A Dual Control Approach for Indirect Configuration Propagation with Energy-Efficient Scheduling in Multi-agent Networking Systems

Heejung Byun[1] and Junglok Yu[2]
[1] Department of Information and Telecommunication Engineering, Suwon University
17, Wauan-gil, Bongdam-eup, Hwaseong-si, Gyeonggi-do, Korea
[2] Korea Institute of Science and Technology Information, Daejeon-si, Korea
Email: heejungbyun@suwon.ac.kr; junglok.yu@kisti.re.kr

Abstract —This paper presents a dual control approach for indirect system configuration propagation with energy-efficient agent scheduling. The proposed method influences the MNS (Multi-agent Networking Systems) operation by indirectly propagating the system configuration within the framework of local rules. Also, the proposed method adapts agent's operational state according to the convergence rate of configuration propagation in order to balance energy consumption among agents in the MNS. Finally, we propose an optimal timing control for sequent input. Using the operation state control model, the gateway agent determines the optimal timing to give next input based on the value of the operation state. Simulation results are performed to demonstrate the superiority of the proposed method and we observe that the proposed scheme is less susceptible to error and shows more robust performance than the consensus method in an error-prone environment.

Index Terms—Multi-agent networking system, configuration propagation, energy efficiency

I. Introduction

A Multi-agent Networking System (MNS) consists of a large number of agents that coordinate autonomously based on underlying control laws [1]-[12]. In a MNS, an operator directs agents to carry out mission goals or tasks. The resulting behaviors the MNS generates depend on a set of parameters of agent algorithms or choice of system parameters for their operation. In order to perform supervisory control of consensus behaviors, the operator should convey appropriate parameter adjustments or system configuration independent of the number of agents as intended goals change [13]-[15]. In addition, agent members in a MNS are generally battery-powered, so they can be easily depleted of energy if they remain on active while these controls are exerting to the MNS. This leads to the energy imbalance among agents and a shorter life-time of the MNS.

Existing approaches relay influence from the operator to the system by broadcasting the parameter change to all agents in the MNS. Walker *et al*. [16]-[17] focused on two methods of information propagation (flooding and consensus methods) and compared the ability of operators to manage the MNS to the desired goals. Goodrich et al. [18] worked on a leader-based control of systems using tele-operated leaders based on Couzin's control laws. Pendleton *et al*. [19] similarly implemented a leader-based model using both virtual agents and an operator as leaders in a system. In [20], the authors proposed a single-hop broadcast algorithm for new software downloading but agents that are too far from the user will not be reprogrammed. In [21], the authors proposed an architecture for multi-agent communication networks, in which agents are clustered to one or multiple systems and each system can be monitored by some central servers through a wireless mesh backbone. In this way, the existing studies have been addressed the interaction problems between operator and multi-agent system, but little work has focused on how the system configuration should be spread through the MNS via human-agent interactions while achieving energy efficiency of the system.

To address concerns, we propose a dual control approach for indirect propagation of system configuration with an energy efficient agent scheduling in the MNSs. First, we propose a method, which influences MNS operation by indirectly propagating the system configuration from the operator within the framework of local rules in the MNS. Second, we design a scheduling controller of agent state, in which each agent autonomously determines its operation state depending on the configuration propagation rate for saving energy while balancing energy consumption among agent members.

II. Proposed Algorithm

A. Configuration State Propagation Model

We consider a multi-agent system consisting of N nodes. Let $\boldsymbol{N}=\{1,2,\ldots,N\}$ denote the set of nodes in the system. The set of neighboring nodes of node i is denoted as $\boldsymbol{N}_\mathbf{i}$ and the number of neighboring nodes is N_i. The N-th node is a gateway node. We denote $u(t)$ as the desired

Manuscript received April 30, 2019; revised August 6, 2019.
This work was supported by the National Research Foundation of Korea under Grant No. 2017R1A2B1007779.
Corresponding author email: heejungbyun@suwon.ac.kr.
doi:10.12720/jcm.14.9.773-778

system configuration issued by the operator, such as command inputs, parameter setting, and parameter changes that operator wants to achieve with the MNS at time t. Let the vector of the configuration states of all agents at time t as $x(t) = [x_1(t)\ x_2(t)\ \ldots\ x_N(t)]^T$ where $x_i(t) = [x^1_i(t)\ x^2_i(t)\ \ldots\ x^n_i(t)]^T$ is the configuration state vector of agent i at time t and individual agent has n-dimensional configuration state space. The operator interacts with the MNS by applying the desired configuration input to the gateway agent, while the other agents generate their respective configuration state vectors and propagate them by interacting with each other following the local control law of each agent in the MNS:

$$\dot{x}_i(t) = \sum_{j\in\mathbf{N}_i}(x_j(t)-x_i(t)), \forall i, i \neq N. \qquad (1)$$

The operator gives commands or informs the gateway agent of desired configuration sets and parameter changes with $u(t)$, after which the user input is transformed into configuration state vector and propagated to the MNS according to (1).

B. Operation State Scheduling Model

When the system configuration issued by the operator is spread throughout the MNS, an important aspect to consider is to keep the energy consumption of agent members balanced. In this section, we propose an autonomous operation state control model for each agent based on its own configuration state proposed in the previous section. We consider two operational modes of each agent: active and sleep. In the active mode, an agent works normally, sampling and communicating with its environment. When the agent is in the sleep mode, the radio modules are not in use which helps save its energy. The operation state control model focuses on efficient power management by scheduling the operation mode of each agent. We denote the operation state of agent i as r_i, which is determined by the following rule:

$$\dot{r}_i(t) = -\epsilon_r(r_i(t)-r_{\min}) + \frac{\left|\sum_{j\in\mathbf{N}_i}(x_j(t)-x_i(t))\right|}{\sum_{j\in\mathbf{N}_i}x_j(t)} \qquad (2)$$

where ϵ_r is the control parameter to be chosen and $r_{\min}$ is the minimum activation probability value of agents that can be set by the operator, which makes the minimum number of agents activate. When agent i determines r_i, it independently generates a random value following the uniform distribution within [0, 1]. If the value of r_i is less than the random value, then the agent goes to sleep; otherwise, if it is greater than the random value, the agent becomes active by turning on its sensing circuitry. The higher value of r_i results in a higher probability of being active. According to (2), the operation state model works in such as way that the minimum activation probability is achieved when agent i reaches an equilibrium point, where the state of the agent's configuration converges to the user command $u(t)$. As the configuration state of agent i is closer to the averaged state among its neighbor agents, the value of r_i goes near to zero, which results in a lower probability of being active. It means that as the configuration state of an agent becomes identical to those of its neighbors, the value of r_i decreases and the agent enters sleep state more frequently, resulting in energy savings.

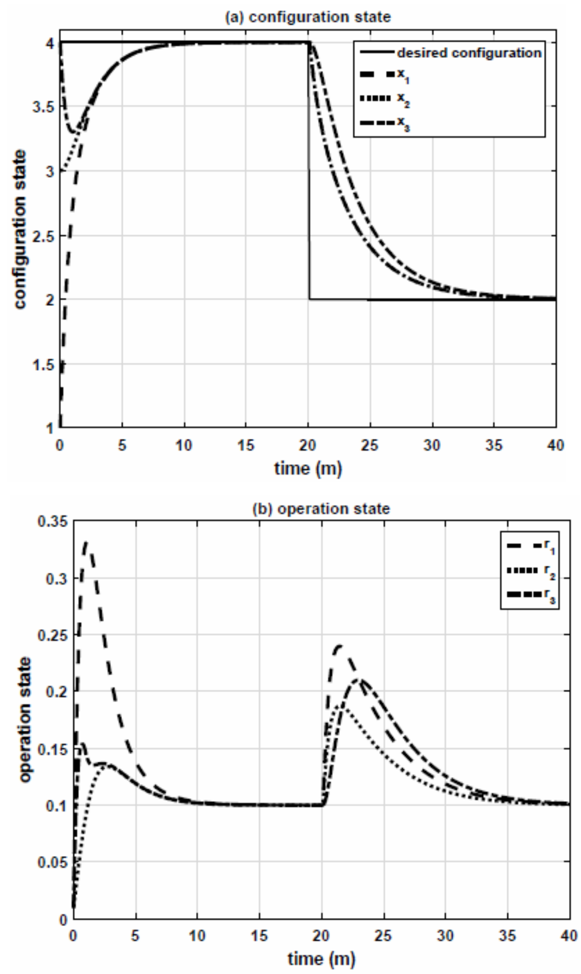

Fig. 1. Numerical results: (a) configuration state and (b) operation state

Fig. 1 shows the numerical results of the proposed controller given by (1) and (2). We consider four agents in a MNS. The configuration and operation states of the agents except for the gateway one are denoted as x_1, x_2, x_3 and r_1, r_2, r_3, respectively. We set the initial values of x_1, x_2, x_3 to 1, 3, 4, respectively. The user input is initially set to u = 4 and changed to u = 2 at 20 min. The minimum activation probability value is set to $r_{\min}$ = 0.1. As shown in Fig. 1 (a), the configuration states of agents change according to the user command and are successfully converged to the user command even though each agent starts with different initial state. For the operation states as shown in Fig. 1 (b), the values of r_1, r_2, r_3 converge to the pre-determined value of $r_{\min}$ at the steady state. When the user input changes in 20 min, the operation states of the agents are also adjusted and finally converged to the value of $r_{\min}$. According to these results, we can see that the operation state is affected by the convergence rate of the configuration state so that when the configuration state approaches user command $u(t)$, the operation state is also close to the minimum activation probability $r_{\min}$.

C. Optimal Timing Control

Another important function is to estimate the system state so that the operator can change or properly give sequent control input to the system. To support this feature, we present an optimal timing control for sequent input. Based on the operational state control model, the gateway agent determines the optimal time to give next input based on the value of operation state of the gateway agent. The proposed dynamic is as follows:

$$r_N(t+1) = r_i(t) - \epsilon_r(r_N(t) - r_{\min}) + \frac{\left\| \sum_{j\in \mathbf{N}_N} (x_j(t) - x_N(t)) \right\|}{\left\| \sum_{j\in \mathbf{N}_N} x_j(t) \right\|} \quad (3)$$

The gateway agent provides a feedback of the operation state to the operator. According to the above stability analysis, we know that when the states of all agents converge to the desired configuration in the steady state that drives the system to $r_N = r_{\min}$. Therefore, the operator can be able to learn to understand system states and estimate the timing to give the next control inputs, given the feedback from the gateway agent only. We define r_D:

$$r_d = r_{\min} - r_N \quad (4)$$

Then, the operator gives a new input to the system when the value of r_D is smaller than a predefined constant, which is the tolerance for convergence.

III. Simulation Results

A. Configurations

To evaluate the performance of our proposed MNS model, we developed a simulation environment using MATLAB. We compared the performance of the proposed method with the consensus method [16], [22]. The consensus algorithm is an asynchronous distributed protocol for distributed averaging, which aims to compute the average values. Each member in a system averages the values of all of its neighbors, and adapts its own value toward that average. Both the proposed approach and the consensus method have distributed nature for data propagation and require only simple iterative computation, as such the consensus algorithm can serve as a baseline for the comparison. The performance comparison is based on four metrics:

- Configuration state: The system configuration state vector controlled by each agent.
- Operation state: The operation state controlled by each agent.
- Energy difference ratio: The energy consumption difference between the agent with the highest energy consumption ratio ($E_{\max}$) and the agent with lowest one ($E_{\min}$) in the MNS. The energy consumption ratio of each agent is the ratio of the agent's consumed power to the initial power E_0. Then, the energy difference ratio is evaluated as

$$E_{\max} - E_{\min}$$

The energy difference ratio approaching zero means that energy balance is achieved.

- Remained energy ratio: The available energy of the agent with the highest energy consumption rate ($E_{\max}$) in the MNS, which is represented as

$$(1 - E_{\max}) \times 100\ (\%)$$

The higher remained energy ratio means the longer lifetime of the MNS.

The simulation area is 100m × 100m, where the entire network is divided into equally shaped grids, and the agents are uniformly deployed. We set N = 50, and the agent members are arbitrary connected. The gateway agent is denoted as R_{50}, which is chosen randomly by the operator. The channel capacity is set to 200 kbps, the transmission range and carrier sense range to 20m and 40m, respectively. The current consumption for Tx, Rx, and mode switch are set to 17.4mA, 19.7mA, and 10.05mA, respectively. The mode switch time and back-off time are set to 300s and 30ms. The parameters of the function in (2) are set to $\epsilon_r = 1$ and $r_{\min} = 0.1$, respectively.

B. Results

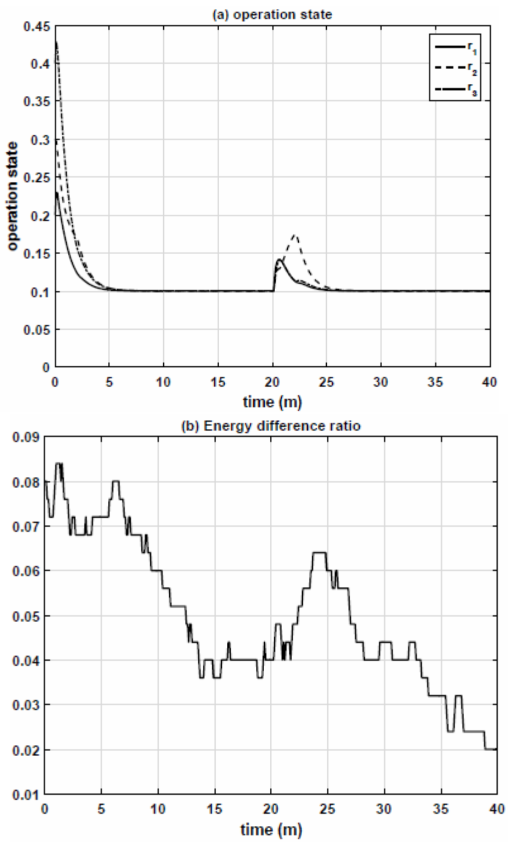

Fig. 2. Operation state adaptation and energy difference ratio for the proposed method: (a) operation state and (b) energy difference ratio.

Fig. 2 (a) shows the operation state adaptation behavior for the proposed method. We set the initial values of r_1, r_2,

r_3 to 0.2, 0.3, 0.4, respectively. From the result, it can be seen that the operation state of each agent is autonomously adjusted and converged to the pre-defined value of r_{min} even though it starts from different initial operation states. Also, the operation state is automatically adjusted and stabilized according to the change of the configuration state and user input as well. Fig. 2 (b) shows the energy difference ratio for the proposed method. After the first user control input is issued, the energy difference ratio approaches zero as the operation state stabilizes. When the next control input is applied in 20 min and the operation state is automatically adjusted, the energy difference increases slightly and then gradually decreases to zero. It means that when the operation state is stabilized, the energy difference converges toward zero, which indicates that the energy balance is achieved because the highest energy consumption ratio and the lowest one are almost identical.

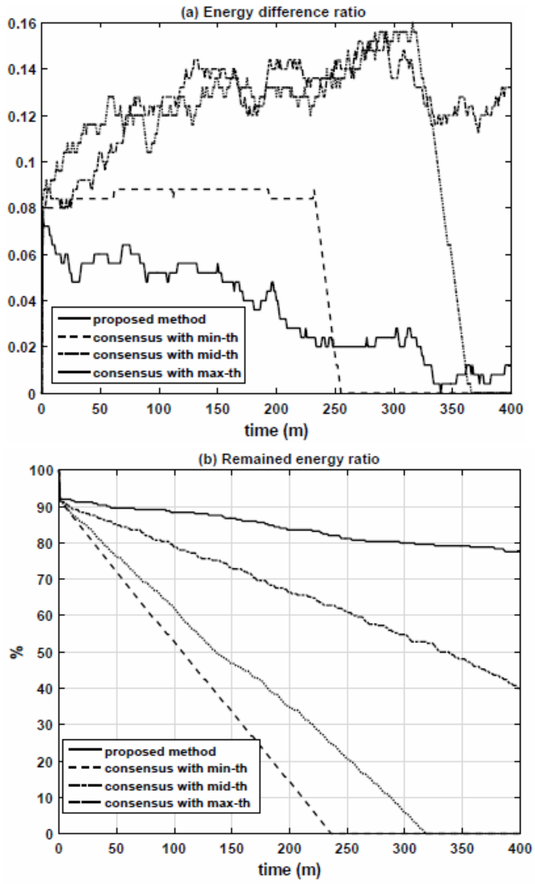

Fig. 3. Energy efficiency: (a) energy difference ratio difference between the agent with the highest energy consumption ratio and the agent with lowest one in the MNS and (b) remained energy ratio in the MNS.

To show the performance of energy efficiency and energy balancing, we set the initial power (E_0) for each differently. We apply the same error condition randomly for all agents in the MNS. Fig. 3 (a) shows the energy difference between agents with the proposed scheme and the consensus scheme. In the case of the consensus method, there is no appropriate mechanisms for operation adaptation to schedule agent's state, so we enable the agents to be activated stochastically.

Each agent randomly generates a number and compares it with a given threshold. If the probability value is larger than the threshold, the agent becomes activated. In this simulation, the threshold values are set to 0.3 (mid-th), 0.02 (min-th), and 0.7 (maxth), respectively. For the metric of energy difference ratio, the smaller the energy difference is, the more the energy balancing is achieved. As shown in Fig. 3 (a), the proposed method shows that the energy difference ratio decreases over time and goes near to zero after 350 min.

It means that the energy consumption ratio is balanced among the agents in the MNS. However, in the case of the consensus method, the energy difference ratio is maintained or becomes larger. In the plot of the consensus method with min-th, the energy difference ratio suddenly drops to zero after 250 min, which means that the agents' batteries are exhausted. The energy depletion with min-th is occurred within a shorter time compared to other threshold values, which is caused by more frequent activation of agent. Fig. 3 (b) shows the remained energy ratio of the proposed and consensus schemes.

According to the result, the proposed method shows the highest available energy ratio, but for the consensus method, we observe that energy depletion occurs relatively early, which leads to a reduction in network lifetime. Also, the consensus method with min-th shows a much shorter network lifetime than with other threshold values.

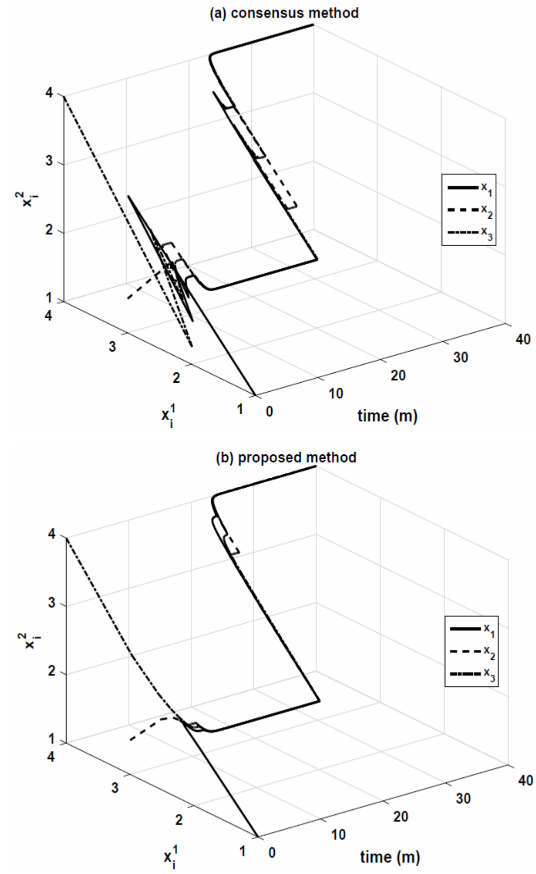

Fig. 4. Configuration state adaptation according to user input change during runtime: (a) consensus method and (b) proposed method.

Fig. 4 shows the configuration state adaptation behavior for consensus method and the proposed method according to control input changes. Both methods show that the configuration state values of all agents are adjusted according to the user control input changes. For the consensus method, however, the malfunction of agent 2 directly affects the state values of the neighboring agents. In the consensus method, the state value of each agent is sensitive to changes in its neighbor agents' states, so there is a large difference in state values between agents. On the other hand, the proposed method shows that the wrong behavior of agent 2 does not have a significant effect on the state control of other agents. The configuration state change between the agents is not large despite the malfunction of agent 2 and it is adapted successfully according to the desired user input. This is because each agent indirectly uses the configuration state of neighboring agents in adjusting its configuration state values, which leads to be less susceptible to error conditions and more robust performance.

Based on these results, we can see that the proposed method automatically adjusts the operation state of each agent so that the energy consumption ration is balanced among the agents. Also, the proposed method can control unnecessary energy waste and increase network lifetime effectively by automatically adjusting the operation state according to the configuration state.

IV. Conclusions

This paper presents a dual control approach for energy efficient MNS interaction system. First, for the system configuration control, the proposed scheme indirectly influences the consensus operation of the multi-agent networking system by propagating the configuration state values to the system based on the proposed control laws of each agent.

Second, we propose a controller for agent's operational state scheduling according to the configuration propagation. Each agent determines the rate of propagation by reducing the number of message to be exchanged while keeping the energy consumption of agent members balanced.

Third, we propose an optimal timing control for sequent input. Using the operation state control model, the gateway agent determines the optimal timing to give next input based on the value of the operation state. From the simulation results, we observe that the proposed scheme is less susceptible to error and shows more robust performance than the consensus method in an error-prone environment.

An important area for further study includes the selection of the values of parameter $r_{\min}$ and ϵr, estimation of performance impact of applied parameters, and state modeling of moving MNS. This research could also investigate the effective collaboration among agents for appropriate decision making.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2017R1A2B1007779).

References

[1] J. C. Barca and Y. A. Sekercioglu, "Swarm robotics reviewed," *Robotica*, vol. 31, no. 3, pp. 345-359, 2013.

[2] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intell.*, vol. 7, no. 1, pp. 1-41, 2013.

[3] Y. Tan and Z. Y. Zheng, "Research advance in swarm robotics," *Defence Technol.*, vol. 9, no. 1, pp. 18-39, 2013.

[4] A. Liekna and J. Grundspenkis, "Towards practical application of swarm robotics: overview of swarm tasks," *Engineering for Rural Development*, pp. 271-277, 2014.

[5] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Trans. on Human-Machine Systems*, vol. 46, no. 1, pp. 9-26, 2016.

[6] Z. Kira and M. A. Potter, "Exerting human control over decentralized robot swarms," in *Proc. Int. Conf. on Autonomous Robots and Agents*, 2009, pp. 566-571.

[7] M. Lewis, "Human interaction with multiple remote robots," *Rev. Human Factors Ergonomics*, vol. 9, no. 1, pp. 131-174, 2013.

[8] S. Kerman, D. Brown, and M. Goodrich, "Supporting human interaction with robust robot swarms," in *Proc. Int. Conf. Resilient Control Systems*, 2012, pp. 197-202.

[9] J. P. de la Croix and M. Egerstedt, "A control Lyapunov function approach to human-swarm interactions," in *Proc. Amer. Control Conf.*, 2015, pp. 4368-4373.

[10] M. A. Goodrich, B. Pendleton, and S. Kerman, "What types of interactions do bio-inspired robot swarms and flocks afford a human?" in *Proc. Robotics, Science, and Systems*, 2012, pp. 105-112.

[11] J. Gancet, E. Motard, A. Naghsh, C. Roast, M. M. Arancon, and L. Marques, "User interfaces for human robot interactions with a swarm of robots in support to firefighters," in *Proc. Int. Conf. Robotics and Automation*, 2010.

[12] S. Bashyal and G. K. Venayagamoorthy, "Human swarm interaction for radiation source search and localization," in *Proc. IEEE Swarm Intelligence Symposium*, 2008, pp. 1-8.

[13] B. Tabibian, M. Lewis, C. Lebiere, N. Chakraborty, K. Sycara, S. Bennati, and M. Oishi, "Towards a cognitively-based analytic model of human control of swarms," in *Proc. AAAI Spring Symp. Series*, 2014, pp. 68-73.

[14] S. Nagavalli, L. Luo, N. Chakraborty, K. Sycara, "Neglect benevolence in human control of robotic swarm," in *Proc. Int. Conf. Robotics and Automation*, 2014.

[15] K. Sycara, C. Lebiere, Y. Pei, D. Morrison, Y. Tang, and M. Lewis, "Abstraction of analytical models from cognitive models of human control of robotic swarms," in *Proc. Int. Conf. Cognitive Model*, 2015, pp. 13-18.

[16] P. Walker, S. A. Amraii, M. Lewis, N. Chakraborty, and K. Sycara "Human control of leader-based swarms," in *Proc. IEEE Int. Conf. Syst. Man, Cybern.*, 2013, pp. 2712-2717.

[17] P. Walker. S. A. Amraii, N. Chakraborty, and K. Sycara, "Human control of robot swarms with dynamic leaders," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2014, pp. 1108-1113.

[18] M. Goodrich, B. Pendleton, P. Sujit, and J. Pinto, "Toward human interaction with bio-inspired robot teams," in *Proc. IEEE Int. Conf. Syst. Man, Cybern.*, 2011, pp. 2859-2864.

[19] B. Pendleton and M. Goodrich, "Scalable human interaction with robotic swarms," *American Institute of Aeronautics and Astronautics*, 2015.

[20] J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, D. Blau, and B. Schmidt, "Speaking swarmish: Human-Robot interface design for large swarms of autonomous mobile robots," in *Proc. AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, 2006, pp. 72-75.

[21] M. Li, K, Lu, H. Zhu, M. Chen, S. Mao, and B. Prabhakaran "Robot swarm communication networks: architectures, protocols, and applications," in *Proc. Int. Conf. Communications and Networking in China*, 2008.

[22] C. C. Moallemi and B. V. Roy, "Consensus propagation," *IEEE Trans. on Information Theory*, vol. 52, no. 11, pp. 4753-4766, 2006.

Heejung Byun received the BS degree from Soongsil University, Korea, in 1999, the MS degree from the Korea Advanced Institute of Science and Technology (KAIST) in 2001, and the PhD degree from KAIST in 2005. She was a senior researcher with Samsung Electronics from 2007 to 2010. She is currently an associate professor in the Department of Information and Telecommunications Engineering, Suwon University, Gyeonggi-do, Korea. Her research interests include swarm network modeling, multi-agent network controller design, and performance analysis.

Junglok Yu received the PhD degree from the Korea Advanced Institute of Science and Technology, Daejeon, in 2007. He was a senior engineer with Samsung Electronics from 2007 to 2010. He is currently a senior researcher with the Department of Super Computing, Korea Institute of Science and Technology Information. His research interests include network control, cluster computing, imbedded system, and parallel processing.