# Hierarchical optimal control method for controlling large-scale self-organizing networks

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 the potential for high scalability, adaptability, flexibility, and robustness, which are vital features for realizing future networks. The convergence of self-organizing control, however, is slow in some practical applications in comparison with control by conventional deterministic systems using global information. It is therefore important to facilitate the 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 the same state, it is difficult for an external controller to collect information about the network and to provide control inputs to the network, especially when the network size is large. This is because the computational cost for designing the external controller and for calculating the control inputs increases rapidly as the number of nodes in the network becomes large. Therefore, we partition a network into several subnetworks and introduce two types of controllers, a central controller and several sub-controllers, that control the network in a hierarchical manner. In this study, we propose a hierarchical optimal feedback mechanism for self-organizing systems and apply this mechanism to potential-based self-organizing routing. Simulation results show that the proposed mechanism improves the convergence speed of potential-field construction (i.e., route construction) up to 10.6 fold with low computational and communication costs.

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

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Controlled Self-organization, Hierarchical Robust Control, Potential-based Routing, Fast Convergence

**ACM Reference Format:**
Naomi Kuze, Daichi Kominami, Kenji Kashima, Tomoaki Hashimoto, and Masayuki Murata, YYYY. Hierarchical optimal control method for controlling large-scale self-organizing networks. *ACM Trans. Autonom. Adapt. Syst.* 0, 0, Article 00 ( 0000), 23 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

This research was supported by a Grant-in-Aid for Young Scientist (Start-up) No. 16H06915 of the Japan Society for the Promotion of Science (JSPS) in Japan and was partially supported by a Grant-in-Aid for Scientific Research (B) No. 26289130 from JSPS in Japan.
Author's addresses: N. Kuze and M. Murata, Graduate School of Information Science and Technology, Osaka University; D. Kominami, Graduate School of Economics, Osaka University; K. Kashima, Graduate School of Informatics, Kyoto University; T. Hashimoto, Faculty of Engineering, Osaka Institute of Technology

DOI: http://dx.doi.org/10.1145/0000000.0000000

## 1. INTRODUCTION

Self-organization, in which components behave individually and autonomously, is a phenomenon in distributed systems such as physical, chemical, biological systems [Pintea 2014; Yang et al. 2013]. In a self-organizing system, each component follows simple rules that rely on locally available information only. Through direct or indirect interactions among components, a global behavior or pattern emerges at 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 for emergence of a global behavior or pattern; this considerably reduces the computational cost and communication overhead that collecting global information would entail. This localized behavior affords the ability to handle local failures and small environmental changes via the interaction of local components. Thus, self-organizing systems are expected to automatically recover from failures and adapt to environmental changes. Therefore, various models based on self-organization 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 that features such as scalability, adaptability, and robustness will be improved to an extent not possible by conventional network control methods [Balasubramaniam et al. 2011].

Although self-organization control that does not use global knowledge about the current network state has various benefits, such control has critical disadvantages that complicate practical use in industrial and business systems [Dressler 2008]. One drawback is that 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 by solely local interactions in self-organizing systems. Furthermore, self-organizing systems that use local information only sometimes become trapped in local optima. This indicates that system requirements are not always satisfied, which is a significant problem for practical use in industrial and business systems. In contrast, in conventional centralized systems, global information can lead to an optimal solution, though the required computational cost may be extremely high.

Such disadvantages of self-organization noted in real applications led to the idea of controlled (guided, managed) self-organization, in which the self-organizing system is controlled through some constraints [Martius and Herrmann 2010; Müller-Schloer et al. 2011; Prokopenko 2014]. For example, the authors of [Arakawa et al. 2011; Kominami et al. 2013] used the concept of controlled self-organization, in which an external observer/controller guides self-organizing optical networks [Arakawa et al. 2011] and sensor network [Kominami et al. 2013] systems through a feedback mechanism that leads them to a desired (globally optimal/semi-optimal) state. These studies showed that self-organizing systems can be guided to the desired state by introducing an external observer/controller. Along another line, we proposed an optimal feedback mechanism for self-organizing systems [Kuze et al. 2014] to enhance the speed of convergence to an optimal or semi-optimal solution. In this mechanism, an external controller collects information about the network and provides optimal feedback inputs to cause faster convergence. For calculating optimal feedback inputs, the external controller estimates the states of nodes in the entire network by using a mathematical model that describes the network dynamics. The simulation results in [Kuze et al. 2014] showed that optimal control by the external controller can facilitate the convergence of self-organizing networks.

We showed that the introduction of an external controller into self-organizing networks can accelerate their convergence; however, this mechanism still has problems

with scalability. The mechanism requires topology information about the whole network, and the computational cost for designing the estimation model is roughly proportional to the cube of the number of nodes, which makes it increasingly difficult for the external controller to collect topology information and to obtain the estimation model of the whole network as the network size becomes larger. These problems become critical when topological changes occur, because the estimation model needs to be redesigned to include the latest topology information so that the external controller can guide the network to converge to a targeted state.

In this study, we propose a hierarchical optimal feedback mechanism to control a self-organizing network while the system has the high scalability, and be able to keep working even when local changes or failure occur, which are originally inherent in self-organizing systems. The basic idea of the mechanism is shown in [Kuze et al. 2015], in which a network has been partitioned into several sub-networks that are controlled in a hierarchical manner by two types of controllers, i.e., a *central controller* and *sub-controllers*. A sub-controller monitors a sub-network, that is, only a part of the entire network, and provides optimal feedback inputs to the sub-network so that fast convergence can be achieved. Specifically, the sub-controller collects information regarding the sub-network, such as node states and network topology. Then, the sub-controller estimates the dynamics of the sub-network by using a mathematical model that describes the sub-network dynamics, and calculates optimal feedback inputs that facilitate the convergence speed of self-organizing systems. This is based on robust control theory [Zhou et al. 1995], in which a controller is designed to work with limited information. However, sub-controllers cannot achieve a global optimality by themselves because they does not have information of sub-networks other than their corresponding ones. The role of a central controller is to guide sub-networks to achieve the identical global optimality. A central controller obtains information about the entire network from sub-controllers, estimates the degrees of interactions among sub-networks, and then returns feedback inputs to the sub-controllers. To reduce the computational cost of the central controller and sub-controllers, we improve [Kuze et al. 2015] by reducing the number of state variables of the estimation models for both the central controller and the sub-controllers with a model reduction technique [Antoulas et al. 2006].

The main contribution of this study is that we realize a hierarchical control mechanism for enhancing the convergence speed of self-organizing systems at fairly low computational cost and reveal the effectiveness of our proposed mechanism through computer simulation. The number of state variables of the estimation model is proportional to the size of the actual network model, namely, the number of nodes. The computational cost needed for the dynamics estimation and the control input calculation by controllers increase rapidly as the number of nodes becomes large. The external controller or sub-controller needs a computational cost on the order of $O({N_{dim}}^2)$ for the dynamics estimation and of $O({N_{dim}}^3)$ for the design of the estimation model, where $N_{dim}$ is the number of state variables of the estimation model. The sum of the computational cost of each sub-controller is much smaller than the computational cost of the external controller (Table I). Moreover, we reduce the number of state variables of the estimation model ($N_{dim}$) via a model reduction technique to reduce the computational cost. Note that the smaller size of a reduced model leads to the loss of the original model while decreasing the computational cost. This trade-off relation requires us to further examine the number $N_{dim}$ of state variables of the model. Furthermore, local environmental changes can be handled by the corresponding sub-network because each sub-controller monitors/controls the corresponding sub-network in an individual and automatic manner. This contributes to high scalability and adaptability of the system.

Table I. Computational costs in each scheme. $N$ is the total number of nodes; $N_{sub}$ is the number of nodes in a sub-network; $h_{ex}$ is the number of state variables in the estimation model used by the external controller in PBR-opt-mr, a non-hierarchical optimal control mechanism proposed in previous work [Kuze et al. 2014]; $h$ and $h_{sub}$ are the number of state variables in the estimation model used by the central controller and each sub-controllers, respectively, in PBR-h-opt-mr.

| Scheme | Controller type | $N_{dim}$ | Computational cost | |
|---|---|---|---|---|
| | | | Controller design | Calculation of feedback inputs |
| PBR-no-ctrl | - | - | 0 | 0 |
| PBR-opt-mr | External | $h_{ex}(< 2N)$ | $O(N_{dim}^3)$ | $O(N_{dim}^2)$ |
| PBR-h-opt | Sub | $2N_{sub}(< 2N)$ | $O(N_{dim}^3)$ | $O(N_{dim}^2)$ |
| | Central | $2N$ | $O(N_{dim}^2)$ | $O(N_{dim}^2)$ |
| PBR-h-opt-mr | Sub | $h_{sub}(< 2N_{sub})$ | $O(N_{dim}^3)$ | $O(N_{dim}^2)$ |
| | Central | $h(< 2N)$ | $O(N_{dim}^2)$ | $O(N_{dim}^2)$ |

The effectiveness of our proposal is evaluated through computer simulation studies in which we consider potential-based routing—a self-organizing routing mechanism for wireless sensor networks—with optimal feedback and evaluate the convergence speed after environmental changes. Our proposal is suitable for wireless sensor networks because the simple behavior of each component in self-organizing systems needs neither much capability nor much energy, whereas optimal control by the external controller improves the performance of these systems. It is worth mentioning that potential-routing for wireless sensor networks is an example of an application of our proposal, and our proposal can be adapted to other types of mechanisms, networks and situations. The optimality of our feedback mechanism is analytically guaranteed in synchronous systems but not in asynchronous systems. In our evaluation, we first assume a wireless sensor network where nodes behave asynchronously for revealing the advantages and properties of our proposal. To evaluate the scalability of our proposal, we next conduct a simulation with a larger network. Finally, we evaluate our method in the case where a new sub-network is added to the network and show that local environmental changes can be handled by the corresponding sub-controller with our method.

The remainder of this paper is organized as follows. First, we briefly explain potential-based routing in Section 2. We propose and explain potential-based routing with optimal feedback in Section 3. We then show, through simulation, the proposed method's ability to facilitate fast adaptation to environmental changes and discuss our proposal in Section 4. Finally, in Section 5, we present our conclusions and suggest areas for future work.

## 2. POTENTIAL-BASED ROUTING

Potential-based routing is a self-organizing routing mechanism in which each node chooses a route with a hop-by-hop forwarding rule. Such mechanisms are actively used 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 2014; Eum et al. 2014; Lee et al. 2014]. Here, we assume that potential-based routing is used in wireless sensor networks, where information gathering is not frequent 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 wireless sensor networks, data packets are generally sent to a sink node, and fewer hops to the sink node is reflected in a lower potential value. 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 1. Potential-based routing has high scalability because each node uses only local information for

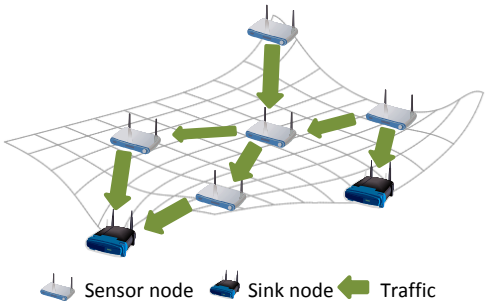



Fig. 1. Potential-based routing

calculating potentials and a local rule for forwarding data. In Sections 2.1 and 2.2, we describe a potential field construction and show how to select the next hop node by using the potential field.

### 2.1. Potential Field Construction

Sheikhattar and Kalantari [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 potential convergence. Node $n$'s potential at time $t$, $\theta_n(t)$, is calculated by Equation (1).

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

Here, $\mathcal{N}(n)$ is a set of node $n$'s neighbors, and $\alpha$ is a parameter that determines the weights of the increased amount of the potential value from time $t-1$ to $t$ when calculating the next potential. Larger values of $\alpha$ mean that the amount of the potential change is more heavily weighted, and therefore the system becomes less subject to current noise, though the convergence speed is slower. The parameter $\beta$ determines the amount of influence of neighbor-node potentials. In this, $\sigma_n$ is defined as $\sigma_0/|\mathcal{N}(n)|$ ($\sigma_0$ is a parameter), and $f_n(t)$ corresponds to the flow rate of node $n$ at time $t$. For sensor node $n$, $f_n(t)$ is a negative value that indicates the data generation rate of node $n$. The data generation rates of sensor nodes are generally determined by the applications. On the other hand, for sink node $n$, $f_n(t)$ is a positive value that determines the rate of data packets 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, that is, $\sum_{n \in \{1, \cdots, N\}} f_n(t) = 0$, then a potential field is constructed such that the actual rates of data packets delivered to nodes satisfy the given flow rates (i.e., all gradients), which are potential differences between next-hop nodes, correspond to the appropriate flow rates.

The convergence speed based on Equation (1) is faster than that of simple Jacobi iterations (such as in our previous work [Kominami et al. 2013]), but it still takes a long time to converge owing to its calculation being based on local information only. We introduce external controllers into potential-based routing to observe network states (potential values), estimate future states, and regulate the potentials of a partial set of nodes for faster convergence.

### 2.2. Routing

If a node has a data packet, then it forwards the packet 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 having a lower potential value than itself, and the packet eventually arrives at a sink node in this way. Specifically, a next-hop node is selected with a probability in proportion to the difference of 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

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

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

## 3. POTENTIAL-BASED ROUTING WITH HIERARCHICAL OPTIMAL FEEDBACK

In this section, we describe a model of the network dynamics and explain our hierarchical optimal control scheme.

### 3.1. Overview

We propose a hierarchical optimal feedback mechanism that facilitates the convergence speed of self-organizing systems. We apply our proposed mechanism to potential-based routing since potential-based routing, whose dynamics routing can be described as a simple linear system, is appropriate for our goal to reveal the upper limit of self-organizing network control systems with the hierarchical optimal feedback. Our proposed mechanism is shown in Figure 2. A network is partitioned into several sub-networks. Each sub-network is connected to a sub-controller via a part of the nodes that belong to the sub-network, whereas a central controller is connected to all sub-controllers. Note that our proposed mechanism work without controllers because nodes behaves autonomously and automatically in a self-organizing manner although the convergence speed is low.

A *sub-controller* monitors information about its corresponding sub-network, and in particular the potential values of a partial set of nodes, which we call *observable nodes*. The sub-controller then returns suitable control inputs to another partial set of nodes, which we call *controllable nodes*, for accelerating the convergence of the potential distribution of the sub-network toward the target potential distribution, which is described in Section 3.2. The control input from the sub-controller affects the potential values of the controllable nodes. Nodes interact with each other in a self-organizing manner so that the effect of the control inputs propagates to the entire network through local interactions. The information about the sub-network topology is needed for designing the sub-controller. Furthermore, we need the flow rates of nodes in the sub-network for calculating target potential values. Although this information cannot be estimated with our proposal, we assume that changing the frequencies of the network topology and flow rates occurs much less frequently than changes to potentials and that changes can be reported to the sub-controllers only when they occur. Such data collection is feasible with the proper choice of control interval.

The role of the *central controller* is to estimate interactions among sub-networks and to provide feedback inputs to sub-controllers so that the whole network reaches the targeted state. For this purpose, the central controller obtains the network informa-

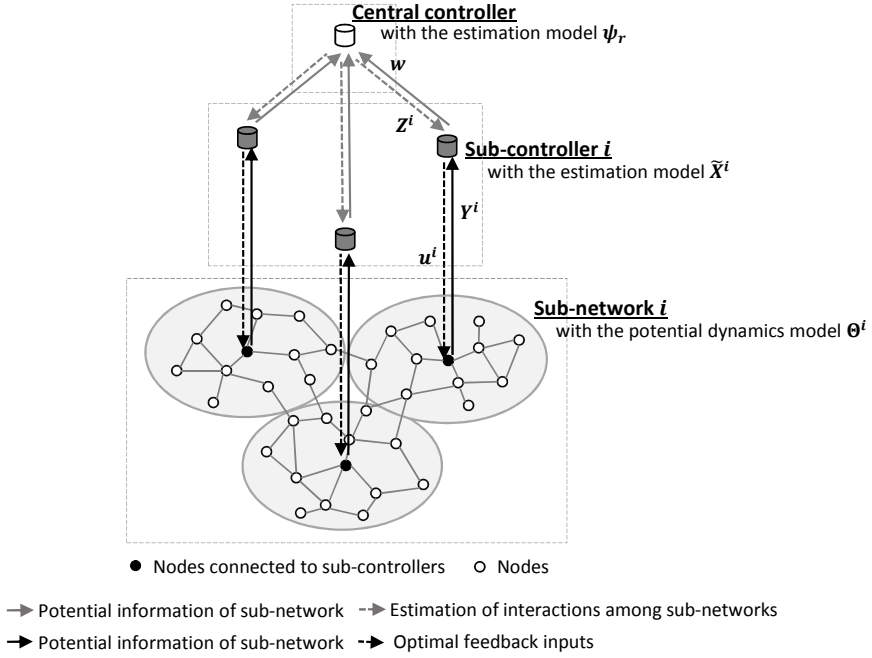



Fig. 2. Potential-based routing with a hierarchical optimal control scheme

tion from sub-controllers and then returns suitable feedback inputs to sub-controllers. Without the central controller, the ignored interactions among sub-networks lead to an undesirable target potential distribution and can even cause network instability. Because the estimation by the central controller needs a high computational cost, we introduce the reduced-order model to reduce this computational cost for estimating the potential values of non-observable nodes, as explained in Section 3.4.2.

We assume that the sub-controllers and sink nodes are supplied power so that these sink nodes can have a reliable direct connection with sensor nodes and sub-controllers at all times. Therefore, in our proposal, sub-controllers monitor sub-network information and provide control inputs via connected sink nodes. We consider controllable nodes as sink nodes and observable nodes as nodes within $p$ hops from sink nodes. For estimating the interactions among sub-networks, the central controller obtains the potential information about all observable nodes from the sub-controllers.

### 3.2. Dynamics of Sub-networks

Let the dynamics of the potentials be given by a deterministic discrete-time model. Nodes locally interact with each other for updating their potentials. With our proposed mechanism, sub-controllers send feedback inputs $\boldsymbol{u}(t) = [\eta_1(t) \ \cdots \ \eta_{N_{ctrl}}(t)]^T$ to $N_{ctrl}$ controllable nodes for facilitating 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

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

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

Next, we describe the potential dynamics of sub-networks. Potentials of nodes in a sub-network are updated in accordance not only with interactions of nodes within sub-network $i$ but with interactions among sub-networks. We define $\mathcal{S}$ as a set of sub-networks, where a sub-network $i$ ($\in \mathcal{S}$) includes $N^i$ nodes. The potential values of nodes in sub-network $i$ are described as a vector $\boldsymbol{\Theta}^i(t) = \left[\boldsymbol{\theta}^i_1(t) \ \cdots \ \boldsymbol{\theta}^i_{N^i}(t)\right]^T$ using $\boldsymbol{\theta}_n(t) = [\theta_n(t) \ \theta_n(t+1)]$. The potential dynamics of sub-network $i$ is given by Equation (3) using flow matrix $\boldsymbol{F}^i$ and control inputs $\boldsymbol{u}^i$.

$$\boldsymbol{\Theta}^i(t+1) = \boldsymbol{A}^i\boldsymbol{\Theta}^i(t) + \sum_{j \in \mathcal{S}-\{i\}} \boldsymbol{A}^{i,j}\boldsymbol{\Theta}^j(t) + \left(\beta\sigma\boldsymbol{F}^i + \boldsymbol{u}^i(t)\right) \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3}$$

$\boldsymbol{A}^i$ and $\boldsymbol{A}^{i,j}$ are matrices that describe interactions, respectively, within sub-network $i$ and between sub-network $i$ and $j$. $\boldsymbol{A}^i$ and $\boldsymbol{A}^{i,j}$ are given by

$$\boldsymbol{A}^i = \boldsymbol{I}_{N^i \times N^i} \otimes \begin{bmatrix} 0 & 1 \\ -\alpha & \alpha+1 \end{bmatrix} + (\boldsymbol{L}^i_{intra} - \boldsymbol{G}^i) \otimes \begin{bmatrix} 0 & 0 \\ 0 & \beta\sigma \end{bmatrix},$$

$$\boldsymbol{A}^{i,j} = \boldsymbol{L}^{i,j}_{inter} \otimes \begin{bmatrix} 0 & 0 \\ 0 & \beta\sigma \end{bmatrix},$$

where the $(N^i \times N^i)$-matrix $\boldsymbol{L}^i_{intra}$ and the $(N^i \times N^i)$-matrix $\boldsymbol{G}^i$ correspond to the adjacency and degree matrices of sub-network $i$, respectively. $\boldsymbol{L}^{i,j}_{inter}$ is an $(N^i \times N^j)$-matrix that describes links connecting between sub-networks $i$ and $j$. The element $l^{i,j}_{inter}(n,m) \in \{0,1\}$ of $\boldsymbol{L}^{i,j}_{inter}$ is 1 if and only if there is a link between node $n$ of sub-network $i$ and node $m$ of sub-network $j$. $\boldsymbol{I}_{N \times N}$ is the $N \times N$ identity matrix.

Under these dynamics, the potentials $\boldsymbol{\Theta}^i$ converge to $\bar{\boldsymbol{\Theta}}^i$ and we consider them as the target potentials, as given by the solution of

$$(\boldsymbol{A}^i - \boldsymbol{I}_{2N^i \times 2N^i})\bar{\boldsymbol{\Theta}}^i + \sum_{j \in \mathcal{S}-\{i\}} \boldsymbol{A}^{i,j}\bar{\boldsymbol{\Theta}}^j = -\beta\sigma\boldsymbol{F}^i \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{4}$$

Each sub-controller calculates the target potential distribution of the corresponding sub-network in advance for calculating control inputs.

### 3.3. Optimal Control of Sub-networks by Sub-controllers

*3.3.1. Design of Sub-controllers.* We explain how the sub-controllers calculate control feedback to corresponding sub-networks. For all $i$ ($\in \mathcal{S}$), sub-network $i$ is connected to sub-controller $i$, For calculating control inputs $\boldsymbol{u}^i$, sub-controller $i$ monitors potentials $\boldsymbol{Y}^i$ of observable nodes in sub-network $i$ via nodes that have a direct connection to the sub-controller, and receives information $\boldsymbol{Z}^i$ of interactions among sub-networks from the central controller. The $(2N^i_{obs} \times 1)$-vector $\boldsymbol{Y}^i(t)$ is given by $\boldsymbol{Y}^i(t) = \boldsymbol{C}^i\boldsymbol{X}^i(t)$ using $\boldsymbol{X}^i(t) = \bar{\boldsymbol{\Theta}}^i - \boldsymbol{\Theta}^i(t)$, the gap between the current and target potential values. $N^i_{obs}$ is the number of observable nodes in sub-network $i$, and $(2N^i_{obs} \times 2N^i)$-matrix $\boldsymbol{C}^i$ determines the observable nodes. The element $c^{2i}(n,m)$, $c^{2i+1}(n,m) \in \{0,1\}$ of $\boldsymbol{C}^i$ is 1 if

and only if sub-controller $i$ monitors the potential value of node $m$ as the $n$th element of $\boldsymbol{Y}^i$.

First, sub-controller $i$ estimates $\boldsymbol{X}^i(t)$ from observable information $\boldsymbol{Y}^i$ and interaction information $\boldsymbol{Z}^i$. The $(2N^i \times 1)$-vector $\tilde{\boldsymbol{X}}^i(t)$ is the estimation of $\boldsymbol{X}^i(t)$ by sub-controller $i$ and is given by

$$\tilde{\boldsymbol{X}}^i(t+1) = \boldsymbol{A}^i\tilde{\boldsymbol{X}}^i(t) + \boldsymbol{B}^i\boldsymbol{u}^i(t) + \boldsymbol{Q}^i\left(\boldsymbol{Y}^i(t) - \tilde{\boldsymbol{Y}}^i(t)\right) + \boldsymbol{Z}^i(t), \tag{5}$$

$$\tilde{\boldsymbol{Y}}^i(t) = \boldsymbol{C}^i\tilde{\boldsymbol{X}}^i(t). \tag{6}$$

If $\tilde{\boldsymbol{X}}^i(t)$ is close to $0$, then the potentials are estimated to be close to their target values. We need to select $\boldsymbol{Q}^i$ such that it satisfies "$\boldsymbol{A}^i - \boldsymbol{Q}^i\boldsymbol{C}^i$ is stable," which allows the potentials to converge. The matrix $\boldsymbol{B}^i$, which determines the weight of control inputs from the sub-controller $i$, is given by

$$\boldsymbol{B}^i = \boldsymbol{S}^i \otimes \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

The $(N^i \times N^i_{ctrl})$-matrix $\boldsymbol{S}^i$ specifies the controllable node of sub-network $i$. The element $s^i(n,m) \in \{0,1\}$ of $\boldsymbol{S}^i$ is 1 if and only if node $n$ receives the $m$th element of $\boldsymbol{u}^i(t)$ as control input $\eta_n(t)$. Next, the control inputs $\boldsymbol{u}^i(t)$ is calculated according to

$$\boldsymbol{u}^i(t) = -\boldsymbol{V}^i\tilde{\boldsymbol{X}}^i(t). \tag{7}$$

$\boldsymbol{V}^i$ is the optimal gain matrix that minimizes the quadratic cost function $H^i(\boldsymbol{u}^i) = \sum_{n=1}^{\infty}\left(||\tilde{\boldsymbol{X}}^i(n)|| + r||\boldsymbol{u}^i(n)||\right)$ for the system. In other words, the control inputs $\boldsymbol{u}^i$ is calculated to make potentials fast converge to their target values and to reduce the changing amount of potendials due to the control feedback. $r$ is a parameter that regulates the trade-off between the convergence speed and the stability of potentials. With lower values of $r$, the convergence speed of potentials is faster, but potentials change by larger amounts at one time because the sub-controller is allowed to provide larger $\boldsymbol{u}^i$.

*3.3.2. Model Reduction for Sub-controllers.* With the estimation model described by Equations (5)–(7), which has $2N^i$ state variables, the optimal feedback $\boldsymbol{u}^i(t)$ is calculated with computational cost $O({N^i}^2)$. For reducing the computational cost, sub-controllers use reduced-order models that have fewer state variables [Antoulas et al. 2006]. 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 the reduced-order model for sub-controller $i$, the estimation model is expressed as an $(h^i \times 1)$-vector $\tilde{\boldsymbol{X}}^i_r$ whose elements are linear transformations of the original model $\tilde{\boldsymbol{X}}^i$. A constant number $h^i$ is given by the network manager. $\tilde{\boldsymbol{X}}^i_r$ is caluculated by

$$\tilde{\boldsymbol{X}}^i_r(t+1) = \boldsymbol{A}^i_r\tilde{\boldsymbol{X}}^i_r(t) + \boldsymbol{B}^i_r\boldsymbol{u}^i_r(t) + \boldsymbol{Q}^i_r\left(\boldsymbol{Y}^i(t) - \tilde{\boldsymbol{Y}}^i(t)\right) + \boldsymbol{Z}^i(t), \tag{8}$$

$$\tilde{\boldsymbol{Y}}^i(t) = \boldsymbol{C}^i_r\tilde{\boldsymbol{X}}^i_r(t), \tag{9}$$

$$\boldsymbol{u}^i_r(t) = -\boldsymbol{V}^i_r\tilde{\boldsymbol{X}}^i_r(t). \tag{10}$$

The $(2N^i_{ctrl} \times 1)$-vector $\boldsymbol{u}^i_r(t)$ corresponds to control inputs provided to sub-network $i$ by the sub-controller $i$ with the reduced-order model. We choose matrices $\boldsymbol{A}^i_r$, $\boldsymbol{B}^i_r$, $\boldsymbol{Q}^i_r$, $\boldsymbol{C}^i_r$, and $\boldsymbol{V}^i_r$ of compatible dimensions such that $\boldsymbol{u}^i_r(t) \approx \boldsymbol{u}^i(t)$ for all input $\boldsymbol{Y}^i(t)$, $\boldsymbol{Z}^i(t)$.

With a reduced-order model, control inputs $\boldsymbol{u}_r^i(t)$ are calculated with computational cost $O({h^i}^2)$. In general, a model that has more state variables allows the controller to perform estimations more accurately; however, 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, $h^i$ needs to be properly determined in accordance with the requirements or system properties.

### 3.4. Estimation of Interactions among Sub-networks

*3.4.1. Design of Central Controller.* We explain how the central controller estimates interactions among sub-networks. Many researchers have studied hierarchical systems [Sandell et al. 1978; Herrmann et al. 2004] in which components of different layers interact for controlling systems. In our proposed mechanism, we take prior work [Ishizaki et al. 2014], which focused on model reduction for large-scale systems.

The central controller first obtains potential information $\boldsymbol{w}(t) = \left[\boldsymbol{Y}^1(t)^T \ \cdots \ \boldsymbol{Y}^{|\mathcal{S}|}(t)^T\right]^T$ from sub-controllers for estimating the interactions among sub-networks. Then, the central controller estimates degrees $\boldsymbol{Z}(t) = \left[\boldsymbol{Z}^1(t)^T \ \cdots \ \boldsymbol{Z}^{|\mathcal{S}|}(t)^T\right]^T$ of interactions among sub-networks by

$$\boldsymbol{\psi}(t+1) = \boldsymbol{J}\boldsymbol{\psi}(t) + \boldsymbol{K}\boldsymbol{u}(t) + \boldsymbol{O}\boldsymbol{w}(t), \tag{11}$$

$$\boldsymbol{Z}(t) = \boldsymbol{T}\boldsymbol{\psi}(t). \tag{12}$$

The $(2N \times 1)$-vector $\boldsymbol{\psi}$ describes an estimation model for the central controller. $\boldsymbol{J}$, $\boldsymbol{K}$, $\boldsymbol{O}$, and $\boldsymbol{T}$ are given by

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{A}^1 & \boldsymbol{A}^{1,2} & \cdots & \boldsymbol{A}^{1,|\mathcal{S}|} \\ \boldsymbol{A}^{2,1} & \boldsymbol{A}^2 & \cdots & \boldsymbol{A}^{2,|\mathcal{S}|} \\ \vdots & & & \vdots \\ \boldsymbol{A}^{|\mathcal{S}|,1} & \boldsymbol{A}^{|\mathcal{S}|,2} & \cdots & \boldsymbol{A}^{|\mathcal{S}|} \end{bmatrix} - \boldsymbol{Q} \begin{bmatrix} \boldsymbol{C}^1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}^2 & \cdots & \boldsymbol{0} \\ \vdots & & & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{C}^{|\mathcal{S}|} \end{bmatrix}, \boldsymbol{K} = \begin{bmatrix} \boldsymbol{B}^1 \\ \boldsymbol{B}^2 \\ \vdots \\ \boldsymbol{B}^{|\mathcal{S}|} \end{bmatrix},$$

$$\boldsymbol{O} = \boldsymbol{Q}, \boldsymbol{T} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{A}^{1,2} & \cdots & \boldsymbol{A}^{1,|\mathcal{S}|} \\ \boldsymbol{A}^{2,1} & \boldsymbol{0} & \cdots & \boldsymbol{A}^{2,|\mathcal{S}|} \\ \vdots & & & \vdots \\ \boldsymbol{A}^{|\mathcal{S}|,1} & \boldsymbol{A}^{|\mathcal{S}|,2} & \cdots & \boldsymbol{0} \end{bmatrix},$$

where $(2N \times 2N_{obs})$-matrix $\boldsymbol{Q}$ satisfies "$\boldsymbol{J}$ is stable." $\boldsymbol{0}$ is a zero matrix.

*3.4.2. Model Reduction for Central Controller.* With the estimation model described by Equations (11) and (12), which has $2N$ state variables, degrees $\boldsymbol{Z}(t)$ are calculated with $O(N^2)$. For reducing the computational cost, the central controller also uses a reduced-order model.

In the reduced-order model for the central controller, the estimation model is expressed as an $(h \times 1)$-vector $\boldsymbol{\psi}_r(t)$, and the reduced-order model is given by

$$\boldsymbol{\psi}_r(t+1) = \boldsymbol{J}_r\boldsymbol{\psi}_r(t) + \boldsymbol{K}_r\boldsymbol{u}(t) + \boldsymbol{O}_r\boldsymbol{w}(t), \tag{13}$$

$$\boldsymbol{Z}_r(t) = \boldsymbol{T}_r\boldsymbol{\psi}_r(t). \tag{14}$$

The $(2N \times 1)$-vector $\boldsymbol{Z}_r(t)$ corresponds to feedback inputs provided to sub-controllers by the central controller. We need to choose matrices $\boldsymbol{J}_r$, $\boldsymbol{K}_r$, $\boldsymbol{O}_r$, and $\boldsymbol{T}_r$ of compatible dimensions such that $\boldsymbol{Z}_r(t) \approx \boldsymbol{Z}(t)$ for all inputs $\boldsymbol{u}(t)$, $\boldsymbol{w}(t)$.

With a reduced-order model, feedback inputs $\boldsymbol{Z}_r(t)$ are calculated with computational cost $O(h^2)$. The value of $h$ needs to be properly determined as with $h^i$ because there is a trade-off between the accuracy of the estimation and the computational cost for it.

## 4. PERFORMANCE EVALUATION

We evaluate our proposal to clarify the advantages and properties of hierarchical optimal feedback using the reduced-order model.

### 4.1. Overview

We conduct a computer simulation and evaluate the convergence speed by comparing the results with our proposal (potential-based routing using hierarchical optimal feedback with and without model reduction [*PBR-h-opt-mr* and *PBR-h-opt*, respectively]) with potential-based routing using non-hierarchical optimal feedback including model reduction (*PBR-opt-mr*) as proposed in [Kuze et al. 2014], and with the non-control scheme (*PBR-no-ctrl*) proposed in [Sheikhattar and Kalantari 2014]. First, in Section 4.3, we evaluate the potential convergence speed after traffic changes to show that our proposed methods enhance the convergence speed of the self-organizing system. Moreover, we show that our proposal can be adapted to massive and frequent environmental changes, in Section 4.4. Finally, in Section 4.5, to demonstrate that our proposal can handle topological changes via redesigning the estimation model of the corresponding sub-controller, we evaluate the case in which a new sub-network is added to the network.

For the network simulator, we use an event-driven packet-level simulator written in Visual C++ that calls MATLAB functions *dlqr* to design an optimal central controller/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 an Intel(R) Xeon(R) CPU with 2.70 GHz and 64.0 GB memory. The simulator has been developed by us. 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 receiver-driven data transmission protocol. We use a disk model as a physical layer model where data packets drop with chance $100\%$ if they collide with each other. As the capacity of each sensor node is limited in wireless sensor networks, we set the queue size of each sensor node to $1$.

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

### 4.2. Simulation Settings

We evaluate changes in potentials in potential-based routing and the number of data packets delivered to each sink node after environmental changes, such as traffic changes or the addition of a new sub-network. To measure the convergence speed of potentials, we define the degree $\epsilon_n(t)$ of the potential convergence for each node that is given by $\epsilon_n(t) = |\bar{\theta}_n - \theta_n(t)|$, where $\bar{\theta}_n$ corresponds to the target potential value of node $n$. We consider convergence to be achieved when $\epsilon_n(t)$ is sufficiently small for all nodes. The convergence time of potentials is defined as the minimum time taken by all sensor and sink nodes to satisfy the condition

$$\epsilon_n(t) < \delta, \tag{15}$$

where $\delta$ is constant value $0.01$. In an ideal situation, where all nodes are completely synchronized and there is no noise, all sensor and sink nodes satisfy Equation (15)

finally, even if $\delta = 0$; however, in an actual situation, not all nodes can satisfy Equation (15).

In a real network, it is difficult for sub-controllers to monitor up-to-date potential values of all nodes in the corresponding sub-networks because of the overhead for collecting potential values, especially when the number of nodes of each sub-network increases. Therefore, sub-controllers monitor only nodes within $p$ hops from the corresponding sink nodes. Nodes within $p$ hops from sink nodes send a control message to sink nodes at interval $T_s$ to notify sub-controllers of their potential values. The $n$th element $c_n^i$ of $\boldsymbol{C}^i$ ($i \in \mathcal{S}$), which determines a set of observable nodes, is set to

$$c_n^i = \begin{cases} 1, & \text{if node } n \text{ is within } p \text{ hops from a sink node in sub-network } i \\ 0, & \text{otherwise} \end{cases}.$$

Sub-controllers do not directly monitor node potentials beyond $p$ hops from their connected sink nodes; instead, they only estimate them, using the estimation model. Each sub-controller calculates an optimal control using the potential information obtained through the observation and estimation at interval $T_s$. Similarly, the central controller obtains the potential information about observable nodes from all sub-controllers at interval $T_c$ for estimating interactions among sub-networks. In contrast, each node updates its next potential value at an interval $T_p$. Typically, $T_c = T_s = T_p$ for matching with the potential dynamics. In our evaluations, we consider a control cycle ($T_c = T_s = T_p$) as a unit time step.

At the beginning of the simulation, the potential values of all nodes are initialized to $0$. During the first $20$ steps, each node exchanges its potential value with neighbor nodes and updates its potential value so that the potential values are stabilized. At $20$ steps, data packets begin to be generated at sensor nodes according to the Poisson process with their flow rates. We evaluate the convergence speed of potentials and data packets delivered to each sink node after traffic changes.

Energy efficiency is a significant and challenging task for wireless sensor networks. It is true that our proposal needs more computational cost than PBR-no-ctrl for calculating control inputs. However, the computational cost for each sensor node still be very low since sensor nodes, non-controllable nodes, act on local information and simple rules. Moreover, our proposal can achieve load balancing by setting the flow matrix $\boldsymbol{F}$ such that each sink node receives data packets at the same rate as other sink nodes, which we explain in Section 4.3. Finally, our proposal does not depend on specific MAC-layer protocols. Therefore, we can reduce the energy required for data transfer between adjacent nodes by introducing energy-efficient MAC-layer protocols, such as IRDT.

The values of parameters $(\alpha, \beta, \sigma, r)$ are set to $(0.4, 0.2, 0.1, 10)$. The optimal parameters $(\alpha, \beta, \sigma)$ for PBR-no-ctrl are given in [Sheikhattar and Kalantari 2014], but potentials of nodes diverge to infinity and do not converge. Therefore, we use the settings above even for PBR-no-ctrl. All results presented below are averaged over $30$ simulation runs for each parameter setting.

### 4.3. Performance Evaluation of Hierarchical Control Method with Reduced-order Model

First, we evaluate the convergence speed of potentials after traffic changes considering the constraints in wireless sensor networks. To reveal the performance and properties of our proposal, we consider the case where the data generation rates of a partial set of sensor nodes change once, simultaneously.

Figure 3 shows the network model with $309$ nodes (including $9$ sink nodes) that is used in this evaluation. In this figure, sink nodes (resp., sensor nodes) are illustrated with squares (resp., dots). As shown in the figure, each sub-network is connected to a sub-controller via the sink node, whereas the central controller has direct connections

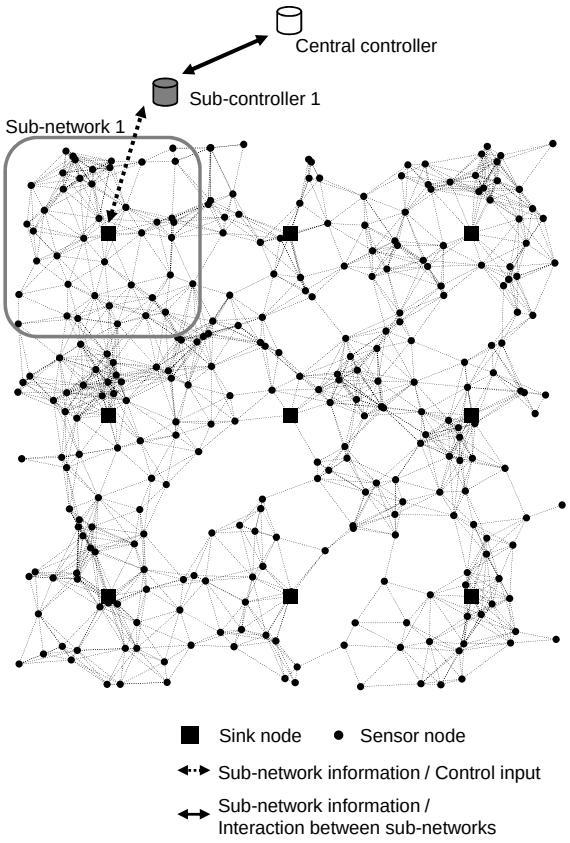



Fig. 3. Network topology ($N = 309$)

Table II. Observable nodes in each sub-network with PBR-h-opt and PBR-h-opt-mr.

| Sub-network No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| # of nodes | 43 | 39 | 38 | 51 | 12 | 25 | 56 | 16 | 29 | 309 |
| # of observable nodes | 10 | 17 | 15 | 16 | 3 | 9 | 22 | 10 | 14 | 116 |

with all sub-controllers. Observable range $p$ of sub-controllers is set to $2$. Details of observable nodes are shown in Table II.

The data generation rates are initially set to be $0.050$ packets/step for all sensor nodes. At $200$ steps from the beginning of the simulation, the data packet generation rates of sensor nodes are changed to examine the convergence speed of our proposed mechanism. After the traffic changes, the data packet generation rates are increased to $0.075$ packets/step for a partial set $\mathcal{N}_{inc}$ ($\in \mathcal{N}^1_{sen}$) of sensor nodes included in sub-network 1. The number of nodes whose data generation rates increase is set to $20$ (i.e., $|\mathcal{N}_{inc}| = 20$). Note that $\mathcal{N}_{sen}$ and $\mathcal{N}_{sin}$ are the set of sensor nodes and that of sink nodes, respectively. $\mathcal{N}^i_{sen}$ corresponds to the set of sensor nodes in sub-network $i$. The initial data generation rate of sensor nodes ($0.050$ packets/step) corresponds to $\bar{f}_n = -1.0$; therefore, before traffic changes, the flow rate vector $\boldsymbol{F} = \left[\bar{f}_1 \ \cdots \ \bar{f}_N\right]^T$ is given by

$$\bar{f}_n = \begin{cases} -1.0, & \text{if } n \in \mathcal{N}_{sen} \\ \frac{1.0\times|\mathcal{N}_{sen}|}{|\mathcal{N}_{sin}|}, & \text{if } n \in \mathcal{N}_{sin} \end{cases} . \tag{16}$$

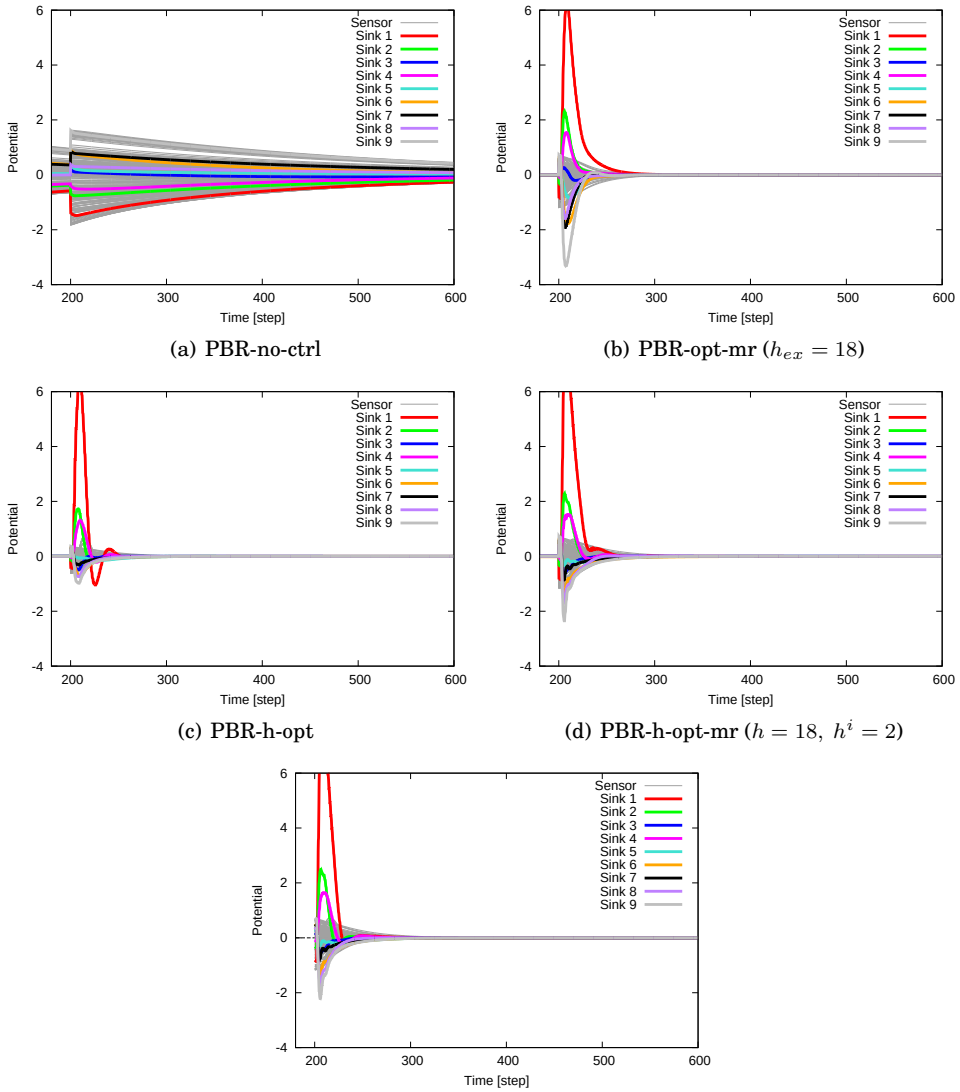
We construct the potential fields such that all sink nodes can receive data packets equally because load balancing is known to be a challenging task for wireless sensor networks. For the purpose, the flow rate at each sink node is ideally given by $\frac{1.0\times|\mathcal{N}_{sen}|}{|\mathcal{N}_{sin}|}$ $(= 25)$. Similarly, after the traffic changes, $\boldsymbol{F}$ is given by

$$\bar{f}_n = \begin{cases} -1.0, & \text{if } n \in \mathcal{N}_{sen} - \mathcal{N}_{inc} \\ -1.5, & \text{if } n \in \mathcal{N}_{inc} \\ \frac{1.0\times(|\mathcal{N}_{sen}|-|\mathcal{N}_{inc}|)+1.5\times|\mathcal{N}_{inc}|}{|\mathcal{N}_{sin}|}, & \text{if } n \in \mathcal{N}_{sin} \end{cases} . \tag{17}$$

Because the data generation rates of sensor nodes included in $\mathcal{N}_{inc}$ increase, the flow rate at each sink node also increases to $\frac{1.0\times(|\mathcal{N}_{sen}|-|\mathcal{N}_{inc}|)+1.5\times|\mathcal{N}_{inc}|}{|\mathcal{N}_{sin}|}$. In the evaluation of PBR-h-opt-mr, we set $h^i$ to the same value ($h_{sub}$) for all $i \in \mathcal{S}$. The degrees $(h, h_{sub})$ of the reduced order models for controllers are set to $(18, 2)$ and $(27, 3)$. Although it is difficult to quantitatively express the relation between $h$ and $h_{sub}$ (we note that $h = |\mathcal{S}|\cdot h_{sub}$ is not always satisfied), $h$ is typically larger than $h_{sub}$ because the central controller manages a larger amount of information than each sub-controller. The degree $h_{ex}$ of the external controller in PBR-opt-mr is set to $18$.

Figure 4 shows the changes in the potential values of PBR-no-ctrl, PBR-opt-mr, PBR-h-opt, and PBR-h-opt-mr. More precisely, this figure shows a plot of $\boldsymbol{X}(s) = \bar{\boldsymbol{\Theta}} - \boldsymbol{\Theta}(s)$ against step $s$, and potential convergence is achieved when each element of $\boldsymbol{X}(s)$ is sufficiently close to $0$, that is, when Equation (15) is satisfied. In this figure, thick colored lines correspond to potential changes of the $9$ sink nodes and thin gray lines correspond to potential changes of sensor nodes. Sink node potentials change more than those of sensor nodes because they receive feedback inputs $\boldsymbol{u}$ from the sub-controllers, whereas sensor nodes are indirectly affected by them via sink nodes.

As shown in Figure 4, the convergence speed of potentials is enhanced by introducing optimal feedback mechanisms, including non-hierarchical and hierarchical mechanisms. This proves the effectiveness of optimal feedback in a hierarchical manner. Our result indicates that sub-controllers can correctly estimate node potentials in their corresponding sub-network even though they do not directly observe nodes outside their corresponding sub-network. Moreover, it is worth mentioning that each sub-controller of PBR-h-opt and PBR-h-opt-mr observes node potentials within at most $2$ hops of the sink nodes in its corresponding sub-network. Sub-controller $i$ estimates the potentials of non-observable and future potentials of nodes in its corresponding sub-network using feedback inputs $\boldsymbol{Z}^i$, which are provided from the central controller to each sub-network, and then determines the optimal feedback inputs $\boldsymbol{u}^i$. Table III shows the time needed from traffic changes until the potential convergence is achieved (with $\delta = 0.01$). Figure 4 shows potential changes only within $180$–$600$ steps; however, it takes $1484.0$ steps for the potential to converge with PBR-no-ctrl. From the result, compared with PBR-no-ctrl, potential convergence is accelerated around $10.8$ fold with PBR-opt-mr, $10.6$ fold with PBR-h-opt, and $7.97$ and $8.93$ fold with PBR-h-opt-mr with $(h, h_{sub}) = (18, 2)$ and $(h, h_{sub}) = (27, 3)$, respectively. The convergence speed of potentials with our proposed mechanisms (PBR-h-opt, PBR-h-opt-mr) is a bit slower than that with PBR-opt-mr. This is because each sub-controller directly collect and estimate the potential values of only the nodes in the corresponding sub-network, and they receive the potential information about other sub-networks only via the central controller.




(a) PBR-no-ctrl

(b) PBR-opt-mr ($h_{ex} = 18$)

(c) PBR-h-opt

(d) PBR-h-opt-mr ($h = 18,\ h^i = 2$)

(e) PBR-h-opt-mr ($h = 27,\ h^i = 3$)

Fig. 4. Potential convergence in the case with a large-scale network

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

| Scheme | Convergence time [step] |
|---|---|
| PBR-no-ctrl | 1484.0 |
| PBR-opt-mr ($h_{ex} = 18$) | 137.61 |
| PBR-h-opt | 140.47 |
| PBR-h-opt-mr ($h = 18,\ h^i = 2$) | 186.18 |
| PBR-h-opt-mr ($h = 27,\ h^i = 3$) | 166.18 |

The computational cost $O({h^i}^2)$ of each sub-controller for estimating the potential dynamics and calculating control inputs is much smaller than the computational

Table IV. Computational time for each scheme

| Scheme | Controller Type | Computational time [s] | | |
|---|---|---|---|---|
| | | Controller design (*dlqr* or *dhinflmi*) | Model reduction (*balred*) | Calculation of feedback inputs |
| PBR-no-ctrl | – | 0 | 0 | 0 |
| PBR-opt-mr ($h_{ex} = 18$) | External | 6398 | 4.084 | $6.313 \times 10^{-5}$ |
| PBR-h-opt | Sub | 0.04003 | 0 | $9.360 \times 10^{-5}$ |
| | Central | 30.75 | 0 | $1.798 \times 10^{-3}$ |
| PBR-h-opt-mr ($h = 18, h_{sub} = 2$) | Sub | 0.08880 | 0.04311 | $3.853 \times 10^{-6}$ |
| | Central | 35.38 | 2.196 | $3.815 \times 10^{-5}$ |
| PBR-h-opt-mr ($h = 27, h_{sub} = 3$) | Sub | 0.09006 | 0.04396 | $4.125 \times 10^{-6}$ |
| | Central | 34.35 | 2.428 | $4.810 \times 10^{-5}$ |

cost $O(h^2)$ of the external controller of PBR-opt-mr proposed in [Kuze et al. 2014], where only one controller monitors/controls the network. This is because the number of state variables that suffices to describe the network/sub-network is generally larger if the number of nodes in the network/sub-network is larger. Moreover, given the information about the network topology and flow rates, the central controller is designed with the computational cost $O(h^2)$ as described in Section 3.4, which is much smaller than the computational cost $O(h^3)$ needed for designing the external controller proposed in [Kuze et al. 2014]. Table IV shows the computational time required for the controller design, model reduction, and calculation of feedback inputs. The computational time for the controller design and the model reduction results from one simulation run for each scheme are shown, as measured by using the *timeit* function in MATLAB. The computational time for the calculation of feedback inputs is averaged over $30$ calculations of $\boldsymbol{u}$ or $\boldsymbol{u}^i$ for each scheme and measured by using the *QueryPerformanceCounter* function in C++. The computational time for the sub-controller corresponds to the average amount of time required for each sub-controller. As shown in Table IV, the computational time for the controller design with our proposed mechanisms are much smaller than that with PBR-opt-mr. Moreover, the time required for the calculation of feedback inputs with our proposed mechanisms is a little smaller than that with PBR-opt-mr. It is worth mentioning that because feedback inputs are calculated at every step, the difference in the computational time for each step is small but makes a good contribution to the reduction in the computational time in the long run. Therefore, our proposed mechanisms can enhance the convergence speed of potentials with low computational cost, even in large-scale networks. Consequently, the hierarchical optimal feedback mechanism is more scalable than the non-hierarchical one.

In general, as degree $h^i$ of the reduced-order model becomes smaller, the approximation error becomes larger while the computational cost becomes much smaller. Actually, as shown in Figures 4(c)–4(e) and Table III, the convergence speed of PBR-h-opt-mr with $h = 18$ and $h_{sub} = 2$ is a bit slower than that of PBR-h-opt and PBR-h-opt-mr for higher values of $h$ and $h_{sub}$. Consequently, there is a trade-off between the computational cost and the performance of our proposal.

The estimation errors of the potentials cannot be avoided completely because of disturbances, modeling errors of the system dynamics, and so on. Specifically, in our evaluations, nodes are not synchronized against the potential dynamics described by Equation (2) and the potential values the controller collects are not always correct owing to communication delays, dropped data, or interference. Our evaluations prove that our proposal can work even when there are such estimation errors. In this evaluation, a partial set of control messages drop because traffic congestion occurs around sink nodes. This occurs because the external controller/sub-controllers collect the network information via sink nodes in potential-based routing with optimal feedback. Control messages are sent by sensor nodes within $p$ hops from sink nodes for sending potential

information to sub-controllers. If control messages drop, sub-controllers cannot receive potential information, which leads to errors in the estimation of potentials. Moreover, the asynchrony of the controller and nodes is also a cause of estimation errors because PBR-h-opt and PBR-h-opt-mr, as well as PBR-opt-mr, inherently assume 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 disturbances exist.

In Sections 4.3, we have shown that a hierarchical optimal control by the central controller and sub-controllers is effective in wireless sensor networks where the capacity and energy of each node are limited. Moreover, PBR-h-opt-mr enhanced the convergence speed of potentials with much lower computational cost than PBR-opt-mr. This 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.

### 4.4. Adaptability to Massive and Frequent Environmental Changes

Next, we evaluate the changes in potential values and traffics in cases where the data generation rates of all sensor nodes change several times, using this to demonstrate that our proposal can work even if there are frequent and massive environmental changes. We use the network model with $104$ nodes (including $4$ sink nodes) described in Figure 5 for this evaluation, and the data packet generation rates of sensor nodes in sub-network $1$ (resp., $3$) and that in sub-network $2$ (resp., $4$) are exchanged every $200$ steps. Each sub-controller collects potential information about nodes within $1$ hop from its connected sink node; that is, $p$ is set to $1$.

The data packet generation rates are initially set to be $0.025$ packets/step for sensor nodes in sub-networks $1$ and $3$ of Figure 5, and $0.075$ packets/step for sensor nodes in sub-networks $2$ and $4$. After traffic changes at $200$ steps, the data packet generation rates are increased to $0.075$ packets/step for the left half sensor nodes and decreased to $0.025$ packets/step for the right half nodes. Subsequently, the data packet generation rates of sensor nodes in sub-network $1, 3$ and that in sub-network $2, 4$ are exchanged at interval $200$ step. The average data generation rate of a node of $0.050$ packets/step corresponds to $\bar{f}_n = -1.0$, and therefore, the flow rate vector $\boldsymbol{F}$ during the first $200$ steps is given by

$$\bar{f}_n = \begin{cases} -0.5, & \text{if } n \in \mathcal{N}^1_{sen} \cup \mathcal{N}^3_{sen} \\ -1.5, & \text{if } n \in \mathcal{N}^2_{sen} \cup \mathcal{N}^4_{sen} \\ \frac{0.5\times(|\mathcal{N}^1_{sen}|+|\mathcal{N}^3_{sen}|)+1.5\times(|\mathcal{N}^2_{sen}|+|\mathcal{N}^4_{sen}|)}{|\mathcal{N}_{sin}|}, & \text{if } n \in \mathcal{N}_{sin} \end{cases}. \tag{18}$$

Similarly, $\boldsymbol{F}$ during the next $200$ steps is given by

$$\bar{f}_n = \begin{cases} -1.5, & \text{if } n \in \mathcal{N}^1_{sen} \cup \mathcal{N}^3_{sen} \\ -0.5, & \text{if } n \in \mathcal{N}^2_{sen} \cup \mathcal{N}^4_{sen} \\ \frac{1.5\times(|\mathcal{N}^1_{sen}|+|\mathcal{N}^3_{sen}|)+0.5\times(|\mathcal{N}^2_{sen}|+|\mathcal{N}^4_{sen}|)}{|\mathcal{N}_{sin}|}, & \text{if } n \in \mathcal{N}_{sin} \end{cases}. \tag{19}$$

Figure 6 shows the changes in the potential values of PBR-no-ctrl, PBR-opt-mr with $h_{ex} = 8$, and PBR-h-opt-mr with $h = 8$ and $h^i = 2$. Specifically, Figure 6 shows a plot of $\boldsymbol{X}(s)$ against step $s$. As shown in Figure 6, the hierarchical control feedback mechanism can enhance the convergence speed of potentials, as compared with the case of the non-control scheme.

Figure 7 shows changes in the number of data packets delivered to each sink node every $20$ steps. In each case, the number of data packets delivered to each sink node becomes disproportionate after the traffic changes at $200$ steps. Then sink nodes gradually become able to receive data packets equally, because potentials are updated to

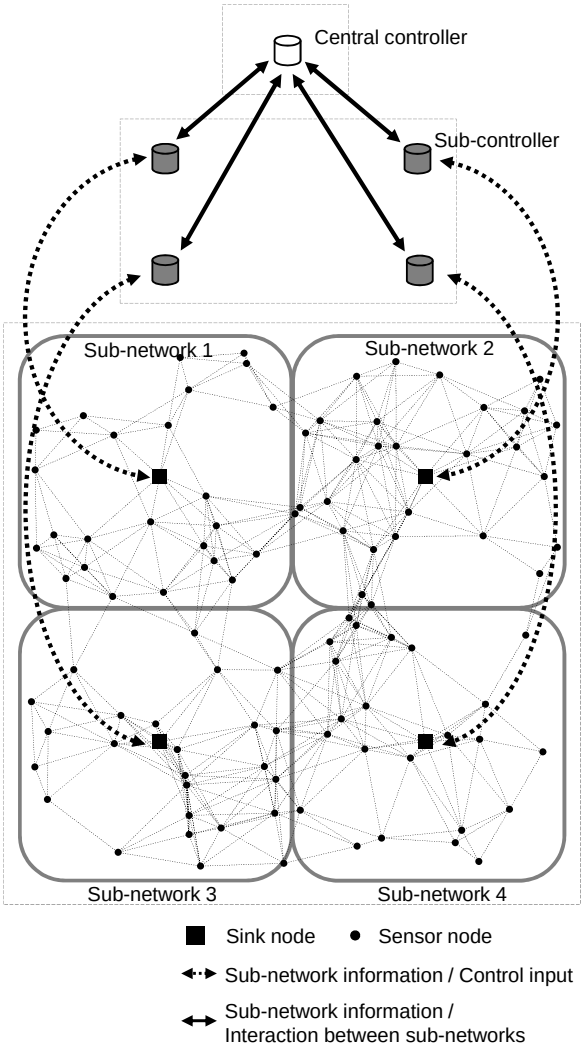
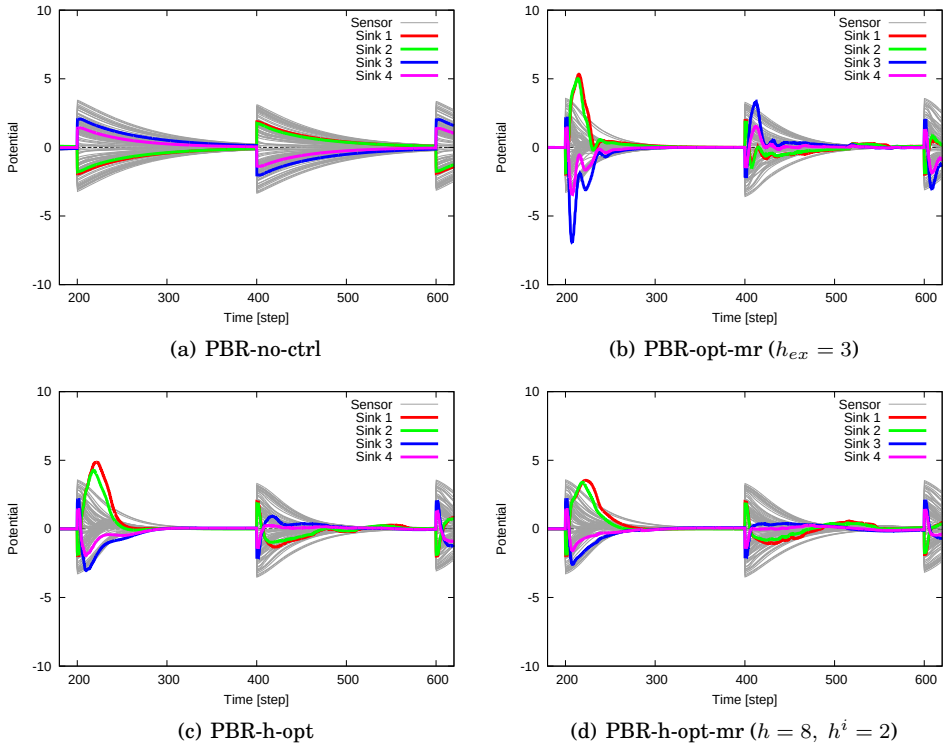



Fig. 5. Network topology ($N = 104$)

adapt to the current packet rate. We can observe that the traffic convergence speed is also accelerated by optimal feedback. This is because the potential convergence speed is enhanced by the optimal feedback mechanism. As shown in Figures 6(a) and 7(a), as the convergence speed of potentials is too low, traffic cannot adapt to frequent and massive changes with PBR-no-ctrl. In contrast, by introducing optimal control mechanisms, the adaptability to adapt to such changes is improved (Figures 6(b)–6(d) and 7(b)–7(d)).

One problem we note 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, data packets will drop when the controller makes large changes to the po-




(a) PBR-no-ctrl

(b) PBR-opt-mr ($h_{ex} = 3$)

(c) PBR-h-opt

(d) PBR-h-opt-mr ($h = 8,\ h^i = 2$)

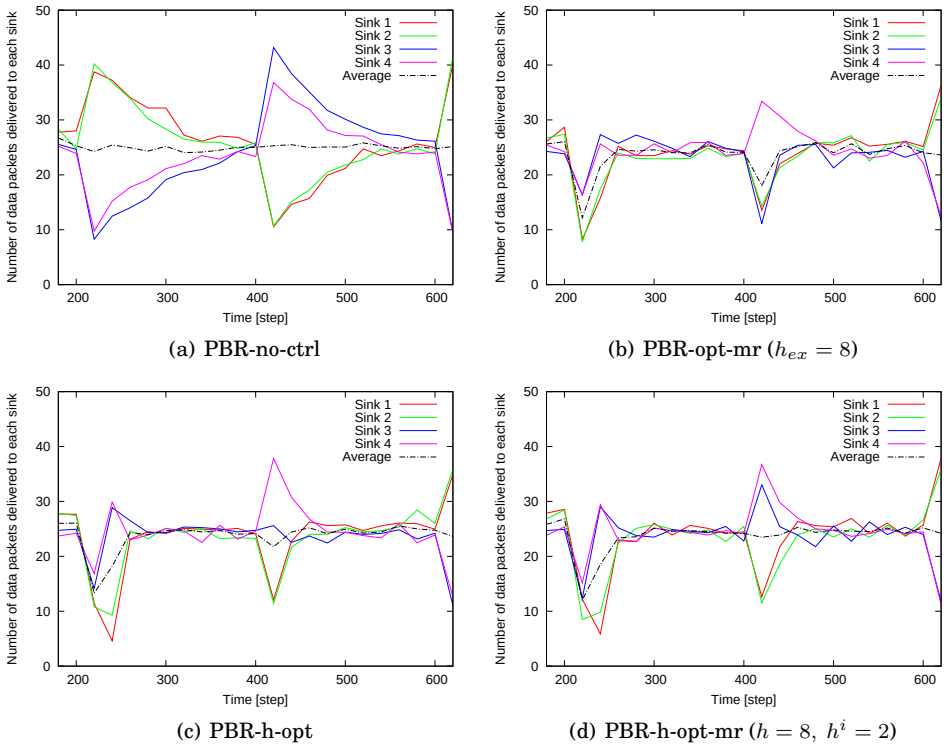
Fig. 6. Potential convergence in case of traffic changes

tentials, 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 in which 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(b)–7(d) show worst-case scenarios for temporal packet drops, because the controller changes sink node potentials (in other words, data packet destinations), and therefore, many data packets are dropped when a sink node temporarily gets the highest potential within its communication range owing to control inputs. If the sub-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 as the sub-controller can make large changes to the potentials, whereas the recovery speed of data packets delivered to each sink node increases. This indicates that there is trade-off between the convergence speed of potentials and potential fluctuations.

In Section 4.4, we have shown that our proposal can adapt to frequent and massive environmental changes.

### 4.5. Integration of Two Different Networks

We finally evaluate how our proposal works when a new sub-network is added to the network. The network model shown in Figure 5 is used in this evaluation. First, the network only consists of nodes of sub-networks $1$, $2$, and $3$. Then, at $200$ steps from




(a) PBR-no-ctrl (b) PBR-opt-mr ($h_{ex} = 8$)

(c) PBR-h-opt (d) PBR-h-opt-mr ($h = 8,\ h^i = 2$)

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

the beginning of the simulation, sub-network $4$ is added to the network. To evaluate the influence of the addition of a new sub-network, and not that of traffic changes, the data packet generation rates of all sensor nodes are fixed at $0.050$ packets/step, and therefore, $\boldsymbol{F}$ before the addition of a new sub-network is given by

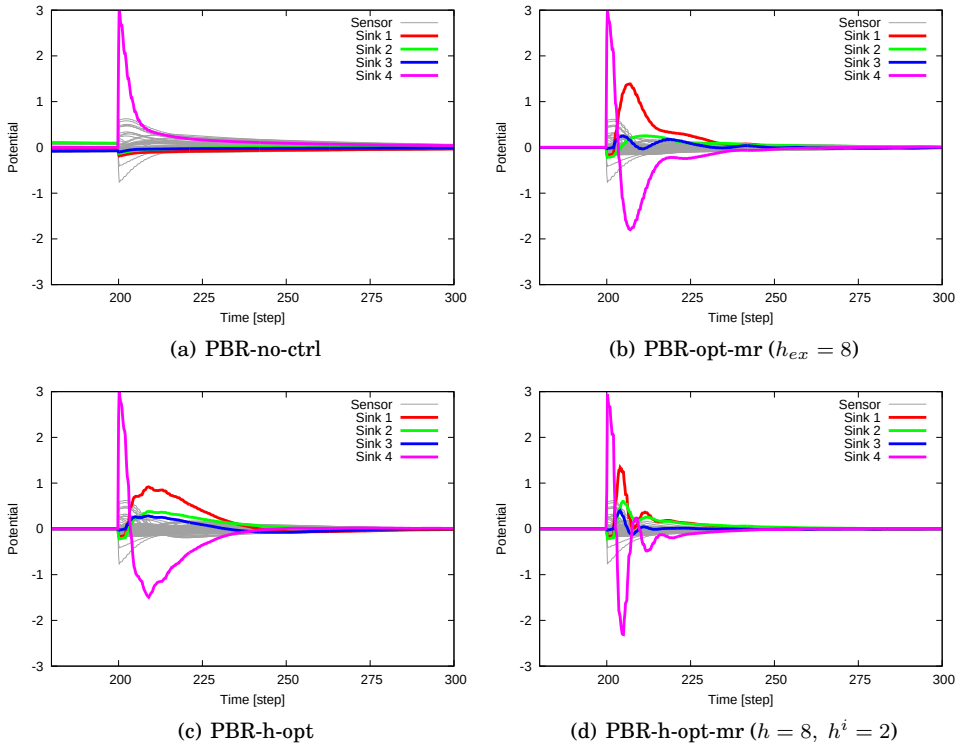
$$\bar{f}_n = \begin{cases} -1.0, & \text{if } n \in \mathcal{N}^1_{sen} \cup \mathcal{N}^2_{sen} \cup \mathcal{N}^3_{sen} \\ \frac{1.0 \times |\mathcal{N}^1_{sen}| + |\mathcal{N}^2_{sen}| + |\mathcal{N}^3_{sen}|}{|\mathcal{N}^1_{sin}| + |\mathcal{N}^2_{sin}| + |\mathcal{N}^3_{sin}|}, & \text{if } n \in \mathcal{N}^1_{sin} \cup \mathcal{N}^2_{sin} \cup \mathcal{N}^3_{sin} \end{cases} . \tag{20}$$

Then, $\boldsymbol{F}$ after the addition of a new sub-network is given by

$$\bar{f}_n = \begin{cases} -1.0, & \text{if } n \in \mathcal{N}_{sen} \\ \frac{1.0 \times |\mathcal{N}_{sen}|}{|\mathcal{N}_{sin}|}, & \text{if } n \in \mathcal{N}_{sin} \end{cases} . \tag{21}$$

In this evaluation, we set degrees $(h, h_{sub})$ to $(6, 2)$ before the addition of a new sub-network at $200$ steps, whereas we set $(h, h_{sub})$ to $(8, 2)$ after $200$ steps with PBR-h-opt-mr. After the addition of a sub-network, we increase the degree $h$ of the estimation model for the central controller because the number of nodes in the entire network increases. With PBR-opt-mr, we set degree $h_{ex}$ to $6$, $8$ before and after the addition of a new sub-network, respectively. Other simulation settings are the same as in Section 4.4.

Figure 8 shows the potential changes with PBR-no-ctrl, PBR-opt-mr, PBR-h-opt, and PBR-h-opt-mr, and Table V shows the convergence time of potentials. The figure plots $\boldsymbol{X}(t)$ against step $s$. As the figure and table show, the convergence speed of potentials is improved by $1.87$ times with PBR-opt-mr, $2.02$ times with PBR-h-opt, and $2.75$ times




(a) PBR-no-ctrl (b) PBR-opt-mr ($h_{ex} = 8$)

(c) PBR-h-opt (d) PBR-h-opt-mr ($h = 8,\ h^i = 2$)

Fig. 8. Potential convergence in case of addition of new sub-network

Table V. Potential convergence time in the case of adding a new sub-network

| Scheme | Convergence time [step] |
|---|---|
| PBR-no-ctrl | 234.71 |
| PBR-opt-mr ($h_{ex} = 8$) | 125.33 |
| PBR-h-opt | 116.28 |
| PBR-h-opt-mr ($h = 8,\ h^i = 2$) | 85.500 |

with PBR-h-opt-mr. This indicates that both the hierarchical optimal feedback mechanism and the non-hierarchical one can enhance the convergence speed in cases where a new sub-network is added to the network.

The dynamics of potentials depends on the network topology, and therefore, the estimation model of the dynamics also depends on the network topology. In other words, unlike cases of traffic changes (Subsections 4.3 and 4.4), the estimation model needs to be redesigned when topological changes occur. With PBR-opt-mr, the estimation model needs to be redesigned entirely and the computational cost for redesigning is given by $O(h^3)$. On the contrary, with PBR-h-opt and PBR-h-opt-mr, the estimation model is designed for each sub-network. Therefore, only the estimation model of the sub-network whose topology changes needs to be redesigned and the computational cost is given by $O(N^{i3})$ and $O(h^{i3})$ with PBR-h-opt and PBR-h-opt-mr, respectively. The degree of the approximation model for a sub-network is generally smaller than that for the entire network. Consequently, the computational cost for redesigning the estimation model

due to topological changes is smaller with the hierarchical optimal feedback mechanism (PBR-h-opt-mr) than that with the non-hierarchical one (PBR-opt-mr).

In conclusion, hierarchical optimal feedback by several controllers can work with low computational cost even in the case of topological changes in the network. Specifically, owing to the high adaptability to the integration of several networks as shown in this evaluation, our proposal can be adapted to much larger-scale networks.

## 5. 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 hierarchical optimal feedback using a reduced-order model, in which several controllers monitor and estimates system states, and provide optimal feedback in a hierarchical manner. Simulation results have shown that hierarchical 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 because the computational cost is much smaller than in the non-hierarchical scheme.

However, some challenges still need to be overcome. First, the optimal feedback mechanism assumes that the controller has information about the network topology and 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, and therefore, the potential cannot converge if environmental changes occur more frequently than the iterative behavior. Third, the controllability and stability of the networks depends on the network topology. Finally, the optimal control improves the convergence speed of potentials but also causes potential fluctuations, as shown by the simulation results. These fluctuations lead to data packet drops because sink nodes temporally have the highest potentials between their neighbors. There is a trade-off between the improvement of the potential convergence speed and the potential fluctuations.

We are now studying a predictive control method that can adapt to the dynamically changing network. For this purpose, it would be a promising direction to design a controller based on Bayesian inference.

## ACKNOWLEDGMENTS

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

## REFERENCES

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.

Shiníchi 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.

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

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).

J Michael Herrmann, Michael Holicki, and Ralf Der. 2004. On Ashby ' s homeostat: A formal model of adaptive regulation. *From animals to animats* (2004), 324–333.

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.

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. 2014. Enhancing Convergence with Optimal Feedback for Controlled Self-organizing Networks. In *Proceedings of IEEE 80th Vehicular Technology Conference*. 1–7.

Naomi Kuze, Daichi Kominami, Kenji Kashima, Tomoaki Hashimoto, and Masayuki Murata. 2015. Hierarchical optimal control method for controlling self-organized networks with light-weight cost. to be presented in IEEE GLOBECOM 2015. (Dec. 2015).

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).

Georg Martius and J Michael Herrmann. 2010. Taming the beast: Guided self-organization of behavior in autonomous robots. In *In Proceedings of International Conference on Simulation of Adaptive Behavior*. Springer, 50–61.

Christian Müller-Schloer, Hartmut Schmeck, and Theo Ungerer. 2011. *Organic computing-a paradigm shift for complex systems*. Birkhaeuser, Berlin.

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

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

N. Sandell, P. Varaiya, M. Athans, and M. Safonov. 1978. Survey of decentralized control methods for large scale systems. *IEEE Trans. Automat. Control* 23, 2 (April 1978), 108–128. DOI:http://dx.doi.org/10.1109/TAC.1978.1101704

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

Received xxxx xxxx; revised xxxx xxxx; accepted xxxx xxxx