) no failure

(b) 3 nodes fail

(c) 10 nodes fail

Fig. 12.   Data packets delivered to each sink node in case of node failures (PBR-opt-mr with $h = 3$)

Table IV. Data arrival rate

q	Data arrival rate [%]
0	99.12
3	99.03
10	99.18

Figure 12 shows the number of data packets delivered to each sink node every $1,000$ s, and the average number of packets. As the figure shows, each sink node receives data packets, equally even in the case where node failures occur and the controller cannot detect them. With $q = 10$, the number of data packets delivered to sink nodes varies, but these differences are not very large. Compared with the case where node failures do not occur, the average number of data packets delivered to each node is smaller when $q = 3, 10$. This is because the number of data packets generated in the entire network is decreased due to the node failures. The data arrival rate is shown in Table IV. The data arrival rate corresponds to the ratio of data packets that arrive at any sink nodes to all generated data packets. This result indicates that the data arrival rate in the case where node failures occur is approximately equal to that in the case where no node fails. In conclusion, optimal feedback by the external controller can enhance convergence while maintaining the high robustness against inaccurate information due to node failures that is an essential characteristic of self-organizing systems.

6. CONCLUSION AND FUTURE WORK

In self-organizing systems, each component behaves according to only local information, which leads to slow convergence. We propose and evaluate potential-based routing with optimal feedback using a reduced-order model, where a controller monitors and estimates system states, and provides optimal feedback for the fastest convergence. Simulation results have shown that optimal feedback using a reduced-order model can facilitate the convergence of potentials while reducing costs for collecting system information and estimating system dynamics. Moreover, our proposal has high scalability and robustness against information loss from node failures.

On the contrary, our proposal remains some challenging tasks. First, the optimal feedback mechanism assumes that the controller has the information of the network topology and the flow rates, which reduces the practicality of our proposal. Second, the potential convergence is achieved as a result of the iterative behavior, i.e., the controller's optimal feedback and nodes' potential updates, so that potential cannot converge if environmental changes occur more frequently than the iterative behavior. Third, the controllability and the stability of networks depends on the network topology. Finally, the optimal control improves the convergence speed of potentials but also causes potential fluctuations as shown by simulation results. These fluctuations lead to data packet drops because sink nodes temporarily have the highest potentials among their neighbors. There is a trade-off between the improvement of the potential convergence speed and potential fluctuations.

We are now studying a distributed version where several controllers independently monitor, estimate and control the system dynamics. For this purpose, it would be a promising direction to design hierarchical controllers based on *the clustered model reduction* [Ishizaki et al. 2014]. With optimal control by a controller, considerable control overhead is needed for collecting network states and estimating network dynamics information when the network is large. By introducing a distributed control mechanism, control overhead can be expected to be reduced because it is not necessary to collect network-wide information.

REFERENCES

Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74, 1 (Jan. 2002), 47–97. Issue 1.

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

Shin'ichi Arakawa, Yuki Minami, Yuki Koizumi, Takashi Miyamura, Kohei Shiomoto, and Masayuki Murata. 2011. A Managed Self-organization of virtual network topology controls in WDM-based optical networks. *Journal of Optical Communications* 32, 4 (Dec. 2011), 233–242.

Sasitharan Balasubramaniam, Kenji Leibnitz, Pietro Lio, Dmitri Botvich, and Masayuki Murata. 2011. Biological principles for future internet architecture design. *IEEE Communications Magazine* 49, 7 (July 2011), 44–52.

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, USA, 37–48.

Jürgen Branke, Moez Mnif, Christian Muller-Schloer, and Holger Prothmann. 2006. Organic Computing - Addressing Complexity by Controlled Self-Organization. In *Proceedings of Leveraging Applications of Formal Methods, Verification and Validation, 2006. ISoLA 2006. Second International Symposium on*. 185–191. DOI:http://dx.doi.org/10.1109/ISoLA.2006.19

Falko. Dressler. 2008. *Self-organization in sensor and actor networks*. Wiley, USA.

Paul Erdős and Alfréd Rényi. 1961. On the strength of connectedness of a random graph. *Acta Mathematica Hungarica* 12, 1 (Dec. 1961), 261–267.

Suyong Eum, Yozo Shoji, Masayuki Murata, and Nozomu Nishinaga. 2014. Design and Implementation of ICN-enabled IEEE 802.11 Access Points as Nano Data Centers. *Journal of Network and Computer Applications* (Aug. 2014).

Pascal Gahinet and Pierre Apkarian. 1994. A linear matrix inequality approach to H   control. *International Journal of Robust and Nonlinear Control* 4, 4 (March 1994), 421–448.

Takayuki Ishizaki, Kenji Kashima, Jun-ichi Imura, and Kazuyuki Aihara. 2014. Model reduction and clusterization of large-scale bidirectional networks. *Automatic Control, IEEE Transaction on* 59, 1 (Jan. 2014), 48–63.

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

Donald E Kirk. 2012. *Optimal control theory: an introduction*. Courier Corporation, Massachusetts.

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, Kashima Kashima, Tomoaki Hashimoto, and Masayuki Murata. 2014. Enhancing Convergence with Optimal Feedback for Controlled Self-organizing Networks. In *to be presented at IEEE 80th Vehicular Technology Conference*. IEEE, Canada.

Munyoung Lee, Junghwan Song, Kideok Cho, Sangheon Pack, Jussi Kangasharju, Yanghee Choi, and Ted Taekyoung Kwon. 2014. SCAN: Content Discovery for Information-Centric Networking. *Computer Networks* (Oct. 2014).

John M Mulvey, Robert J Vanderbei, and Stavros A Zenios. 1995. Robust optimization of large-scale systems. *Operations research* 43, 2 (April 1995), 264–281.

Camelia-Mihaela Pintea. 2014. *Advances in bio-inspired computing for combinatorial optimization problems*. Springer, Berlin.

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

Hartmut Schmeck, Christian Müller-Schloer, Emre Çakar, Moez Mnif, and Urban Richter. 2010. Adaptivity and Self-organization in Organic Computing Systems. *ACM Transactions on Autonomous and Adaptive Systems* 5, 3, Article 10 (Sept. 2010), 32 pages. DOI:http://dx.doi.org/10.1145/1837909.1837911

Alireza Sheikhattar and Mehdi Kalantari. 2013. Fast convergence scheme for potential-based routing in wireless sensor networks. In *Proceedings of Wireless Communications and Networking Conference (WCNC)*. IEEE, Shanghai, 1980–1985.

A. Varga. 1991. Balancing free square-root algorithm for computing singular perturbation approximations. In *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, Vol. 2. IEEE, Brighton, 1062–1065.

Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of ' small-world ' networks. *Nature* 393 (1998), 440–442.

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, Singapore, 113–117.

Xin-She Yang, Zhihua Cui, Renbin Xiao, Amir Hossein Gandomi, and Mehmet Karamanoglu. 2013. *Swarm intelligence and bio-inspired computation: theory and applications*. Elsevier, Nederland.

Zhongshan Zhang, Keping Long, Jianping Wang, and Falko Dressler. 2013. On swarm intelligence inspired self-organized networking: its bionic mechanisms, designing principles and optimization approaches. *Communications Surveys & Tutorials* 16 (July 2013), 513–537.

Chenyu Zheng and Douglas C Sicker. 2013. A Survey on Biologically Inspired Algorithms for Computer Networking. *IEEE Communications Surveys and Tutorials* 15, 3 (Jan. 2013), 1160–1191.

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