Robotica (2013) volume 31, pp. 1239–1249. © Cambridge University Press 2013
doi:10.1017/S0263574713000374

Faulty robot rescue by multi-robot cooperation

Gyuho Eoh*, Jeong S. Choi and Beom H. Lee

Department of Electrical Engineering, Seoul National University (ASRI), Seoul, Republic of Korea

(Accepted April 9, 2013. First published online: May 29, 2013)

SUMMARY

This paper presents a multi-agent behavior to cooperatively rescue a faulty robot using a sound signal. In a robot team, the faulty robot should be immediately recalled since it may seriously obstruct other robots, or collected matters in the faulty robot may be lost. For the rescue mission, we first developed a sound localization method, which estimates the sound source from a faulty robot by using multiple microphone sensors. Next, since a single robot cannot recall the faulty robot, the robots organized a heterogeneous rescue team by themselves with pusher, puller, and supervisor. This self-organized team succeeded in moving the faulty robot to a safe zone without help from any global positioning systems. Finally, our results demonstrate that a faulty robot among multi-agent robots can be immediately rescued with the cooperation of its neighboring robots and interactive communication between the faulty robot and the rescue robots. Experiments are presented to test the validity and practicality of the proposed approach.

KEYWORDS: Faulty robot; Rescue robot; Multi-agent cooperation; Sound localization.

1. Introduction

Saving faulty robots is essential in a multi-robot team because if a faulty robot is left alone for a long time, it may cause serious problems to a multi-agent system such as a drastic decline in performance or loss of collected matters in the faulty robots. It is also profitable for operating multi-agent system to reuse the collected faulty robots after repair. So far, little attention has been paid to the rescue of faulty robots[9] since most studies assume that robots in a multi-agent team are always operational.[11,23,24,26] However, robotic systems are sometimes out of order for various reasons such as collision[28,29] or robot deterioration, and thus the ability to rescue faulty robots is an important issue in the field of multi-agent applications. In a rescue mission, prompt action and strong cooperation among the operational robots are the key points from a practical point of view. Therefore, it is a challenging work to develop a multi-agent rescue method that can quickly recall faulty robots with only its neighboring robots.

Saving a faulty robot is technically analogous to object manipulation and object transportation in multi-agent systems since a faulty robot can be considered as a static object. Generally, a single robot alone has no sufficient power to handle a faulty robot, and thus strong cooperation among multiple robots is required for a successful rescue mission. Parker[19] classified cooperation into two types: *swarm* and *intentional*. Swarms are robotic systems consisting of large groups of relatively small and simple robots that work together and cooperate with each other to solve difficult tasks that a single robot cannot solve.[27] Intentional cooperation means that robots are controlled to explicitly coordinate actions with their teammates through the philosophy of a human designer. *Box-pushing* is a typical problem of intentional cooperation with the following specific behaviors: carry, push, and manipulate an object.[10,11] Variations of the box-pushing problem include several recent forms such as the *object closure*[20,21] condition and the *pusher–watcher*[13,22] problem. The object closure is a condition under which an object is trapped so that there is no feasible path for the object to move along to any position that is beyond a specified threshold distance. If we apply the closure condition to robot motion control, the object can be transported to a desired goal while maintaining a formation for the object closure. However, this approach has limitations such as the inability to handle heterogeneous robots and the lack of guarantee on the completeness of the object closure. In the pusher–watcher problem, pusher and watcher robots enable the cooperative manipulation of large objects by teams of autonomous mobile robots. This approach is tolerant to robot failures because a watcher plays the role of a supervisor.

Identifying the relative position of a faulty robot is a priority during a rescue mission. Recent studies mainly used vision and laser range finder (LRF) sensors to identify relative positions.[2,6] A robot can distinguish diverse objects using a vision sensor and can thus localize itself by detecting artificial or natural landmarks.[3,4] However, the performance of a vision sensor strongly depends on the intensity of illumination and the degree of visibility.[1] LRF sensor can measure data accurately and requires relatively little time to identify a position. However, it is limited because the recognition rate decreases particularly in the environments containing many angular objects.[5]

In contrast, sound is detectable regardless of the degree of visibility. Sound localization is a useful means of determining relative positions because sound can propagate everywhere in an existing medium. In some environments, light intensity and visibility conditions can vary greatly, which introduces more difficulties when seeking to detect robot positions with vision and LRF sensors. In such environments, a sound signal is the most informative source for rescue robots. In addition,

* Corresponding author. E-mail: geni0620@snu.ac.kr

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press

a sound sensor is inexpensive compared with other types of sensors.[30] Therefore, sound sensors are suitable for systems with multiple robots. For these reasons, this paper focuses on developing a method that utilizes sound signals for the identification of robot positions in environments in which the use of vision and LRF sensors is limited.

There are two representative approaches to identify the position of a sound source using physical differences in a sound signal: interaural time difference (ITD) and interaural level difference (ILD). The ITD is the time delay of sound at different observation points. We can know a sound source's position from the difference between right and left arrival times. This approach to position identification is well used in human localization[8,12] because it is more accurate and robust in voice frequency, which is closely related to time delay. The ILD is caused by sound attenuation as the distance traveled increases. According to the inverse-square law, decrease in sound level, which is propagated by the point source, is inversely proportional to the total distance traveled in the free field. Due to the difference between observation points, the sound levels of two microphones differ from each other. This difference is used as a clue to identify the sound source.[14,15] Although this cue has been known as the oldest theory of sound localization, the ILD localization has been rarely utilized in computer-based system. Recently, however, Birchfield and Gangishetty[16] have shown the possibility of using the ILD for sound localization. They used two microphones for detecting sound signal and succeeded in accurate localization using only the sound level difference. Other studies showed joint evaluation between ITD and ILD[17,18] for sound localization. Only one information is insufficient to localize a sound source, and thus they combined ITD and ILD for accuracy.

This paper presents a multi-robot cooperative rescue team that uses sound signals from a faulty robot. We focused on manipulating the faulty robot of the same size and weight by heterogeneous rescue robots for simplifying the problem formulation and for practical reasons that the same size robots are used for convenient programming, easy maintenance, and redundancy in a multi-robot system. To complete the rescue mission, we organized a systematic robot team and rescue processes. The proposed rescue team is composed of a *supervisor–puller–pusher* according to their functions. For practical applications to general environments, we used only sound data to determine where the faulty robot is and divided the rescue mission into the following three steps. The first step is position identification by generating sound vectors (SV) in the neighboring robots to identify the position of the faulty robot. The second step is position assignment of rescue robots for pushing or pulling of faulty robot. The third step is the rescue of the faulty robot through the corresponding sound and puller's direction vector.

The proposed method actively uses interactive real-time communication between the faulty robot and the rescue robots to share data. Unlike previous studies,[13,19] rescue robots can use partial functional abilities of the faulty robot to accomplish rescue mission. For example, the faulty robot can emit a sound signal using a speaker to induce rescue robots to move without the help from a global positioning system (GPS). Moreover, proximity sensing data

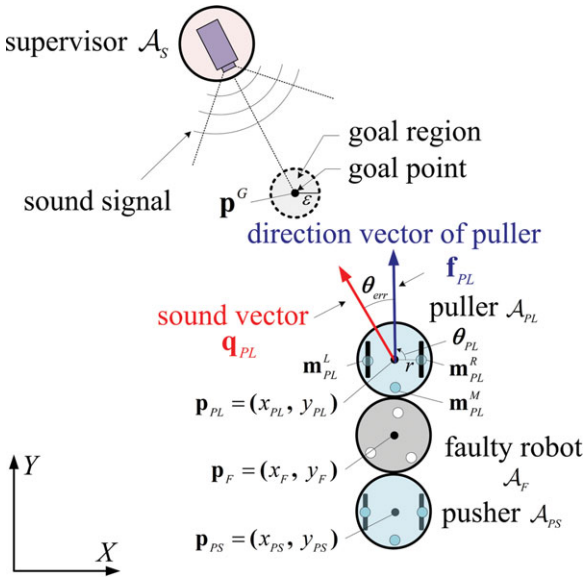


Fig. 1. (Colour online) Rescue mission of a faulty robot.

from the faulty robot allows the rescue robots to move to the assigned position around the faulty robot. The rescue robots communicate with the faulty robot and share the sensing information from the faulty robot. This is more advantageous than methods that use an outstanding robot for localization, or where rescue robots only communicate with each other.

This paper is organized as follows. Section 2 defines the problem of this paper and presents the assumptions. In Section 3, the method for generating the sound vector is presented; the results are used to localize a faulty robot in Section 4. In this section, three steps for faulty robot's rescue are examined. Real experiments are presented in Section 5. Conclusions are given in Section 6.

2. Problem Formulation

2.1. Definitions

The rescue mission is said to be successful if a faulty robot $\mathcal{A}_F$ is carried to a designated region by a rescue team. The *success of rescue* is denoted as

$$||\mathbf{P}^G - \mathbf{P}_F|| < \varepsilon, \tag{1}$$

where $\mathbf{P}_F \in \mathbf{R}^2$ is the faulty robot's position, $\mathbf{P}^G \in \mathbf{R}^2$ is the final destination of the faulty robot, and ε is the goal boundary as shown in Fig. 1.

As already mentioned, sound localization uses difference in sound levels between microphones at different locations. Therefore, at least two microphones are needed for localization. To estimate sound location more precisely, we used three microphones, which were equipped on rescue robots. The microphone that is located on the left has the position vector $\mathbf{m}_i^L = (x_i - r\,\mathbf{sin}\theta_i,\ y_i + r\,\mathbf{cos}\theta_i)$ $(i = PL,\ PS)$; the microphone that is located on the right has the position vector $\mathbf{m}_i^R = (x_i + r\,\mathbf{sin}\theta_i,\ y_i - r\,\mathbf{cos}\theta_i)$, and the microphone that is located in the middle has the position vector $\mathbf{m}_i^M = (x_i - r\,\mathbf{cos}\theta_i,\ y_i - r\,\mathbf{sin}\theta_i)$ as shown in Fig. 1. These three position vectors and the microphones' sound levels, E_L,E_R,E_M, combine to form the normalized sound vector, $\mathbf{q}_{PL}$.

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press

Table I. Description of notations in Fig. 1.

Notations	Description
i	Robot's name according to function ($i : F\ (Faulty),\ S\ (Supervisor),\ PS\ (Pusher),\ PL\ (Puller)$).]
k	Microphone's name according to relative position ($k : L\ (Left),\ M\ (Middle),\ R\ (Right)$).
$\mathcal{A}_i$	Robot i.
$\mathbf{p}^G$	Final destination of the faulty robot.
$\mathbf{p}_i$	Robot i's position vector.
$\mathbf{m}_i^k$	Microphone's position vector according to relative position k.
$\mathbf{q}_i$	Robot i's normalized sound vector.
$\boldsymbol{\theta}_i$	Robot i's heading angle.
$\mathbf{f}_i$	Robot i's normalized direction vector.

The rescue team moves to the goal keeping a constant shape. The leader of the rescue team guides its team at the front of the faulty robot. We defined the leader's normalized direction vector as $\mathbf{f}_{PL}$; the direction of the vector should correspond with the direction of the sound vector to arrive final destination.

2.2. Assumptions and notations

The problem presented in this paper is the rescue problem of a faulty robot using colleague robots as illustrated in Fig. 1, and notations of the problem are described as Table I. To formulate the problem, the following assumptions were made:

- Assumption 1: All robots move in a two-dimensional (2D) plane, and the robots are modeled as circles.
- Assumption 2: Reverberation and echo are not considered in the formulation of the problem.
- Assumption 3: The faulty robot cannot move by itself owing to mobility error. However, the functioning of proximity sensors, the communication and functioning of the speaker are normal.
- Assumption 4: A single rescue robot cannot move a faulty robot alone.

Various geometric primitives can be used for robot modeling instead of a circular model in Assumption 1. However, we restricted the robot model so that it had a symmetric shape in all directions in an effort to simplify the rescue problem. Considering reverberation and echo make it difficult to formulate a sound localization problem, we assumed that sound reflections do not exist when formulating the problem. This assumption will be relaxed in a real experiment, as the ILD algorithm is able to localize a sound source by means of the installation of absorption wall. In a multi-robot system, reliability of proximity sensors and communication are relatively high regardless of mobility error.[7] This allows us to use the sensing data acquired by the faulty robot as illustrated in Assumption 3. The last assumption takes into consideration the fact that robots have limited output power.

2.3. Rescue team organization

To save the faulty robot, the rescue team should be organized appropriately. The coordination model of the rescue team is inspired by how people carry large objects such as furniture. One person watches the workspace at a short distance from the object, and another pushes or pulls the object according to the first person's directions. This coordination is efficient to correct direction error and induce a goal. Therefore, we divided rescue robots into three roles to imitate human behavior: *supervisor*, *pusher*, and *puller*. Each robot has different abilities according to its role. A supervisor robot emits a sound signal with a speaker to induce the faulty robot to proceed to a goal point. In addition, the supervisor watches all rescue processes and transmits screen data acquired from a wireless camera to the user. This is done so that the user can take appropriate action in the event of an unexpected situation. A pusher robot pushes the faulty robot from behind. The pusher cannot move the faulty robot alone due to lack of power. This assumption was presented in Section 2.2. Thus, we added a puller robot, which pulls the faulty robot. The puller detects the sound signal in front of the faulty robot, and it has the role of leading the pusher. Hence, the formation of a rescue team is made up of a supervisor, a pusher in the back, and a puller in the front, thereby utilizing the minimum number of robots necessary to perform a successful rescue.

3. Sound Vector Generation Using Triple Microphones

In this section we describe the detailed formulation of a sound vector using a sound signal. To generate the sound vector, the direction and the magnitude have to be known. Determining the direction of the sound vector starts by acquiring candidate regions (CR) of a sound source using the ILD localization scheme proposed by Birchfield and Gangishetty.[16] Two microphones on the robot's left and right are used to generate a *sound isocontour* (SI), and a microphone in the middle of the robot is used to generate a *sound circle* (SC) by the inverse-square law. Finally, we can generate the sound vector by combining the sound isocontour and the sound circle. The magnitude of the sound vector is assumed as a unit vector for a convenient comparison with the formation vector. These processes for the generation of the sound vector are briefly presented in Table II.

3.1. Sound isocontour generation using ILD

Suppose we have N microphones, then $s(t)$ is the sound source signal; $h_k(t)$ is the sound signal received by the microphone located at position k. The measured sound signal can be modeled as

$$h_k(t) = s(t)/d_k + \zeta_k(t)(k = L, R, M), \tag{2}$$

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press

Table II. Method for the generation of a sound vector.

Purpose	Method
Sound isocontour generation	Uses ILD proposed by Birchfield and Gangishetty.[16]
Sound circle generation	Uses inverse-square law.
Sound vector generation	Uses the intersection region of the sound circle and sound isocontour.

where d_k is the distance from the microphone to the sound source, and $\zeta_k(t)$ is the additive Gaussian noise. An index i for each microphone is denominated according to its relative position, *Left(L)*, *Right(R)*, and *Middle(M)*.

Using (2), the energy received by the ith microphone can be calculated as follows:

$$\mathrm{E}_k = \int_0^{\mathrm{T}} h_k^2(t)\,dt = \frac{1}{d_k^2}\int_0^{\mathrm{T}} s^2(t)\,dt + \int_0^{\mathrm{T}} \zeta_k^2(t)\,dt, \quad (3)$$

where T is the time period of the sound source.

To simplify the equation, let us assume that noise is ignored ($\eta = 0$); the coordinate of the sound source is $(x,\ y)$, and the coordinates of the microphones on the left and right microphones are $\mathbf{m}_i^L = (x_i - r\,\mathbf{sin}\theta_i, y_i + r\,\mathbf{cos}\theta_i)$ and $\mathbf{m}_i^R = (x_i + r\,\mathbf{sin}\theta_i, y_i - r\,\mathbf{cos}\theta_i)$. Then we can estimate the position of the sound source as follows:

$$\left(x - \frac{c_x}{c_\Delta}\right)^2 + \left(y - \frac{c_y}{c_\Delta}\right)^2 = \frac{16r^4\mathrm{E}_L\mathrm{E}_R}{c_\Delta^2}, \quad (4)$$

where $c_x = \mathrm{E}_L(x_i - r\sin\theta_i) - \mathrm{E}_R(x_i + r\sin\theta_i)$, $c_y = \mathrm{E}_L(y_i + r\cos\theta_i) - \mathrm{E}_R(y_i - r\cos\theta_i)$, $c_\Delta = \mathrm{E}_L - \mathrm{E}_R$ and (4) can also be written as (5) if the distance from the sound source to the left and right microphones is the same, $\mathrm{E}_L = \mathrm{E}_R$,

$$2c_x x + 2c_y y = c, \quad (5)$$

where $c = (\mathrm{E}_L - \mathrm{E}_R)(x_i^2 + y_i^2 + r_i^2 - 2rx_i\sin\theta_i + 2ry_i\cos\theta_i)$. As shown in Fig. 2, we can generate a 3D isocontour by (4) and (5) with the variation of energy level, and the shape is defined as sound isocontour.

In (4), c_Δ is the difference in sound level between the left and right microphones. If c_Δ is bigger, then the radius of sound isocontour is smaller. It means that the estimated region of the sound source decreased. As a result, we can estimate the location of the sound source more accurately.

3.2. Sound circle generation using inverse-square law

Using sound isocontour, we can roughly estimate the location of the sound source. The estimated location, however, is not reliable because the region of sound isocontour is very wide. Thus, additional information is needed to find the direction of the sound source precisely.

We use sound attenuation to acquire additional information. As the sound propagates, it is attenuated by a factor inversely proportional to the distance traveled. This is called the inverse-square law of attenuation. Applying this law, we can estimate the distance from the sound source to the measurement point. A new function $r_{SC}(\mathbf{x_1},\mathbf{x_2})$ can be calculated by (6),

$$r_{SC}(\mathbf{x_1}, \mathbf{x_2}) = \frac{k}{\|\mathbf{x_1} - \mathbf{x_2}\|} \quad (\mathbf{x_1} \neq \mathbf{x_2}), \quad (6)$$

where $\mathbf{x_1}, \mathbf{x_2}$ are the measurement point and the sound source point respectively, and k is an empirically determined proportionality constant. In (6), $\mathbf{x_1}$, $\mathbf{x_2}$ are not the same because the rescue robot does not collide with the faulty robot, which emits a sound signal. Using (6), we can define

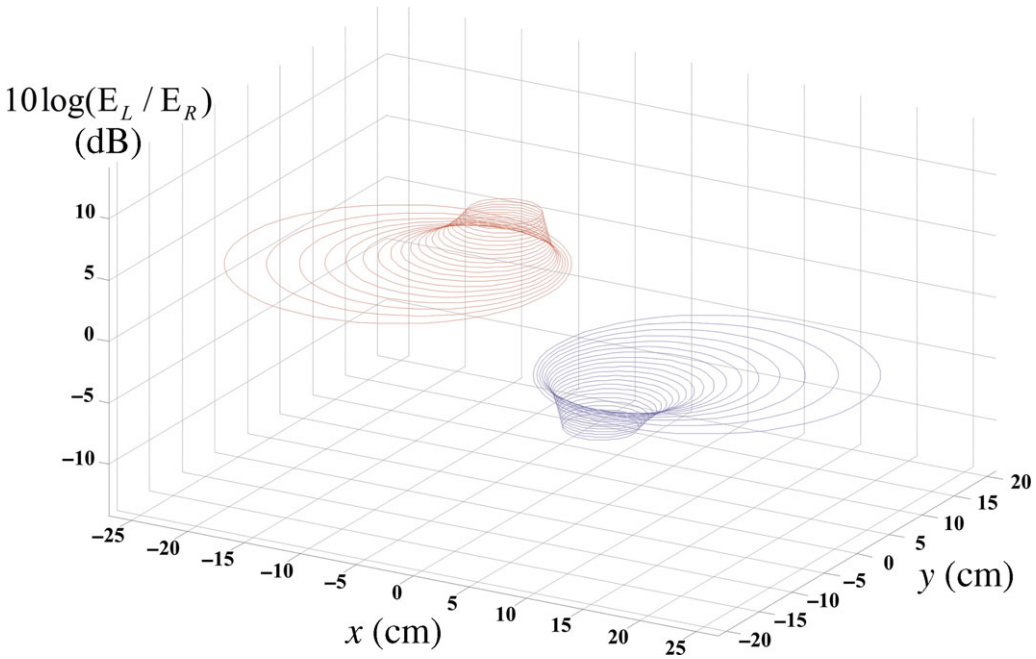


Fig. 2. (Colour online) 3D sound isocontour. The isocontour is drawn differently according to the ratio of the sound level.

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press

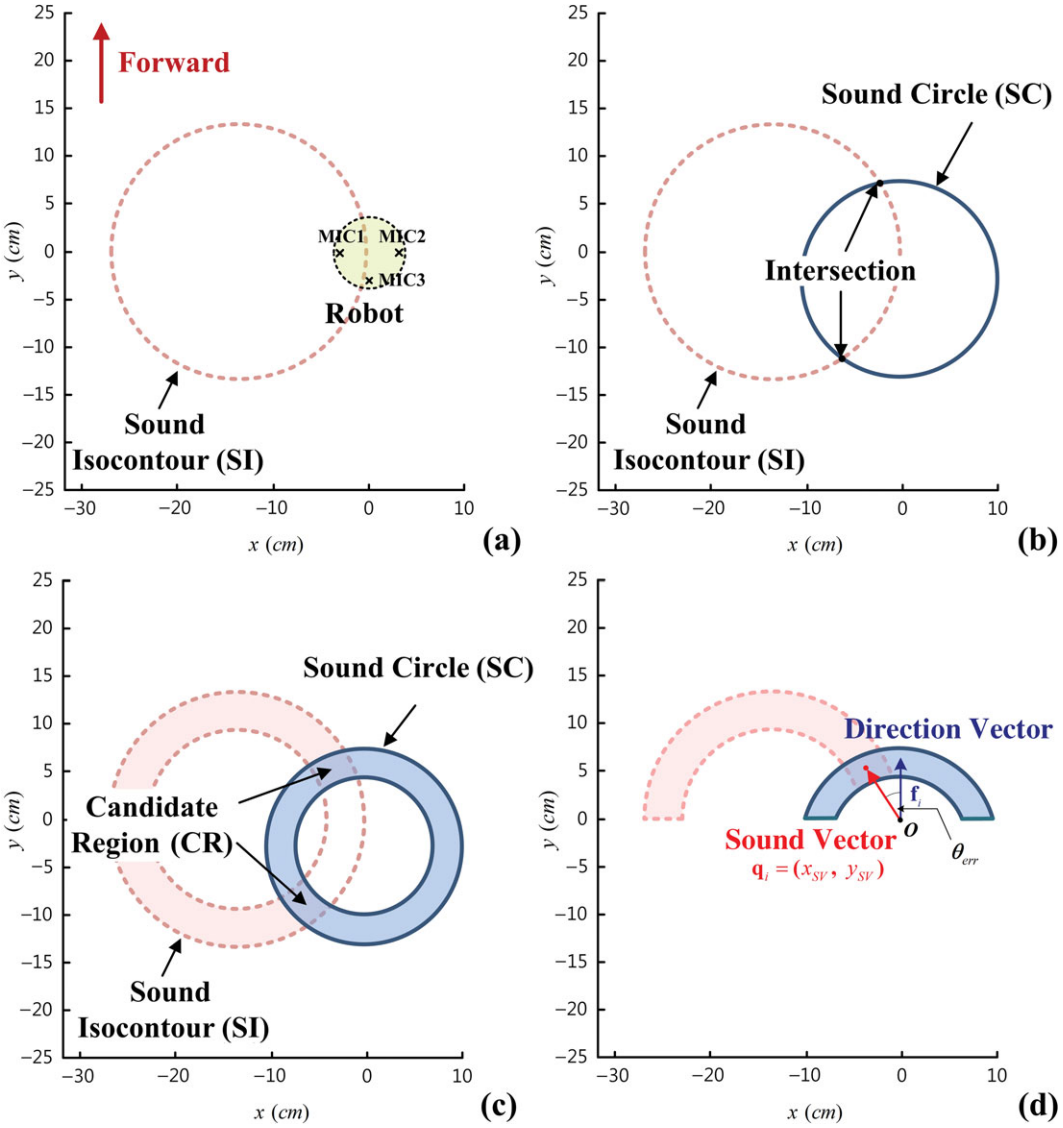


Fig. 3. (Colour online) Process for the generation of sound vector. The center of the robot is the origin of Cartesian coordinate. (a) Sound isocontour generation using ILD with two microphones. (b) Sound circle generation using inverse-square law with one microphone. (c) Sound source's candidate region generation using sound isocontour and sound circle. (d) Sound vector generation. The front–back confusion is eliminated by comparison with triple microphones' relative sound level.

a set from the point to the same distance: sound circle. The center point of the sound circle is the measurement point and the radius of the sound circle is the estimated distance between the sound source and the measurement point. The sound circle can be written as follow:

$$(x - x_i)^2 + (y - y_i)^2 = \{r_{SC}(\mathbf{x}_i, \mathbf{x}_F)\}^2, \tag{7}$$

where (x_i, y_i) is a coordinate of a pusher or puller.

3.3. Sound vector generation

So far we have derived the sound isocontour using ILD with two microphones, and the sound circle using the inverse-square law with one microphone. Combining with these two circles, we can get the two intersection points of the circles. These points can possibly locate the sound source as shown in Fig. 3(b) and calculated by Pythagoras theorem applying to (4) and (7):

$$x_{SV} = x_i + r_{SC}\cos\left(\tan^{-1}\frac{Y}{X} \pm \cos^{-1}\left(\frac{r_{SC}^2 - r_{SI}^2 + D^2}{2dD}\right)\right),$$
$$y_{SV} = y_i + r_{SC}\sin\left(\tan^{-1}\frac{Y}{X} \pm \cos^{-1}\left(\frac{r_{SC}^2 - r_{SI}^2 + D^2}{2dD}\right)\right), \tag{8}$$

where $X = \frac{c_x}{c_\Delta} - x_i$, $Y = \frac{c_y}{c_\Delta} - y_i$, $r_{SC} = r_{SC}(\mathbf{x}_i, \mathbf{x}_F)$, $r_{SI} = \frac{4r^2\sqrt{E_L E_R}}{c_\Delta}$, and $D = \sqrt{X^2 + Y^2}$.

However, the accuracy of the points is low since sound levels are constantly changing. One measured sound level is not reliable because noise and reverberation can be mixed with the direct sound during travel. Therefore, we used sound data from five measurements to find the intersection precisely. Consequently, the intersection is not a point but

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press

regions, which are similar to a circular ring. The minimum calculated radius is the inner radius of the ring and the maximum radius is the outer radius of the ring as shown in Fig. 3(c). These regions are denominated as the *candidate region*.

If the ratio of the distance from the sound source to the two microphones is the same, we cannot know whether the candidate region is in front or not. This confusion is called the front–back confusion,[1] which occurs because of an identical difference in the sound level. To eliminate this confusion, the sound level on the middle microphone (E_M) is compared with the average sound level between the left (E_L) and right (E_R) microphones. If the average of sound levels at the left and right microphones is bigger than the sound level at the middle microphone, then the sound source is in front as shown in (9). If it is the other way, then the sound source is in back. We can reduce to a half the region through (9) compared with the original candidate region. This reduction of the candidate region can also be described by θ_{err}, which is an included angle between the sound vector and the direction vector of the puller. If the θ_{err} is in quadrants I and IV, then the sound source is located in the front of the measurement point. On the other hand, if θ_{err} is in quadrants II and III, the sound source is located in the back of the measurement point,

$$\text{direction of sound source} = \begin{cases} \textit{forward} & \textit{if } E_M \le \frac{E_L+E_R}{2} \text{ or } -90^\circ \le \theta_{\text{err}} \le 90^\circ \\ \textit{backward} & \textit{if } E_M > \frac{E_L+E_R}{2} \text{ or } 90^\circ < \theta_{\text{err}} < 270^\circ \end{cases}, \tag{9}$$

where $\theta_{\text{err}} = \tan^{-1}\frac{Y}{X} \pm \cos^{-1}\left(\frac{r_{SC}^2 - r_{SI}^2 + D^2}{2dD}\right)$.

Consequently, we can estimate the location of the sound source; it is defined as the sound vector as shown in Fig. 3(d).

4. Cooperative Control of Rescue Robots Using Sound Vector

4.1. Role assignment of the rescue team

The rescue robot team is heterogeneous, and each rescue robot thus has different capabilities according to their roles. For example, the supervisor is equipped with the wireless camera to transmit the accident scene to the user. It also has a speaker to emit sound signals. The puller and the pusher have a manipulator to grasp an object, and the puller detects sound signals from the faulty robot or from the supervisor robot using three microphones. Therefore, the roles of rescue robots are predetermined by their capabilities before the rescue of the faulty robot. When a robot failure occurs, rescue robots are thrown into an accident scene according to their inherent roles. First, the supervisor moves to the faulty robot and verifies that the scene is secured. Second, the puller moves to the faulty robot and takes its position in front of the faulty robot. Likewise, the pusher is positioned behind the faulty robot.

Table III. Notations for Figs. 4 and 5.

Notation	Description
x_i, y_i	x, $y$coordinates of robot i.
ϕ_i	Included angle between robot i and the faulty robot's center.
r	Radius of robot.
s_i^j	Contact sensor of robot i at position j $(0 \le j \le 5)$.
$\mathbf{S}_i^k$	Contact sensor set of robot i at state k (k:D(disabled), R(ready), T(total)).

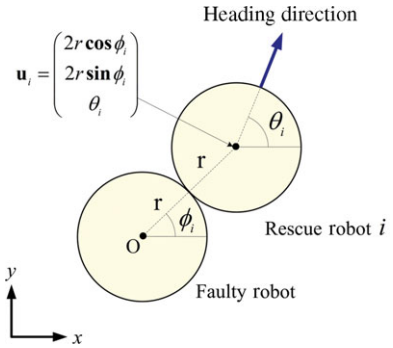


Fig. 4. (Colour online) Posture coordinates of robot i.

4.2. Position assignment of rescue robots

Moving a faulty robot is possible only when each rescue robot is located in a position at which the rescue robots can apply force to the faulty robot, with pusher's and puller's heading directions nearly equal so as to maximize the applied force. To formulate the problem, we defined the following notations as shown in Table III.

We defined *posture coordinates* to include 2D coordinates and heading direction as shown in Fig. 4. The posture coordinates are expressed in (10):

$$\mathbf{u}_i = (\mathbf{p}_i, \theta_i)^{\mathrm{T}} = (x_i, y_i, \theta_i)^{\mathrm{T}} = (2r\,\mathbf{cos}\phi_i, 2r\,\mathbf{sin}\phi_i, \theta_i)^{\mathrm{T}}, \tag{10}$$

where ϕ_i is the angle between robot i and the faulty robot's center, θ_i is the heading angle of robot i, and r is the radius of robot.

As shown in Fig. 1, the puller is in front of the faulty robot, and the pusher is behind the faulty robot to complete the rescue mission. This position is defined as the final posture coordinate ${}^A\mathbf{u}_i$, which is expressed as the multiplication of transitional matrix and rotation matrix,

$${}^A\mathbf{u}_i = \mathbf{R}(\theta_i)\mathbf{T}(\mathbf{p}_i, \mathbf{p}_F^S){}^B\mathbf{u}_i \quad (i = PL,\ PS), \tag{11}$$

$$\mathbf{R}(\theta_i) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{\pi}{2\theta_i} \end{bmatrix}, \mathbf{T}(\mathrm{p}_i, \mathbf{p}_F^S) = \begin{bmatrix} 1 & 0 & \frac{1}{\theta_i}d_x \\ 0 & 1 & \frac{1}{\theta_i}d_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{12}$$

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press

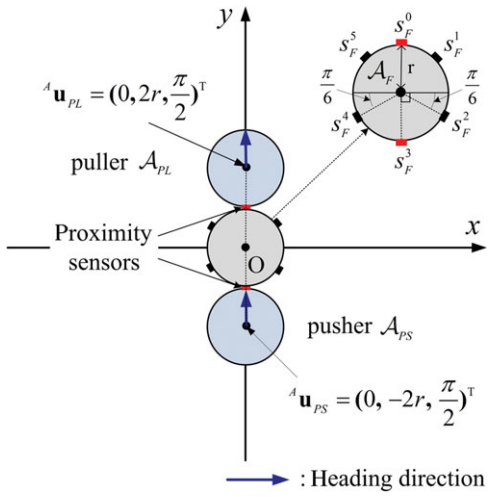


Fig. 5. (Colour online) Position assignment of rescue robots. The puller should be in front of the faulty robot, and the pusher should be behind the faulty robot just before the rescue team escorts the faulty robot.

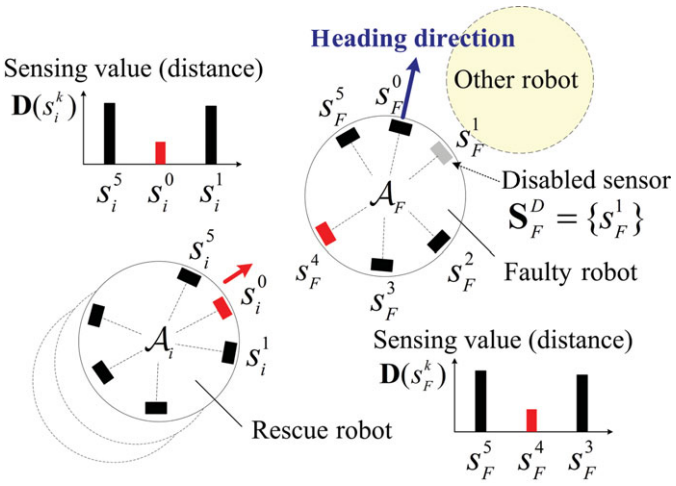


Fig. 6. (Colour online) Approach process of rescue robots. This is just an example of approaching a faulty robot. Rescue robots can approach the faulty robot from any direction.

where $\mathbf{R}$ is the rotational matrix, $\mathbf{T}$ is the transitional matrix, d_x,d_y are x, y coordinates of direction vectors from the present point to the goal point, and ${}^{B}\mathbf{u}_i$ and ${}^{A}\mathbf{u}_i$ are the initial and final posture coordinates.

All rescue robots have a direction vector that goes straight when the rescue mission starts; the heading direction is $\frac{\pi}{2}$. The puller and pusher are located at $\mathbf{p}_{PL} = (0, 2r)$ and $\mathbf{p}_{PS} = (0, -2r)$ as shown in Fig. 5.

Two proximity sensors which are located at ${}^{A}\mathbf{u}_i$ $(i = PL,\ PS)$ are defined as $\mathbf{S}_F^R = \{s_F^0, s_F^3\}$ in the sequence among the $\mathbf{S}_F^T = \{s_F^j | 0 \le j \le 5\}$ of $\mathcal{A}_F$ proximity sensor arrays as shown in Fig. 5. These sensors have the role to make sure that all rescue robots are in their assigned positions by measuring relative distance to the faulty robot.

4.3. Rescue process of a faulty robot

Rescue processes are divided into four steps as shown in Table IV. First, if the sound is emitted as a distress signal, then each rescue robot approaches the faulty robot using the $Move - to - sound$ command as shown in Fig. 6. The distance between the faulty robot $\mathcal{A}_F$ and the rescue robot $\mathcal{A}_i$ is

$$d_F^i = \min(\mathbf{D}(\cup_j s_F^j - \mathbf{S}_F^D, \cup_j s_i^j)) \qquad (0 \le j \le 5), \quad (13)$$

where $\mathbf{D}(\mathbf{S}_F, \mathbf{S}_i)$ is the output distance between proximity sensors belonging to $\mathbf{S}_F$ and $\mathbf{S}_i$, and $\mathbf{S}_F^D$ is the exclusive sensor's set at the measuring distance because another robot has already been contacted.

Second, if the rescue robot succeeded in contacting the faulty robot ($d_F^i \le \varepsilon$), then it turns on the faulty robot's border keeping a safe distance and stops at the assigned position satisfying (14). This process is called the $Follow - in - contact$. The faulty robot cannot use the proximity sensor located at the assigned position because the rescue robot covers the sensor. Thus, this disabled proximity sensor should be added to $\mathbf{S}_F^D$ after the rescue robot's stop,

$$\min(\cup_j s_F^j - \mathbf{S}_F^D) \in \mathbf{S}_F^T \quad \text{and} \quad d_F^i \le \varepsilon. \qquad (14)$$

Third, the $Aligned - the - direction$ command carries out the rescue robots' spin to adjust the heading direction $\frac{\pi}{2}$ at the assigned position. In the process, the rescue robots check their proximity sensors whether sensing something or not. Each rescue robot has a limited heading direction condition as shown in (15),

$$\begin{aligned} &\mathcal{A}_{PL} : \min(\textstyle\bigcap_j s_{PL}^j) > \varepsilon (j \ne 3) \text{ and } s_{PL}^3 \le \varepsilon, \\ &\mathcal{A}_{PS} : \min(\textstyle\bigcap_j s_{PS}^j) > \varepsilon (j \ne 0) \text{ and } s_{PS}^0 \le \varepsilon, \end{aligned} \qquad (15)$$

where ε is the constant that is used for "sensing or not." If the proximity sensor detects a distance smaller than ε, then it means that a robot is present at the location.

Finally, if the rescue robots are ready to move, then $Move - to - goal$ command is carried out. The puller robot

Table IV. Movement order of rescue robots.

Command	Description
$Move - to - sound$	Rescue robot moves to the faulty robot emitting sound source.
$Follow - in - contact$	Rescue robot moves to the assigned position maintaining constant distance with the faulty robot.
$Aligned - the - direction$	Rescue robot's heading direction is aligned to the goal.
$Move - to - goal$	Rescue team escorts faulty robot to goal.

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press

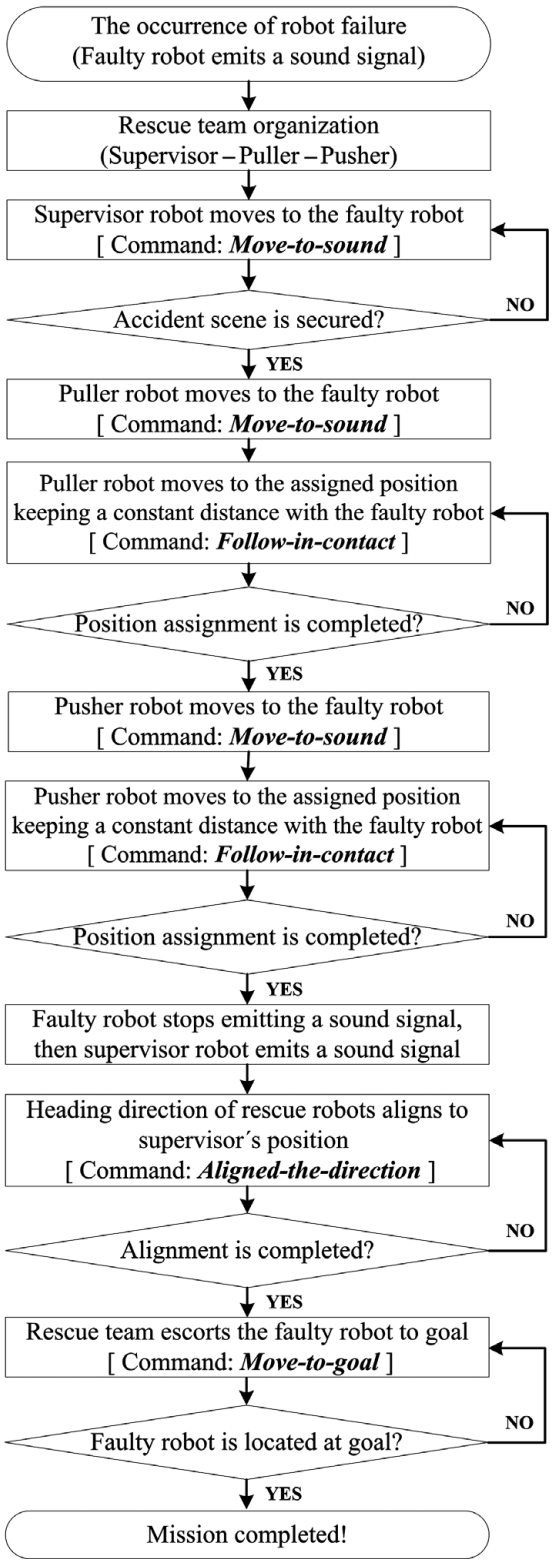


Fig. 7. Rescue process of a faulty robot.

pulls the faulty robot in front and adjusts its heading direction to correspond with sound signal direction. The pusher robot pushes the faulty robot from behind for supplying force.

The entire process of the rescue mission is shown in Fig. 7. When a robot experiences a system failure, the faulty robot emits a sound signal. Next, a rescue robots

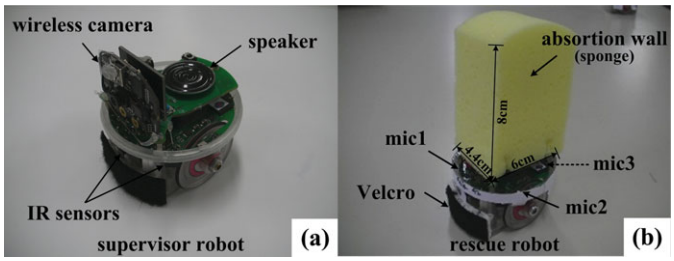


Fig. 8. (Colour online) The variety of sensors and components of robots.

team is organized into three components: supervisor, puller, and pusher. The supervisor robot moves to the faulty robot using sound signals. The supervisor checks whether the failure accident scene is secured or not. If the supervisor is positioned at an appropriate location at which the user can cover the scene through a wireless camera, then the puller and pusher move to the faulty robot via sound signals. The puller and pusher use proximity sensors to assign positions close to the faulty robot. The faulty robot stops emitting sound signals after the rescue robots are located at the desired positions, and the supervisor emits sound signals to induce the rescue robot team. According to the sound signal of the supervisor, the heading direction of rescue robots aligns with the supervisor's position, which is the goal point. When the alignment of rescue robots is completed, rescue team escorts the faulty robot to the goal.

5. Experimental Results

We did experiments to evaluate the success of the proposed method. The experiments are divided into sections depending on their purpose. First, we investigated the estimation of the sound vector by varying the distance and direction and comparing it to the real value. Second, we described rescue processes and analyzed each process: approach to the faulty robot, assigning the positions of the rescue robots, and saving the faulty robot using the rescue team.

5.1. Robot platform and experimental environment

To validate the proposed method, four robots were needed. These were the supervisor, the puller, the pusher, and the faulty robots. We used four e-puck robots equipped with a variety of sensors, including eight infrared radiation sensors, three microphones, and a speaker. The diameter of the e-puck, which is a circular shape robot, is 7 cm, the maximum sensing range is 6 cm, and the maximum speed is 15 cm/s. We restricted the maximum speed to 2 cm/s considering the short sensing range. A wireless camera was attached to the supervisor robot as shown in Fig. 8(a). Velcro replaced the robot manipulator for simple grasping as shown in Fig. 8(b). The experiments were conducted in a normal indoor space ($7.5 \times 5.0 \times 2.7\text{m}^3$) with no sound-absorbing materials to mimic a real workspace. In such a situation, the accuracy of the proposed sound localization method can be highly dependent on the hardware, i.e., the performance and geometrical array of microphones, and the type of sound signal. These factors are difficult to change due to

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press

Table V. Estimation of sound vector's direction.

Distance (d) / Angle (θ)	10 cm	20 cm	30 cm	40 cm	50 cm
180°	176.9	162.1	159.8	151.1	135.8
	(3.1)	(17.9)	(20.2)	(28.9)	(44.2)
210°	206.1	199.8	192.3	176.5	179.5
	(3.9)	(10.2)	(17.7)	(33.5)	(30.5)
240°	233.1	229.7	223.0	217.9	169.9
	(6.9)	(10.3)	(17.0)	(22.1)	(70.1)
270°	220.4	249.0	226.0	230.3	213.7
	(49.6)	(21.0)	(44.0)	(39.7)	(56.3)

Values inside parentheses show the estimation error of the real sound vector's direction angle.

the restriction of robot platform. In addition, the distance between the microphones is relatively short, at 6.2 cm, which leads to low localization accuracy because the sound localization method using ILD has a wide candidate region when a small difference in sound levels exists according to an earlier study.[4] To solve these problems related to limited hardware specifications, we installed an absorption wall ($4.4 \times 6 \times 8$ cm^3) onto the rescue robots as shown in Fig. 8(b). The absorption wall makes ILD between the microphones high, which raises the accuracy of the sound localization process. This additional equipment is considered as a component of our system.

5.2. *Results of the sound vector estimation*

Sound level is affected by the location of the microphone. Thus, we estimated the direction angle of the sound vector by varying the measuring distance (d) and the angle (θ) between the robot and the microphone. To estimate the sound vector's direction angle, we measured all sound levels from the microphone while measuring the distance, which varied from 10 to 50 cm but maintaining an angle of 180°. As the traveled distance increased, the influence of the noise also increased. Thus, the candidate region was getting wider because the radius of the sound circle increased rapidly through small level differences at a long distance. As a result, the error of the estimated direction angle of the sound vector increased. More seriously, the accuracy of the direction angle decreased when the "left–right" distinction of the location of the sound source was wrong. Sound isocontour was added not only on one side but also on the other side. The other side was used to calculate the candidate region with the wrong data. In real experiments this confusion appeared at a measuring distance of more than 30 cm; the accuracy of the estimated direction vector decreased rapidly as shown in Table V.

5.3. *The rescue result of a faulty robot*

A faulty robot emitted a sound signal, and a puller detected this sound signal; then it approached the faulty robot as shown in Fig. 9(a). As described in Section 3.3, the puller's heading direction continuously changed at a relatively long detecting distance because of the sound vector estimation errors. Although the direction of the sound vector was not correct, the distinction of left and right was reliable.

Table VI. Success rate according to rescue step.

Rescue step	Traveled time[a] (s)	Success rate (%)
Move – to – sound	18	93
Follow – in – contact	27	83
Aligned – the – direction	3	93
Move – to – goal	33	70
Total Time/Success Rate	81	62

[a]Traveled time is relevant to the pusher only.

Therefore, the wrong path by trial and error could be corrected in real time. In addition, as the sound signal came closer, the accuracy of the estimation became higher. Thus, the puller succeeded in approaching the position of the faulty robot. A supervisor watched these processes, and transmitted to user from wireless camera as shown in Fig. 9(b).

When the puller robot arrived at the faulty robot, the next step was to move in front of the faulty robot. For traveling around the faulty robot, the puller robot adjusted its heading direction to the right using the puller's equipped infrared radiation sensor. Then the puller robot moved in front of the faulty robot maintaining a regular distance. If front sensor detected the faulty robot, the puller robot stopped and contacted the faulty robot using Velcro. The whole process took 27 s, and is illustrated in Fig. 10 .

In Fig. 10, the right graph illustrates the IR sensor data when the puller moved around the faulty robot. When the puller started to move to s_F^3 at 18 s, the puller moved to the right around the faulty robot and the measuring distance decreased in the order of s_F^2,s_F^1. Finally, when the puller arrived at the front of the faulty robot, the s_F^0 decreased and the puller stopped. The pusher followed the same procedure as the puller except moving around the faulty robot.

If the pusher and puller robots were ready to escort the faulty robot, they applied force to the faulty robot as illustrated in Fig. 11. The puller decided the heading direction and pulled the faulty robot by detecting the sound signal. The pusher applied force to help move the faulty robot. The two rescue robots headed for the direction generated by the supervisor robot's sound signal. Rescue process updated continuously until the rescue operation was completed, which means the faulty robot was in the goal region (ε). In the experiment, we decided the radius of the goal region as 9 cm in consideration of the radius of robot and extra distance.

Success rate according to rescue step is illustrated in Table VI. We attempted the same experiment 30 times, and averaged the traveled time. $Move - to - sound$ and $Aligned - the - direction$ commands are rarely fail because turning around at present place was not difficult. The success rate of $Follow - in - contact$ step was more than 80%, it was good to apply system. However, success rate of $Move - to - goal$ was relatively low because the puller could not change heading direction easily during pulling the faulty robot. Consequently, although each step has relatively high success rate, total success rate was 62%.

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press

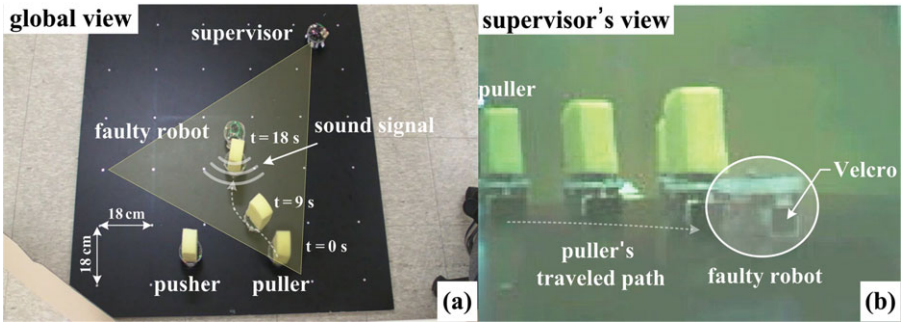


Fig. 9. (Colour online) Puller's approach to a faulty robot. (a) A puller approached the faulty robot emitting a sound signal. (b) This process was transmitted to user from the supervisor robot.

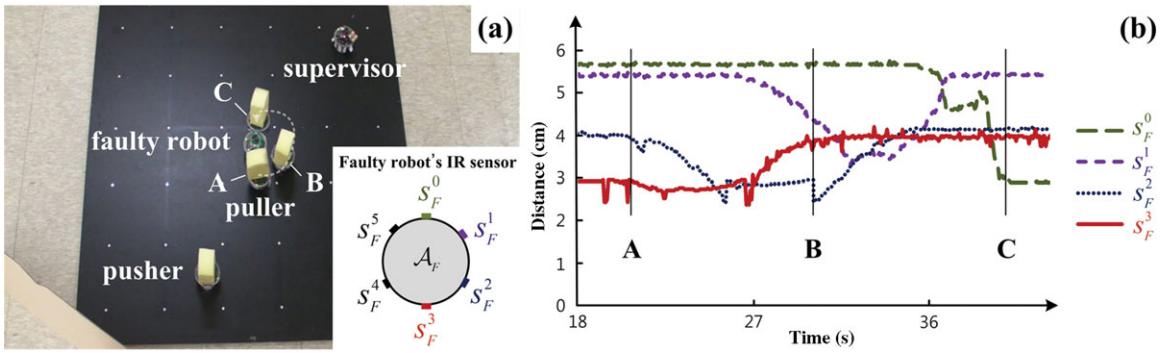


Fig. 10. (Colour online) The puller moved in front of the faulty robot. (a) The whole processes. (b) Variation of the IR sensor data according to puller's movement. The puller turned right at A, the puller was located to the right side of the faulty robot at B, and C is complete time the puller contacted the front of the faulty robot. The maximum measured distance was different from sensor to sensor because sensors' performance is not good. Sensors s_F^0,s_F^1's maximum measured distance is about 5.5 cm, and s_F^2,s_F^3's maximum measured distance is about 4 cm.

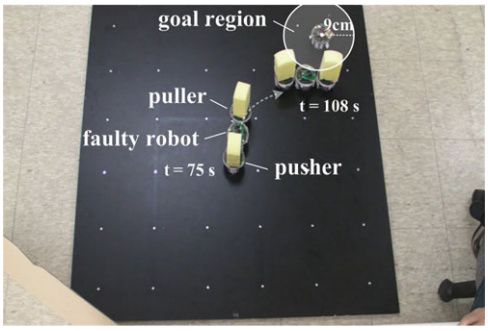


Fig. 11. (Colour online) The rescue team escorted the faulty robot and moved it to goal region. When the faulty robot was in the goal region, rescue mission was completed. In this step, traveled time was 33 s, and the total traveled time was 108 s.

6. Conclusions

The results of this study indicate that a faulty robot can be retrieved by rescue robots using a cooperative control scheme. Each rescue robot generates a sound vector to identify the position of the faulty robot with a sound signal. A combination of sound isocontour and sound circle was used for the generation of sound vector, and the sensing data from the faulty robot were used to assign the positions of the rescue robots. The proposed method actively uses interactive real-time communication between the faulty robot and the rescue robots for data sharing. Through these processes the faulty robot was automatically saved by the rescue robots. Consequently, it is hoped that our approach will serve as a method for managing multi-agent systems.

Acknowledgements

This work was supported in part by a Korea Science and Engineering Foundation (KOSEF) NRL Program grant funded by a Korean government (MEST) (No. R0A-2008-000-20004-0), in part by the Brain Korea 21 Project, in part by ASRI, and in part by the Industrial Foundation Technology Development Program of MKE/KEIT (Development of Collective Intelligence Robot Technologies (CIRT)).

References

1. H. Li, S. Ishikawa, Q. Zhao, M. Ebana, H. Yamamoto and J. Huang, "Robot Navigation and Sound-Based Position Identification," **In**: *Proceedings of the IEEE International Conference on Systems, Mans and Cybernetics*, Montreal (2007) pp. 2449–2454.
2. J. A. Pérez, J. A. Castellanos, J. M. M. Montiel, J. Neira and J. D. Tardos, "Continuous Mobile Robot Localization: Vision vs. Laser," **In**: *Proceedings of the IEEE Conference on Robotics and Automation*, Detroit, Michigan (1999) pp. 2917–2923.
3. F. Dellaert, W. Burgard, D. Fox and S. Thrun, "Using the condensation algorithm for robust, vision-based mobile robot localization," **In**: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition II*, Fort Collins, Colorado (1999) pp. 588–594.

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press

4. S. Se, D. Lowe and J. Little, "Global Localization Using Distinctive Visual Features," **In**: *Proceedings of the International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland (2002) pp. 226–231.
5. B. D. Yim, S. Y. Hwang and J. B. Song, "Mobile Robot Localization Based on Fusion of Vision and Range Information," **In**: *Control and Automation System Symposium*, Ilsan, Republic of Korea (2006) pp. 183–188.
6. J. J. Little, J. Lu and D. R. Murray, "Selecting Stable Image Features for Robot Localization Using Stereo," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, B.C., Canada (1998) pp. 1072–1077.
7. S. Stancliff, J. Dolan and A. Trebi-Ollennu, "Towards a Predictive Model of Mobile Robot Reliability," *Technical Report*, CMU-RI-TR-05-38, Carnegie Mellon University, Pittsburgh, PA (2005).
8. P. Aarabi, "Robust Multi-Source Sound Localization Using Temporal Power Fusion," *Proceedings of Sensor Fusion: Architectures, Algorithms, and Applications V (AeroSense'01)*, Orlando, Florida (2001).
9. A. F. T. Winfield and J. Nembrini, "Safety in numbers: Fault tolerance in robot swarms," *Int. J. Model. Ident. Control* **1**(1), 30–37 (2006).
10. Y. U. Cao, A. S. Fukunaga and A. B. Kahng, "Cooperative mobile robots: Antecedents and directions," *Auton. Robots* **4**, 7–27 (1997).
11. M. J. Mataric, M. Nilsson and K. T. Simsarian, "Cooperative Multi-Robot Box-Pushing," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, Pennsylvania USA (1995) pp. 556–561.
12. F. L. Wightman and D. J. Kistler, "The dominant role of low-frequency interaural time differences in sound localization," *J. Acoust. Soc. Am.* **91**(3), 1648–1661 (1992).
13. B. P. Gerkey and M. J. Mataric, "Pusher-Watcher: An Approach to Fault-Tolerant Tightly Coupled Robot Coordination," **In**: *Proceedings of the International Conference on Robotics and Automation*, Washington, D.C. (2002) pp. 464–469.
14. J. Backman and M. Karjalainen, "Modelling of Human Directional and Spatial Hearing Using Neural Networks," **In** *Proceeding of the ICASSP-93*, Vol. I, Minneapolis, MN (1993) pp. 125–128.
15. J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization* (MIT Press, Cambridge, MA, 2001).
16. S. T. Birchfield and R. Gangishetty, "Acoustic Localization by Interaural Level Difference," **In**: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, Vol. 4 (2005) pp. 1109–1112.
17. W. Cui, Z. Cao and J. Wei, "Dual-microphone source location method in 2-D space," **In**: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* Vol. 4, Toulouse, France (2006) pp. 845–848.
18. M. Raspaud, H. Viste and G. Evangelista, "Binaural source localization by joint estimation of ILD and ITD," *Trans. Audio Speech Lang. Process.* **18**(1), 68–77 (2010).
19. L. E. Parker, "Heterogeneous Multi-Robot Cooperation," *Ph.D. Thesis* (Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1994).
20. Z. Wang and V. Kumar, "Object Closure and Manipulation by Multiple Cooperation Mobile Robots," **In**: *Proceedings of the International Conference on Robotics and Automation*, Vol. 1, Washington, D.C. (2002) pp. 394–399.
21. G. A. S. Pereira, V. Kumar, J. R. Spletzer, C. J. Taylor and M. F. M. Campos, "Cooperative Transport of Planar Objects by Multiple Mobile Robots Using Object Closure," **In**: *8th Symposia on Experimental Robotics, International Experimental Robotics VIII, Springer Tracts in Advanced Robotics* Vol. 5, (2003) pp 275–284.
22. K. Kovac, I. Zivkovic and B. Dalbelo Basic, "Simulation of Multi-Robot Reinforcement Learning for Box-Pushing Problem," **In**: *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (MELECON 2004)* Vol. 2, Dubrovnik, Croatia (2004) pp. 603–606.
23. Y. Li and X. Chen, "Modeling and Simulation of a Swarm of Robots for Box-Pushing Task," *Proceedings of the 12th Mediterranean Conference on Control and Automation*, Kusadasi, Aydin, Turkey (2004).
24. Y. Wang and C. W. de Silva, "Multi-Robot Box-Pushing: Single-Agent Q-Learning vs. Team Q-Learning," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China (2006) pp. 3694–3699.
25. Y. Inoue, T. Tohge and H. Iba, "Cooperative transportation system for humanoid robots using simulation-based learning," *Appl. Soft Comput.* **7**(1), 115–125 (2007).
26. F. R. Noreils, "Toward a robot architecture integrating cooperation between mobile robots: Application to indoor environment," *Int. J. Robot. Res.* **12**(1), 79–98 (1993).
27. M. Dorigo and E. Sahin, "Guest editorial. Special issue: Swarm robotics," *Auton. Robots* **17**(2–3), 111–113 (2004).
28. J. S. Choi, G. Eoh, S. Lee, J. Kim and B. H. Lee, "Collision Prevention System Using Velocity Obstacle and Artificial Potential Field," *Proceedings of the International Symposium on Intelligent Systems*, Hachioji, Japan (2010).
29. J. S. Choi, G. Eoh, J. Kim, Y. Yoon, J. Park and B. H. Lee, "Analytic Collision Anticipation Technology Considering Agents' Future Behavior," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan (2010) pp. 1656–1661.
30. J. Vermaak, M. Gangnet, A. Blake and P. Pérez, "Sequential Monte Carlo Fusion of Sound and Vision for Speaker Tracking," **In**: *Proceedings of the IEEE International Conference on Computer* Vision, Vol. I, Vancouver, Canada (2001) pp. 741–746.

https://doi.org/10.1017/S0263574713000374 Published online by Cambridge University Press