A Common Coordinates/Heading Direction Generation Method for Wirelessly Networked Robots with only Ranging Capability

Shinsuke Hara
Graduate School of Engineering
Osaka City University
Sumiyoshi-ku, Osaka 558-8585, Japan
Email: hara@info.eng.osaka-cu.ac.jp

Tatsuya Ishimoto
Graduate School of Engineering
Osaka City University
Sumiyoshi-ku, Osaka 558-8585, Japan
Email: tatsuya@comm.info.eng.osaka-cu.ac.jp

***Abstract*—Wirelessly networked robots having unknown locations and different individual heading directions cannot accomplish a unified task as a group. Therefore, it is necessary to generate a set of common coordinates among them and to notify each robot of its heading direction in the generated common coordinates. However, when the robots are not equipped with sensors to identify their locations or bearings, we can use only a function in wireless communication protocol to network them as a tool to generate a set of common coordinates among them.**

The present paper proposes a set of common coordinates and a heading direction generation method for wirelessly networked robots with only ranging capability, which is supported in wireless communication protocol. We explain the principle of the proposed method and show some computer simulation results on the location and direction estimation errors. Finally, we outline our prototype wirelessly networked robots system to experimentally evaluate the proposed method.

I. Introduction

In a group of wirelessly networked robots [1], [2], [3], unless each robot knows its own location and heading direction, the robots cannot accomplish a unified task as a group. Therefore, it is necessary to generate a set of common coordinates among the robots and to notify each robot of its heading direction in the generated coordinates.

However, in wirelessly networked *"micro"* robots such as smart pills and drug delivery systems, since the size, energy, and memory, of each robot are limited, such robots cannot be equipped with high-intelligent sensors, such as GPS (global positioning system) or geomagnetic force sensor, which could be used to determine location and bearing. On the other hand, wirelessly networked *"rescue"* robots often operate in disaster conditions, such as underground or inside buildings, in which GPS and geomagnetic force sensor do no work. Therefore, for applications of such wirelessly networked robots, it is necessary to generate a set of common coordinates among the robots and to notify each robot of its heading direction without the use of special sensors.

Since robots are generally networked by wireless communication, only functions supported by wireless communication protocols can be used to generate a set of common coordinates and a heading direction. In current wireless communication

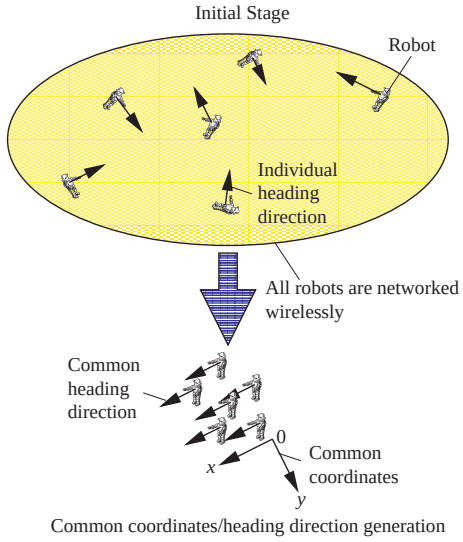

Fig. 1. Problem of common coordinates/heading direction generation.

standards, a ranging capability, which measures the distance between a transmitter and a receiver, is supported in most physical layer (PHY) and medium access control (MAC) protocols. For instance, the IEEE 802.11 and 802.15.4 standards [4] support ranging with received signal strength indication (RSSI), whereas the IEEE 802.15.4a standard with time of arrival (TOA) makes effective use of the characteristics of impulse radio (IR). Therefore, it is natural to assume that the distance between any pair of robots in the communication range can be easily measured through wireless communication without the need for any special devices for measuring distance.

In the present paper, we propose a set of common coordinates and a heading direction generation method for wirelessly networked robots with only ranging capability. The proposed method employs RSSI-based ranging because we are currently developing a prototype wirelessly networked robots system by means of the RSSI-measurable IEEE 802.15.4 standard.

ROBOCOMM 2007, 14th – 16th Oct 2007, Athens, Greece.

ISBN 978-963-9799-08-0
DOI 10.4108/ICST.ROBOCOMM2007.2155

The present paper is organized as follows. Section II states the problem of common coordinates and heading direction generation and some assumptions to solve the problem. Section III presents the details of the proposed method, which is composed of three major components. Section IV shows some computer simulation results on the performance of the proposed method in terms of the location and direction estimation errors. Section V outlines the prototype wirelessly networked robots system that we are currently developing in order to experimentally demonstrate the process of the proposed common coordinates and heading direction generation method. Finally, Section VI concludes the paper.

II. Problem Statement of Common Coordinates/Heading Direction Generation

Figure 1 shows the problem of the common coordinates and heading direction generation discussed in the present paper.

In the initial stage, it is assumed that all robots have been wirelessly networked, but that each robot has its own (different) individual heading direction (for instance, north) and no knowledge of its coordinates. The problem is generating a set of common coordinates among all of the robots and notifying each robot of its heading direction in the generated coordinates using only a function in wireless communication. Here, for the sake of simplicity, we assume that all of the robots are on the same plane, namely, we discuss a two-dimensional common coordinate generation: (x, y). Note that the proposed method can be easily extended to the three-dimensional case.

A signal emitted from a robot experiences multipath reflections by other robots and surrounding obstacles, and furthermore, other robots in motion introduce a time-varying aspect to the signal received by each robot [5]. Therefore, the power (RSSI) of a received signal fluctuates in time, and so we herein assume the following two-layered model on the distribution of the received power: [6],[7]

$$\overline{P} = \alpha d^{-\beta} \quad (1)$$

$$p(P|d) = \frac{1}{\overline{P}} \exp(-\frac{P}{\overline{P}}) \quad (2)$$

where P, $\overline{P}$, and d denote the received power, the average received power and the distance between a transmitter robot and a receiver robot, respectively, and $p(P|d)$ denotes the conditional probability density function (pdf) of P when d is given. In (1), α and β are constants that are uniquely determined by the medium of the channel and the carrier frequency and bandwidth of the signal.

III. Proposed Common Coordinates/Heading Direction Generation Method

The proposed common coordinates/heading direction generation method is composed of three elements, such as pivot robots selection, location estimation, and heading direction estimation.

A. Pivot Robots Selection

On a plane, if we know the locations of three different robots, we can uniquely determine the location of any robot according to these three locations. The proposed pivot robots selection chooses three robots with different locations as "pivot robots." Here, we assume that there are M robots communicating with each other and the robots are autonomously numbered at random as 1 to M (ID number).

In the first step, each robot broadcasts N_o "hello packets" containing its ID number to all other robots. Defining P_{ijs} as the RSSI of the s-th packet ($s = 1, \cdots, N_o$) transmitted from the i-th robot and received at the j-th robot, with (1), the j-th robot can calculate the average RSSI and then the distance between them as ($j = 1, \cdots, N_o, i \neq j$)

$$\overline{P}_{ij} = \frac{1}{N_o} \sum_{s=1}^{N_o} P_{ijs} \quad (3)$$

$$d_{ij} = \left(\overline{P}_{ij}/\alpha\right)^{-1/\beta} \quad (4)$$

and broadcasts d_{ij} to all other robots. In this way, all of the robots can share the information on the distances between all pairs of robots d_{ij} ($i, j = 1, \cdots, M, i \neq j$).

In the second step, each robot autonomously select a pair of robots separated by the largest distance:

select robots *i* and *j*

$$i, j = \arg\max_{i,j} \{d_{ij} | i, j = 1, \cdots, M, i \neq j\} \quad (5)$$

and each arbitrarily designates one of the two robots as the "master pivot robot", with the location vector of $\mathbf{Z}_1 = [X_1, Y_1] = [0, 0]$, whereas the other robot is designated as the "slave pivot robot", with the location of $\mathbf{Z}_2 = [X_2, Y_2] = [d_{ij}(> 0), 0]$.

In the third step, as "another slave pivot robot," each robot autonomously selects a robot located farthest from the pivot robots selected in the second step:

select robot *k*

$$k = \arg\max_{k} \{d_{ik} + d_{kj} | k = 1, \cdots, M, k \neq i, j\} \quad (6)$$

with the location vector of $\mathbf{Z}_3 = [X_3, Y_3 > 0]$ satisfying

$$X_3^2 + Y_3^2 = d_{ik}^2 \quad (7)$$

$$(d_{ij} - X_3)^2 + Y_3^2 = d_{kj}^2. \quad (8)$$

In this way, each robot autonomously selects three pivot robots that are located far from each other. Finally, each robot then re-numbers the master pivot robot as 1. The slave pivot robots are re-numbered as 2 and 3, and the other non-pivot robots are re-numbered as 4 to M. Figure 2 shows an example of the (far) pivot robots selection with $M = 4$ after the re-numbering is finished.

Note that each robot can randomly select three robots as one alternative, and they can re-numbered as select three robots located in close proximity to each other as another alternative. We will compare the location estimation performance among the far, random, and near pivot robots selections in Section IV.

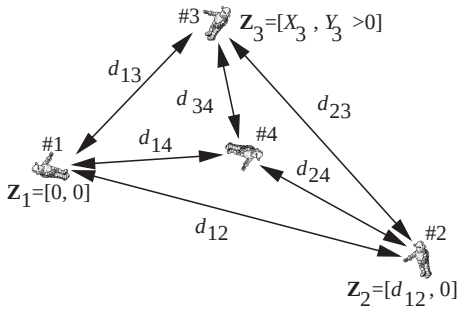

Fig. 2. Pivot robots selection with $M = 4$.

B. Iterative Maximum Likelihood Location Estimation

Once the three pivot robots have been selected, they begin to broadcast their locations to all other robots. Here, we apply the index l to the pivot robots ($l = 1, 2, 3$), whereas the index m is applied to the non-pivot robots ($m = 4, \cdots, M$).

In the first step, each pivot robot broadcasts N packets containing its ID number and location to all other robots. Defining the location vector of the m-th non-pivot robot as $\mathbf{z}_m = [x_m, y_m]$, the distance between the l-th pivot robot and the m-th non-pivot robot is written as

$$\begin{aligned} d_{lm} &= |\mathbf{Z}_l - \mathbf{z}_m| \\ &= \sqrt{(X_l - x_m)^2 + (Y_l - y_m)^2}. \end{aligned} \tag{9}$$

Then, define the RSSI vector as

$$\mathbf{P}_{mn} = [P_{1mn}, P_{2mn}, P_{3mn}] \tag{10}$$

where P_{lmn} denotes the RSSI of the n-th packet ($n = 1, \cdots, N$) transmitted from the l-th pivot robot and received at the m-th non-pivot robot. Since the unknown location of the non-pivot robot $\mathbf{z}_m$ is estimated with the measured RSSIs, the log-likelihood function on $\mathbf{z}_m$ is written with the conditional *pdf* of $\mathbf{P}_{mn}$ ($n = 1, \cdots, N$) when $\mathbf{z}_m$ is given as

$$L(\mathbf{z}_m) = \log p(\mathbf{P}_{m1}, \mathbf{P}_{m2}, \cdots, \mathbf{P}_{mN} | \mathbf{z}_m). \tag{11}$$

Assuming that P_{lmn} is the statistically uncorrelated value of $P_{lmn'}$ ($n \neq n'$) (*temporal whiteness*) and $P_{l'mn}$ ($l \neq l'$) (*geographical whiteness*), replacing d by d_{lm} and P by P_{lmn}, respectively, in (1) and (2), (11) yields

$$\begin{aligned} L(\mathbf{z}_m) &= \log \left[\prod_{n=1}^{N} \prod_{l=1}^{3} \left\{ \frac{1}{\alpha {d_{lm}}^{-\beta}} \exp\left(\frac{-P_{lmn}}{\alpha {d_{lm}}^{-\beta}} \right) \right\} \right] \\ &= N \sum_{l=1}^{3} \left\{ \log\left(\frac{1}{\alpha {d_{lm}}^{-\beta}} \right) - \frac{\sum_{n=1}^{N} P_{lmn}/N}{\alpha {d_{lm}}^{-\beta}} \right\}. \end{aligned} \tag{12}$$

The ML estimation gives $\hat{\mathbf{z}}_{m0} = [\hat{x}_{m0}, \hat{y}_{m0}]$, which maximizes (12) [8]:

$$\left. \frac{\partial L(\mathbf{z}_m)}{\partial \mathbf{z}_m} \right|_{\hat{\mathbf{z}}_{m0} = [\hat{x}_{m0}, \hat{y}_{m0}]} = 0 \quad (m = 4, \cdots, M). \tag{13}$$

Since the locations of the non-pivot robots have been estimated in the first step, the robots also begin to broadcast their ID numbers and estimated locations to all other robots. In the second step, each non-pivot robot estimates its location each time it receives broadcast packets from all other robots, and then broadcasts back a packet containing its newly estimated location with its ID number to all other robots. On the other hand, each pivot robot improves its location accuracy every time it receives broadcast packets from other pivot robots.

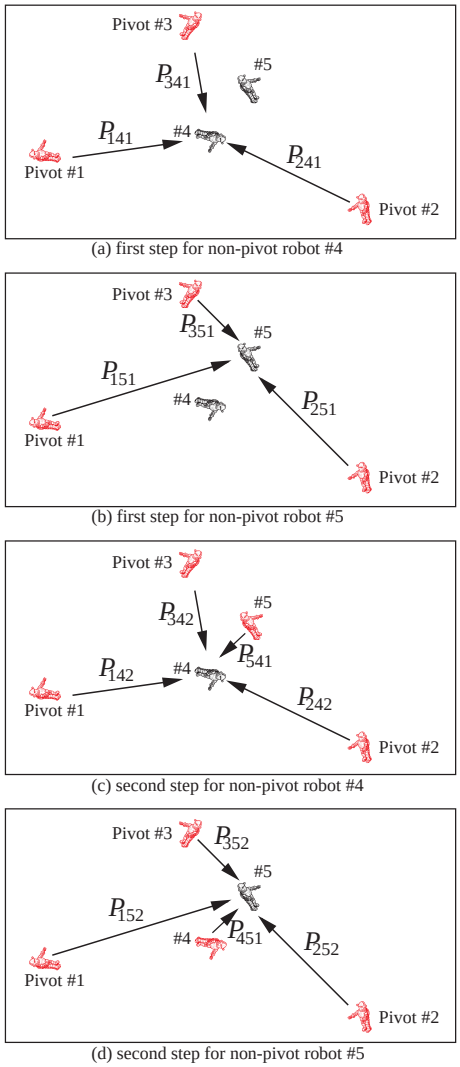

Fig. 3. Iterative maximum likelihood location estimation with $M = 5$, $N = 1$, and $Q = 1$.

Define the estimated location vectors of the l-th pivot robot and the m-th non-pivot robot with the q-th broadcast packet as $\mathbf{Z}_{lq}$ and $\mathbf{z}_{mq}$ ($q = 1, \cdots, Q$), respectively. $\mathbf{Z}_{lq}$ can be estimated by the same procedure in the pivot robots selection replacing N_o by q in (3). Here, the distance between the l-th pivot robot and the m-th non-pivot robot with the q-th broadcast packet is written as

$$d_{lmq} = |\mathbf{Z}_{lq} - \mathbf{z}_{mq}|. \tag{14}$$

On the other hand, when the m-th non-pivot robot receives the q-th broadcast packet from the m'-th non-pivot robot with the RSSI of $P'_{m'mq}$, which contains the estimated location

vector of the m'-th non-pivot robot $\hat{\mathbf{z}}_{m'(q-1)}$, it can use the m'-th non-pivot robot as a *pivot robot* with the location vector of $\hat{\mathbf{z}}_{m'(q-1)}$ ($m' = 4, \cdots, M, m' \neq m$). Namely, the distance between the m-th non-pivot robot and the m'-th non-pivot robot with the *temporarily known* location vector of $\hat{\mathbf{z}}_{m'(q-1)}$ is

$$\begin{aligned} d_{m'mq} &= |\hat{\mathbf{z}}_{m'(q-1)} - \mathbf{z}_{mq}| \\ &= \sqrt{(\hat{x}_{m'(q-1)} - x_{mq})^2 + (\hat{y}_{m'(q-1)} - y_{mq})^2} \end{aligned} \tag{15}$$

so the log-likelihood function on $\mathbf{z}_{mq}$ is written as

$$\begin{aligned} L(\mathbf{z}_{mq}) &= \log \left[\prod_{q'=1}^{q} \prod_{l=1}^{3} \left\{ \frac{1}{\alpha {d_{lmq'}}^{-\beta}} \exp\left(\frac{-P_{lmq'+N}}{\alpha {d_{lmq'}}^{-\beta}} \right) \right\} \right. \\ &\quad \left. \cdot \prod_{\substack{m'=4 \\ m' \neq m}}^{M} \left\{ \frac{1}{\alpha {d_{m'mq'}}^{-\beta}} \exp\left(\frac{-P'_{m'mq'}}{\alpha {d_{m'mq'}}^{-\beta}} \right) \right\} \right] \\ &= \sum_{q'=1}^{q} \left[\sum_{l=1}^{3} \left\{ \log\left(\frac{1}{\alpha {d_{lmq'}}^{-\beta}} \right) - \frac{P_{lmq'+N}}{\alpha {d_{lmq'}}^{-\beta}} \right\} \right. \\ &\quad \left. + \sum_{\substack{m'=4 \\ m' \neq m}}^{M} \left\{ \log\left(\frac{1}{\alpha {d_{m'mq'}}^{-\beta}} \right) - \frac{P_{m'mq'}}{\alpha {d_{m'mq'}}^{-\beta}} \right\} \right]. \end{aligned} \tag{16}$$

The ML estimation yields $\hat{\mathbf{z}}_{mq} = [\hat{x}_{mq}, \hat{y}_{mq}]$, which maximizes (12):

$$\left. \frac{\partial L(\mathbf{z}_{mq})}{\partial \mathbf{z}_{mq}} \right|_{\hat{\mathbf{z}}_{mq} = [\hat{x}_{mq}, \hat{y}_{mq}]} = 0 \quad (m = 4, \cdots, M). \tag{17}$$

In this way, each non-pivot robot and each slave pivot robot can iteratively estimate their current locations with the previous locations of the other pivot robots and non-pivot robots up to $q = Q$. Figure 3 shows an example of iterative maximum likelihood location estimation with $M = 5$, $N = 1$, and $Q = 1$.

Note that all robots autonomously generate a set of common coordinates, so the coordinates have ambiguities such as translation, rotation, and negation with respect to the coordinates of an observer (operator) of the wirelessly networked robots. However, this is not critical because we can determine the relationship between the two coordinates.

C. Heading Direction Estimation

After the pivot robots selection and iterative maximum likelihood location estimation, all robots have generated a set of common coordinates, and each robot knows its location in the generated coordinates.

In the zero-th step, the robots autonomously divide the set of all robots into U subsets with equal numbers of robots. Next, in the u-th location/direction estimation step ($u = 1, \cdots, U$) with Q broadcast packets, each robot in the u-th subset moves through a distance of B, according to its own individual heading direction and stops, and its location is then estimated, starting with the robots in the other subsets ($u' = 1, \cdots, U, u' \neq u$) as pivot robots in the same manner as the iterative maximum likelihood location estimation. This process is repeated until $u = U$, and then the m-th robot ($m = 1, \cdots, M$) can determine its location before and after the movement, namely, $\mathbf{z}_m^b = [x_m^b, y_m^b]$ and $\mathbf{z}_m^a = [x_m^a, y_m^a]$. Finally, with the direction of the movement vector, the robot can estimate the angle between its heading direction and the x-axis in the generated coordinates, that is,

$$\hat{\theta}_m = \arg\,(\mathbf{z}_m^a - \mathbf{z}_m^b). \tag{18}$$

Figure 4 shows an example of heading direction estimation with $M = 6$ and $U = 2$. Note that if a moving robot collides with another stationary robot, the moving robot returns to its original location, changes its heading direction by $+\gamma$ degrees and moves again. This process is repeated until the robot has successfully finished moving through distance of B without collision.

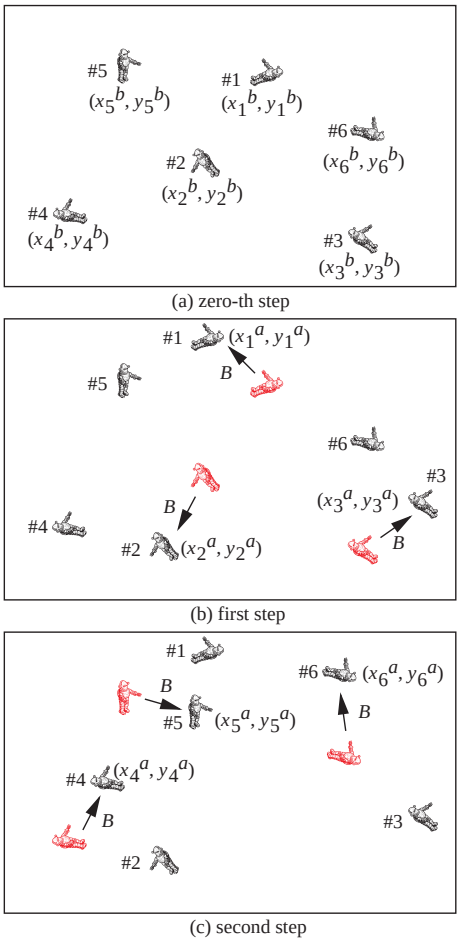

Fig. 4. Heading direction estimation with $M = 6$ and $U = 2$.

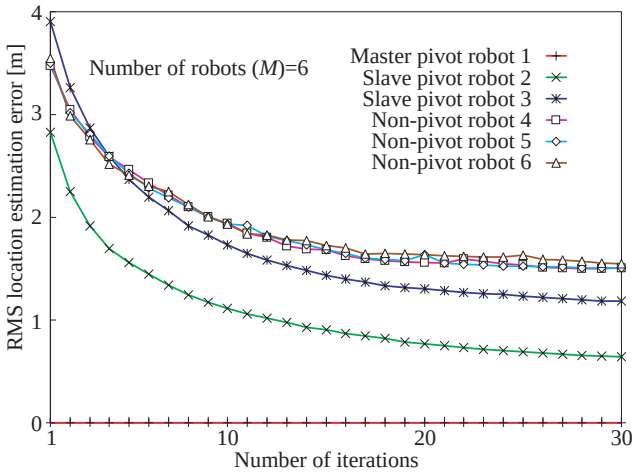

Fig. 5. RMS location estimation error for individual robots.

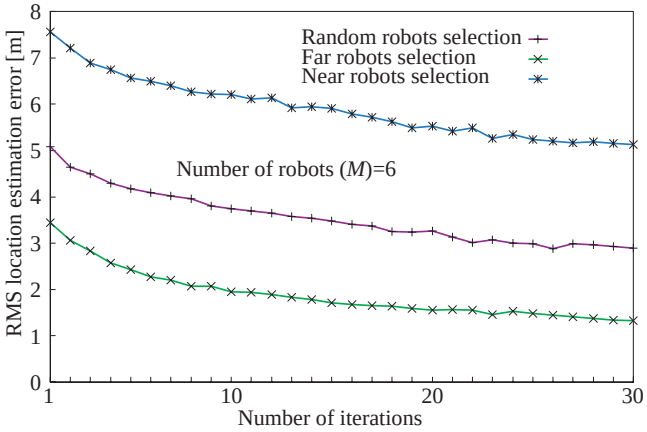

Fig. 6. Effect of robots selection in location estimation.

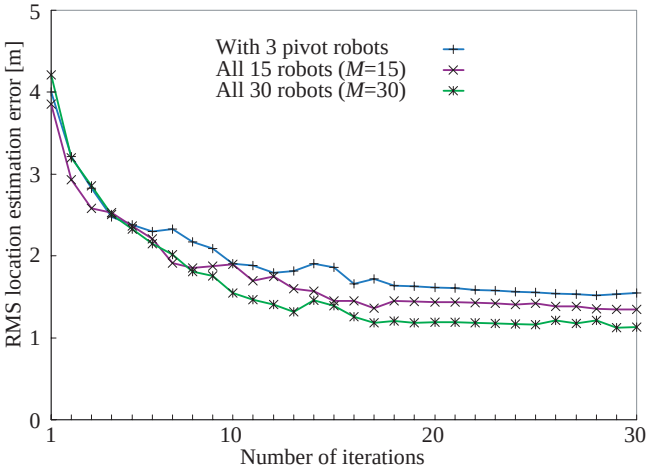

Fig. 7. Effect of the number of robots in iterative location estimation.

IV. Computer Simulation Results and Discussions

As shown in Section V, we are currently developing a prototype wirelessly networked robots system to demonstrate the proposed common coordinates and heading direction generation method experimentally, where the PHY/MAC protocol is based on the IEEE 802.15.4 standard. Therefore, we determined the values of the two parameters in (1) as $\alpha = 2.36 \times 10^{-6}$ and $\beta = 2.37$ by a channel measurement experiment using a set of IEEE 802.15.4-based transceivers in a room.

In a computer simulation, we assume a field of 10 m$\times$10 m and randomly select the locations of robots in the field. To speed up the process of generating the common coordinates and heading direction, we set N_o=1 and N=1. Furthermore, we refer to the number of broadcast packets (Q) as the "number of iterations."

Figure 5 shows the root mean square (RMS) location estimation error with respect to the number of iterations for the case of six robots. For all of the robots, as the number of iterations increases, the location estimation error gradually decreases because more packets (information) can be used for location estimation. For the pivot robots, the master robot is located at the origin, so its location estimation error is always zero, whereas the location estimation error of slave robot 3 is affected by that of slave robot 2, so the location estimation error of slave robot 3 is worse than that of slave robot 2. On the other hand, for the non-pivot robots, location estimation errors are affected by the worst location estimation error among the location estimation errors of the pivot robots. Thus, the location estimation errors of the non-pivot robots are worse than the location estimation error of slave robot 3. However, there is no significant difference in the location estimation error among the non-pivot robots. In the following, the location estimation error is averaged over all types of robots such as master pivot, slave pivot, and non-pivot robots.

Figure 6 shows the effect of pivot robots selection for the case of six robots. The distances between pivot robots and non-pivot robots should be shorter because they have larger receiving powers and, consequently, smaller location estimation errors. In this sense, the case in which non-pivot robots are located in the area of a triangle formed by pivot robots as its three vertexes provides better location estimation performance. When three robots at distant locations from one another are selected as pivot robots, the triangle formed by the three pivot robots tends to include more non-pivot robots, so a smaller location estimation error is obtained, whereas when three robots in close proximity to one another are selected, a larger location estimation error is obtained. The performance provided by random robots selection lies between those of the far and near robots selections.

Figure 7 shows the effect of the number of robots on the iterative location estimation performance. The iterative location estimation dealing with non-pivot robots as *temporal* pivot robots becomes more effective as the number of robots increases because packets from other non-pivot robots, even if they contain some ambiguities with respect to their locations,

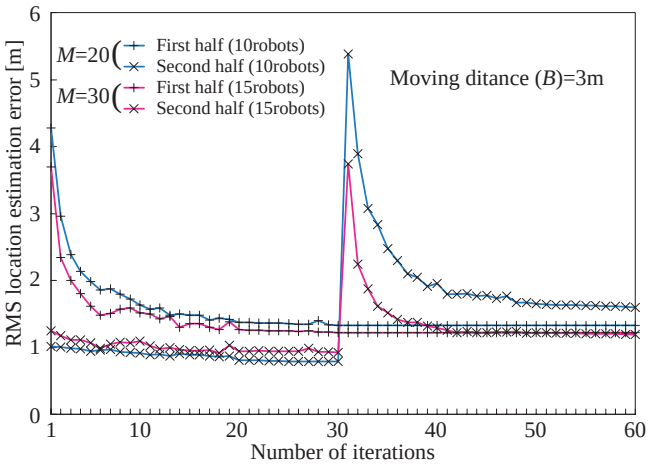

Fig. 8. RMS location estimation error in the heading direction estimation.

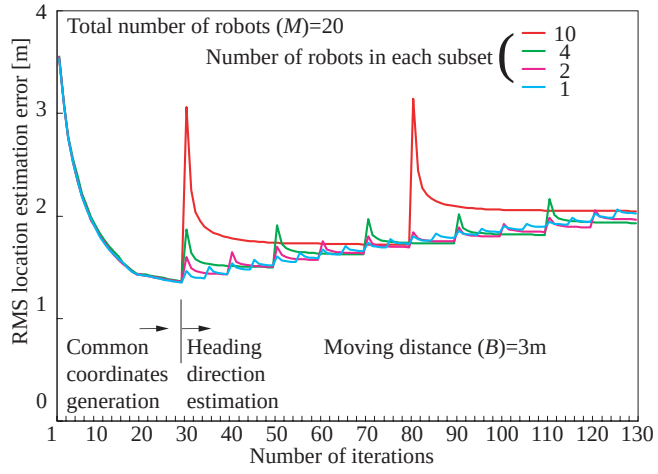

Fig. 9. Effect of the number of location/direction estimation steps on the location estimation error in the heading direction estimation.

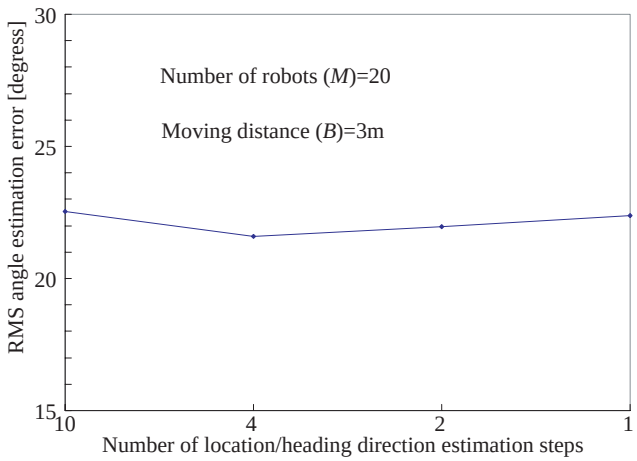

Fig. 10. RMS location estimation error with respect to the number of location/direction estimation steps.

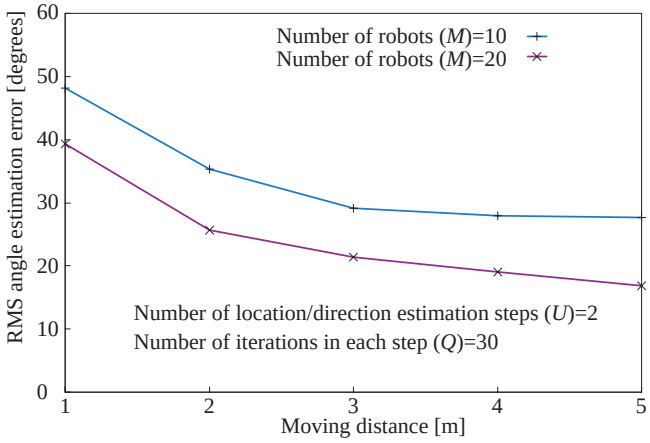

Fig. 11. Effect of moving distance on heading direction estimation.

are helpful for improving the location estimation performance.

Figure 8 shows the RMS location estimation error in the heading direction estimation for the cases of (M) = 20 and 30 robots with (U) = 2 subsets and a moving distance of (B) = 3 m. Here, after the robots in the first location/heading direction estimation step move through $B = 3$, their locations are estimated with 30 packets, and then the robots in the second step move. At the beginning of the iterations in each location/direction estimation step, the location estimation error is large, but it is improved as the number of iterations increases. In addition, a larger total number of robots provides better location estimation performance. The robots in the first location/heading direction estimation step, with their poorer location estimation error, estimate the locations of the robots in the second step, so that the residual location estimation error in the second step is larger than that in the first step.

Figure 9 shows the effect of the number of location/heading direction estimation steps on the RMS location estimation error in the heading direction estimation for the case of (M) = 20 robots. The location estimation with the smaller number of robots in each subset (larger number of subsets) shows a quicker convergence. Therefore, in this case, the total number of iterations decreases as the number of robots in each subset decreases. Note that the location estimation error in a location/direction estimation step is affected by the residual location estimation errors in all of the location/direction estimation steps before the location/direction estimation step of interest. Therefore, the residual location estimation error is a monotonically increasing function on the index number of location/direction estimation steps. In this sense, fewer location/direction estimation steps provides better location estimation performance averaged over all robots. However, when the number of location/direction estimation steps is smaller, the residual location estimation error in each step is larger, because the number of robots acting as pivots decreases. Therefore, for a given total number of robots, there

is an optimal number of location/direction estimation steps that minimizes the location estimation error and, consequently, the heading direction estimation error averaged over all robots. Figure 10 shows the RMS angle estimation error with respect to the number of location/direction estimation steps. This figure clearly shows that, for 20 robots, the angle estimation error is minimized for the case of four steps.

Figure 11 shows the RMS angle estimation error with respect to the moving distance. In addition, Fig.12 shows the *pdf* of the angle estimation error. As the moving distance becomes larger, the RMS angle estimation error, namely, the standard deviation of the angle estimation error, decreases. However, "motion and its control" consumes much more energy than "wireless communications." Therefore, consideration of the energy constraint in the problem of common coordinates and heading direction generation will be investigated in future studies.

Finally, Fig.13 shows an example of the obtained heading directions, where the arrows with solid and dashed lines show the real and estimated heading directions, respectively. With the RSSI-based location estimation, the performance of which is poorer than those of other methods, such as the TOA method, a set of common coordinates can be generated among all robots and each heading direction can be roughly estimated. Because of the poor performance of the RSSI-based location estimation, a fine direction estimation within a few degrees cannot be achieved. Therefore, in the next step to control all of the robots as a group in order to carry out a task, a method to generate a perfectly common heading direction is required.

V. Development of the Prototype Wirelessly Networked Robots System

To evaluate the performance of the proposed common coordinates/heading direction estimation method experimentally, we are currently developing a prototype wirelessly networked robots system. Figure 14 shows a photograph of a robot based on a tank kit manufactured by TAMIYA. Figure 15 shows the inside of the robot, where the control element is composed of an I/O board, a CPU board, and a PHY/MAC board. The I/O board controls the DC motors of the robot for movement, and optionally gathers information, such as temperature, from sensors. The CPU board, which is a Linux PC, is equipped with a high-speed processor (416 MHz) enabling both real-time wireless communications and motion control and sufficient memory (RAM: 64 MB, ROM: 128 MB). The proposed common coordinates and heading direction generation algorithm can be programmed using C++ via the PC. The PHY/MAC board is a CF or PCI card with the IEEE 802.11 standard or the IEEE 802.15.4 standard, which supports the RSSI measurement function as a link quality indicator. We are currently considering the use of the IEEE 802.15.4 (Zigbee) card.

Figure 16 shows the I/O board in detail. The PIC interprets the commands from the CPU and drives the motors accordingly. In addition, the I/O board is equipped with an RS-232C port and a USB port, so that several sensors and

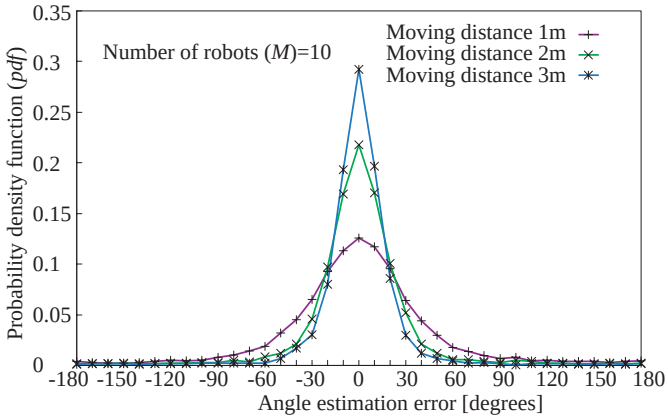

Fig. 12. *pdf* of the angle estimation error.

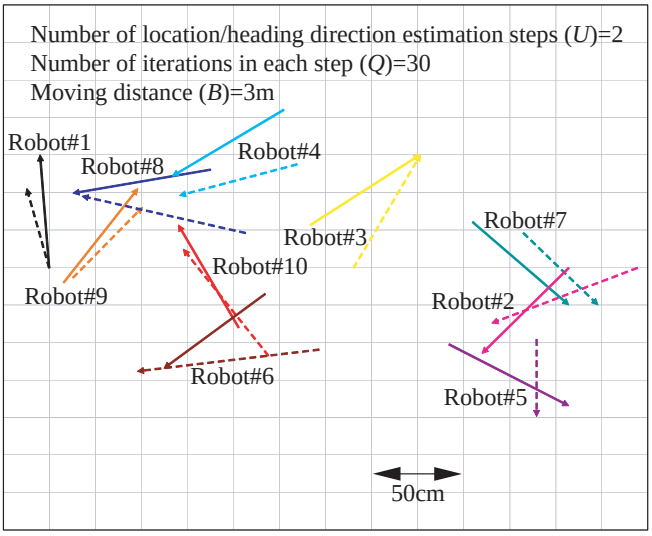

Fig. 13. Example of estimated heading directions for 10 robots.

input/output devices can be connected to the board. We are currently debugging the algorithm of the proposed common coordinates/heading direction generation after assembling six robots.

VI. Conclusions

The present paper has proposed a set of common coordinates and a heading direction generation method with only ranging capability for wirelessly networked robots. We have assumed an RSSI measurement as a ranging method, which is easily realized in wireless communication (however, it is not the only ranging method available to us). Due to the large variation of the received signal power resulting from multipath fading as well as the near/far effect, the proposed method can achieve an angle estimation error from the x-axis of approximately 18 degrees for the case of 20 robots, 30 iterations, two location/direction estimation steps, and a moving distance of 5 m. An additional heading direction

generation method is required that can achieve fine angle estimation after rough angle estimation is achieved using the proposed method. We intend to investigate such a method in the future.

The proposed method is based on the maximum likelihood estimation with a nonlinear function shown in (16), so that the computational complexity is high. If the conditional *pdf* of the distance can be approximated with a Gaussian function, we can use a distributed belief propagation method [9]. Furthermore, even in general form, we can use a distributed particle filter method [10].

To demonstrate a proposed scheme experimentally, we have been developing a prototype wirelessly networked robots system. We are currently working to validate the proposed common coordinates and heading direction generation using the proto type system.

ACKNOWLEDGEMENT

This study was supported in part by a Grant–in–Aid for Scientific Research (No. 17656126) from the Ministry of Education, Science, Sport and Culture of Japan, and by the Research Organization on “Micro-Robotics” of Osaka City University, Japan.

REFERENCES

[1] J.P.Hubaux, T.G.J.Y.Le Boudec, and M.Vetterli “Toward Self-Organized Mobile Ad Hoc Networks: The Terminodes Project,” *IEEE Commun. Mag.*, pp.118-124, Jan. 2001.

[2] W.Ye, R.T.Vaughan, G.S.Sukhatme, J.Heidemann, D.Estrin, and M.J.Mataric, “Evaluating Control Strategies for Wireless-Networked Robots Using an Integrated Robot and Network Simulation,” *Proc. International Conference on Robotics & Automation*, pp.2941-2947, May 2001.

[3] Z.Butler and D.Rus, “Event-Based Motion Control for Mobile-Sensor Networks,” *IEEE Pervasive Computing*, pp.34-42, Oct.-Dec. 2003.

[4] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Layer Wireless Personal Area Networks (WPANs), IEEE Std.802.15.4, 2003.

[5] W.C.Y.Lee, *Mobile Communications Engineering*, McGraw-Hill, 1982.

[6] S.Hara, D.Zhao, K.Yanagihara, J.Taketsugu, K. Fukui, S. Fukunaga and K.Kitayama, “Propagation Characteristics of IEEE 802.15.4 Radio Signal and Their Application for Location Estimation,” *Proc. IEEE VTC 2005-Spring*, pp. 97-101, May 2005.

[7] R.Zemek, D.Zhao, M.Takashima, S.Hara, K.Yanagihara, K.Fukui, S.Fukunaga and K.Kitayama, “A Traffic Reducing Method for Multiple Targets Localisation in an IEEE 802.15.4 Based Sensor Network,” *Proc. IEEE VTC 2006-Fall*, in CD-ROM, Sept. 2006.

[8] N.Patwari, A.O.Hero III, M.Perkins, N.S.Correal, and R.J.O’Dea, “Relative Location Estimation in Wireless Sensor Networks,” *IEEE Trans. Signal Process.*, vol.51, no.8, pp.2137-2148, Aug. 2003.

[9] D.Koller, U.Lerner, and D.Angelov, “A General Algorithm for Approximate Inference and Its Application to Hybrid Bays Nets,” *Proc. UAI-99*, pp.324-333, 1999.

[10] A.T.Ihler, J.W.Fisher III, R.L.Moses, and A.S.Wilsky, “Nonparametric Belief Propagation for Self-Localization of Sensor Networks,” *IEEE J. Sel. Areas Commun.*, vol.23, no.4, pp.809-819, Apr. 2005.

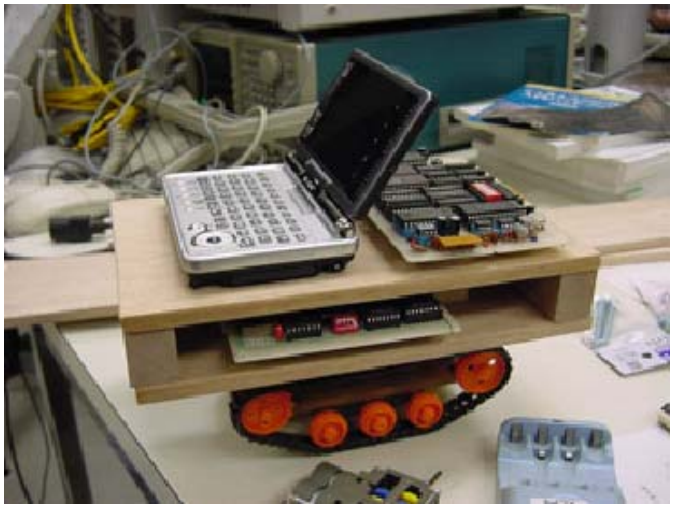

Fig. 14. Photograph of a robot.

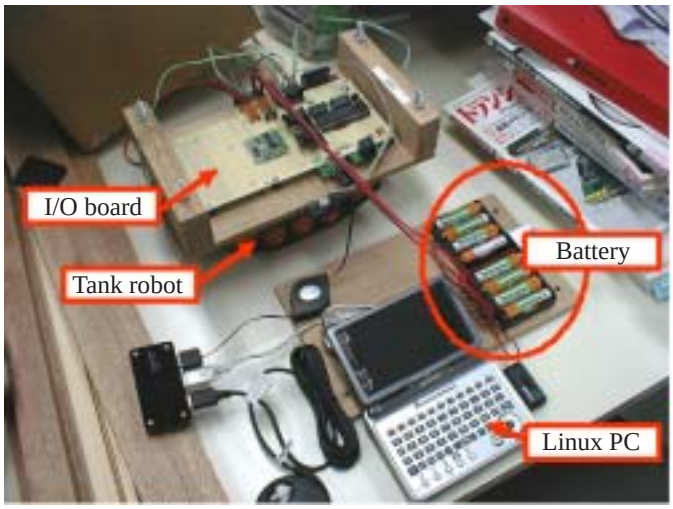

Fig. 15. Photograph of the inside of the robot.

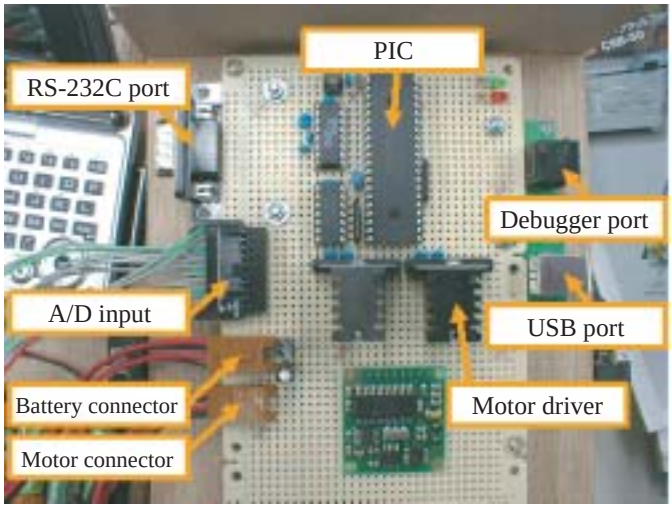

Fig. 16. Detail of the I/O board.