An Extended Any-angle Path Planning Algorithm for Maintaining Formation of Multi-agent Jellyfish Elimination Robot System

Hanguen Kim, Donghoon Kim, Hyungjin Kim, Jae-Uk Shin, and Hyun Myung*

International Journal of Control, Automation and Systems 14(2) (2016) 598-607

ISSN:1598-6446 (print version)
eISSN:2005-4092 (electronic version)

To link to this article:
http://dx.doi.org/10.1007/s12555-014-0349-0

International Journal of Control, Automation and Systems 14(2) (2016) 598-607
http://dx.doi.org/10.1007/s12555-014-0349-0
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

An Extended Any-angle Path Planning Algorithm for Maintaining Formation of Multi-agent Jellyfish Elimination Robot System

Hanguen Kim, Donghoon Kim, Hyungjin Kim, Jae-Uk Shin, and Hyun Myung*

Abstract: In recent years, the increasing influence of climate change has given rise to an uncontrolled proliferation of jellyfish in marine habitats that has visibly damaged many ecosystems and industries and poses a threat to human life. To resolve this issue, our team developed a robotic system called JEROS (Jellyfish Elimination RObotic Swarm) to successfully and efficiently remove jellyfish. JEROS consists of multiple unmanned surface vehicles that freely move in a marine environment to scavenge for and eliminate jellyfish. For controlling formation of JEROS, the leader-follower scheme is used, but this can be sometimes difficult to apply in an ocean environment. When the follower robots are tracking in accordance with the leader's following route without the performance limitation of the robot being considered, the formation cannot be well maintained even if a formation control algorithm is applied to the robots. Maintaining formation is important for efficiency of the jellyfish removal operation. If the formation cannot be well maintained while the robots are moving, the operation area becomes irregular and consequently, the removal operation entails performing repetitive tasks. Therefore, in this paper, we propose the extended any-angle, named extended ARC (Angular-Rate-Constrained)-Theta* path planning algorithm for maintaining formation of the JEROS system to enhance the efficiency of jellyfish removal. To evaluate the performance of the proposed path planning algorithm, we performed field tests at Bang-dong Reservoir in Daejeon, South Korea.

Keywords: Formation control, multi-agent robot, path planning, unmanned surface vehicle.

1. INTRODUCTION

Recently, the proliferation of jellyfish has emerged as an important environmental issue. Jellyfish have threatened marine ecosystems and caused enormous damage to marine-related industries in more than 14 countries around the world. In South Korea, the overall financial damage to marine-related industries was estimated to be over 300 million U.S. dollars per year in 2009 [1]. In particular, the fishing industry, seaside power plants, and oceanic tourism enterprises have been the most significantly affected. The most prevalent species of jellyfish along the coast of South Korea are *Aurelia aurita* and *Nemopilema nomurai*. Some jellyfish species such as *Nemopilema nomurai* even have venom that can be lethal to humans.

To solve this problem, a number of studies pertaining to jellyfish removal have been actively undertaken. In a previous study, a system consisting of two trawl boats equipped with jellyfish cutting nets was developed [2, 3]. Utilizing large ships and many human operators, the system has shown high performance in jellyfish removal, but it was limited because of its difficulty in operating in narrow and shallow coastal areas. In addition, numerous other systems were developed for the purpose of preventing the influx of jellyfish into water intake pipes of power plants. One of these systems consisted of a camera and a water pump [4]. Yet another system utilized a bubble generator and a conveyor device [5]. However, these types of systems are very expensive to install and maintain. To provide a cost-effective solution to the jellyfish problem, an earlier version of the autonomous jellyfish removal robot system, JEROS (Jellyfish Elimination RObotic Swarm), was presented in [6–9]. The design of the ship, navigation, image-processing algorithms, and feasibility tests for the algorithms and jellyfish removal were introduced. This unmanned surface vehicle (USV) is designed as a twin-hull-type ship that is stable to external disturbances, and the remover part shreds the jellyfish using three rapidly rotating blades. An electrical control system for autonomous navigation is embedded in the JEROS. Currently, the robot

Manuscript received September 12, 2014; revised Febrary 11, 2015; accepted March 19, 2015. Recommended by Associate Editor Seul Jung under the direction of Editor Fuchun Sun. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (Grant No. NRF-2013R1A1A1A05011746). This research was also financially supported by the Robot industrial cluster construction program through the Ministry of Trade, Industry & Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT). The students are supported by Ministry of Land, Infrastructure and Transport (MoLIT) as U-City Master and Doctor Course Grant Program.

Hanguen Kim, Donghoon Kim, Hyungjin Kim, Jae-Uk Shin, and Hyun Myung are with Urban Robotics Laboratory (URL), KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Korea (e-mails: {sskhk05, dh8607, hjkim86, jacksju, hmyung}@kaist.ac.kr).
* Corresponding author.

system has been extended to a multi-agent robot system composed of three prototypes to enhance the efficiency of jellyfish removal.

To control the multi-agent robot system, various formation control approaches in an ocean environment were introduced [10–13]. In this paper, the leader-follower scheme [14] commonly employed on the ground is used, but the dynamic model of robot is difficult to be formulated in ocean environment [15]. To cope with this, we utilize the line-of-sight (LOS) guidance algorithm [15] to build and maintain the formation of multiple robots. In this guidance-based scheme, each follower robot follows not only its desired position for maintaining formation but also the speed and heading angle of the leader robot. However, when the follower robots are tracking in accordance with the leader's following route without considering the performance limitation of the robot, the formation cannot be well maintained, even if the formation control algorithm was applied to the robots. Because maintaining formation is crucial to the efficiency of the jellyfish removal operation, if the formation cannot be well maintained while the robots are moving, the operation area becomes irregular and, therefore, the removal operation entails performing repetitive tasks.

To solve this problem, some path planning algorithms have been introduced. Garrido *et al*. [16] proposed applying the Voronoi Fast Marching (VFM) method to path planning of mobile robots to control the leader-follower deformable formation. The VFM method uses the Voronoi extracted areas and potential vector fields to obtain a smooth and safe (collision avoidance) path. This method considers the deformable formation but cannot consider the performance limitation of the robot. Thus, it is not suitable for efficient jellyfish removal. Also, Lázaro *et al*. [17] proposed a path planning algorithm under the uncertainty for robot formations. They used the A* algorithm [18] on the risk map as defined by the probabilistic projection of the feature-based stochastic map into a grid-based representation of the environment. The proposed algorithm is suitable for exploration of the uncertain area; however, it cannot consider the form of formation and performance limitation of the robot. Likewise, other attempted path planning methods for solving the problem of formation control are difficult to be applied to a multi-agent jellyfish removal application [19–22].

In this paper, for maintaining formation of the multi-agent robot system, we extend the ARC (Angular-Rate-Constrained)-Theta* algorithm introduced in our previous study [23]. While ARC-Theta* only applies to a single robot, eARC-Theta* (extended ARC-Theta*) can be applied to multiple robots. Since the maximum turning radius and backward steering of JEROS are limited, and the small turning radius or backward input that exceed performance limitation of the robots can lead to a large formation error, the proposed algorithm generates a smooth path considering these constraints.

Finally, we conducted field tests at Bang-dong Reservoir in Daejeon, South Korea, to demonstrate the improvement in the performance of maintaining formation by using a path generated by the eARC-Theta* algorithm. The results of maintaining formation with the generated path using the eARC-Theta* algorithm is compared with those using conventional gridmap-based path planning algorithms such as A* [24] and any-angle path planner called Theta* [25].

The paper is organized as follows: in Section 2, the guidance-based leader-follower scheme and the extended ARC-Theta* algorithm are proposed; in Section 3, the design and implementation of JEROS and experimental results of field tests for maintaining formation are presented. Finally, in Section 4, we summarize this paper and discuss future works.

2. MULTI-AGENT NAVIGATION SYSTEM

Our multi-agent navigation system utilizes the guidance-based leader-follower scheme and the eARC-Theta* algorithm. The guidance-based leader-follower scheme calculates target heading angle and speed of follower robots for formation building and maintaining. The eARC-Theta* algorithm generates the waypoints by considering the performance limitation of the robots for the guidance-based leader-follower controller to operate normally.

2.1. Guidance-based leader-follower scheme

Currently, JEROS has been extended to a multi-agent robot system for efficient jellyfish removal. When the region where the jellyfish emerge is given, this system performs a jellyfish removal mission, including a repetitive area coverage operation like a cleaning robot [26]. The efficiency of the mission depends on the area coverage rate. The area coverage rate is related to the water volume passing through the jellyfish removal device per unit time, and it relies on the speed of JEROS and dimensions of the jellyfish remover device. However, increasing the dimensions is impractical, because enlarging the device would increase the drag force and drop the speed of JEROS. By performing the mission while maintaining formation, the robot system can overcome the limited area coverage rate and enhance its efficiency. For this reason, a leader-follower formation control method [27] has been employed and modified as appropriate to the JEROS system.

The leader-follower method has a simple architecture that consists of a leader robot and follower robots and one can specify the behavior of the group by directing the leader robot's motion. One robot is assigned to be the leader robot; other robots follow the leader robot while maintaining the specified formation. The difference be-

Table 1. Description of symbols used. Subscript i implies l for a leader robot and f for a follower robot.

Symbol	Description
(x_{los}, y_{los})	LOS point
ρ_d	Desired translation of a follower robot with respect to earth-fixed frame
ρ_{fw}	Target translation of a follower robot with respect to earth-fixed frame
Ψ_i	Heading angle of a robot with respect to earth-fixed frame
Ψ_i^*	Target heading angle of a robot with respect to earth-fixed frame
Ψ_d	Desired heading angle of a robot for formation with respect to earth-fixed frame
Ψ_w	Heading angle of a waypoint of a follower robot with respect to earth-fixed frame
Ψ_w^*	Target heading angle of a follower robot's target waypoint with respect to earth-fixed frame
$\mathbf{X}_i$	Position of a robot
$\mathbf{X}_w$	Position of a follower robot's target waypoint
P_{cm}	Central moment of a formation
P_f	The farthest robot position from a moment point of a formation
r	Angular rate of a robot
R_{los}	Radius of a circle from a follower robot for deriving an LOS point
R_t	Turning radius of a robot
u_i	Surge speed with respect to body-fixed frame
u_i^*	Target surge speed with respect to body-fixed frame
u_w	Target surge speed of a follower robot on a waypoint with respect to body-fixed frame
$V_{f\max}$	Maximum speed of a follower robot
(x_i, y_i)	Location of a robot with respect to earth-fixed frame

tween a leader robot and follower robots is only a software role in the formation. In addition, the role of a leader and a follower can be switched at any time. In our system, the LOS guidance algorithm is used for both following a path and maintaining formation. Because a dynamics model of a small USV system is hard to be analyzed exactly and is greatly influenced by external disturbances in an ocean environment [15], building and maintaining a formation are performed based on a kinematics model and the LOS guidance in this paper.

The LOS guidance algorithm usually generates a target heading angle [15] which is determined by computing an LOS vector which is formed by connecting the robot position to the intersecting point which is denoted as the LOS point (x_{los}, y_{los}) as can be seen in Fig. 1. Description of symbols used hereafter is listed in Table 1. The LOS point can be obtained by calculating an intersection point of a circle of radius R_{los} from the follower robot and a line connecting waypoints. The waypoint is a virtual target described by the desired position, speed, and heading angle; it is generated by a path planning algorithm based on the position and velocity of the leader robot for maintaining the formation. In other words, the LOS point is the control point between successive waypoints. The leader robot follows the reference path generated from the path planning algorithm using the LOS guidance algorithm and determines the motion of follower robots. Each follower robot follows a desired waypoint using the LOS guidance law-

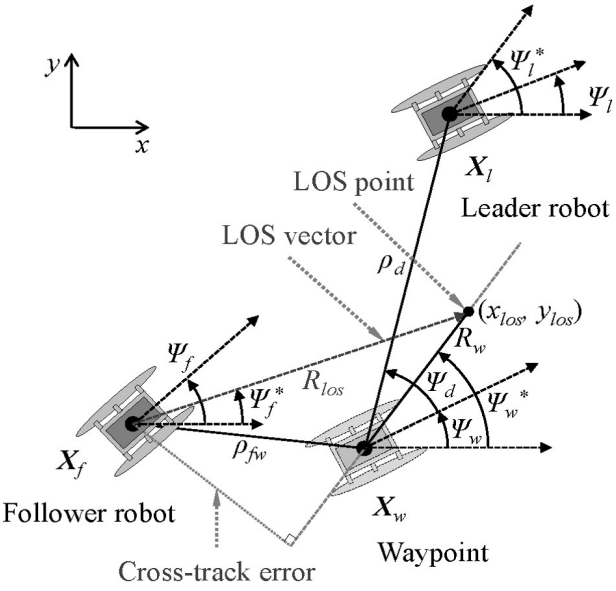


Fig. 1. Simple model of the leader-follower method and explanation of notation.

based guidance strategy. A simple model of the guidance-based leader-follower method is shown in Fig. 1. Since we assume that the USV motion is restricted to a horizontal plane, the position and heading angle of each robot can be described by $\mathbf{X}_i = (x_i, y_i, \Psi_i)^T$, where (x_i, y_i) and

Ψ_i denote location and heading angle with respect to the earth-fixed frame, respectively, and target heading angle is denoted by Ψ_i^*. Surge speed and target surge speed are denoted by u_i and u_i^* with respect to the body-fixed frame, respectively. The positions of the leader and follower robots and the waypoint are denoted by $\mathbf{X}_l$, $\mathbf{X}_f$, and $\mathbf{X}_w$, respectively. The waypoint is the target position of the follower robot for maintaining formation and target speed of the follower robot at this point is denoted by u_w. $\mathbf{X}_w$ is determined by the desired translation and heading angle as follows:

$$X_w = \begin{pmatrix} x_l - \rho_d \cos(\Psi_d + \Psi_l) \\ y_l - \rho_d \sin(\Psi_d + \Psi_l) \\ \Psi_l \end{pmatrix}, \quad (1)$$

where (x_l, y_l) and Ψ_l denote the location and heading angle of leader robot, respectively; ρ_d and Ψ_d denote the desired translation and heading angle for formation, respectively. The waypoint's heading angle, Ψ_w, and target heading angle, Ψ_w^*, should be same to those of leader robot. u_w is determined by the leader robot's speed, u_l, and a speed offset that is needed to make up speed difference between the leader and follower robots whose turning radii are difference, which allows the follower robot to maintain the specified formation while the robots make rotational movements. If the turning radius of the follower robot is larger than that of the leader robot, the follower robot has to be fast in proportion to the radius difference, $\rho_d \sin(\Psi_d)$, and the angular rate of the leader robot, $\dot{\Psi}_l$. Thus, u_w is formulated as follows:

$$u_w = u_l + \rho_d \sin(\Psi_d)\dot{\Psi}_l. \quad (2)$$

The leader robot asymptotically follows a generated path using the LOS guidance algorithm. The follower robot follows the virtual straight path from $\mathbf{X}_w$ in the direction of Ψ_w^* using the LOS guidance law as shown in Fig. 1 to maintain the prescribed formation and to follow the leader robot's motion.

The target heading angle is determined by the following equation:

$$\Psi_f^* = \tan^{-1}\left(\frac{y_{los} - y_f}{x_{los} - x_f}\right). \quad (3)$$

If the follower robot is located at a distance larger than R_{los} from the path and the LOS point cannot be determined, the follower robot pursuits the waypoint and the target heading angle is calculated as follows:

$$\Psi_f^* = \tan^{-1}\left(\frac{y_w - y_f}{x_w - x_f}\right). \quad (4)$$

In addition, when the follower robot passes far beyond the waypoint, the robot returns to the waypoint using (4) since the robot cannot move backward.

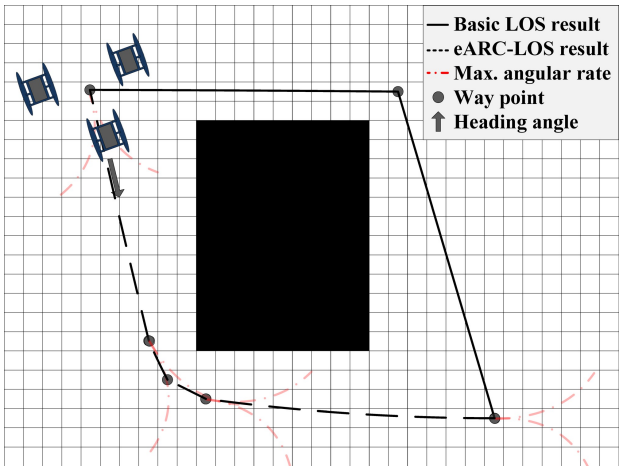


Fig. 2. Concept of the eARC-LOS. This concept makes it possible to maintain multi-agent robot formation while the robots are moving.

The target speed of the follower robot is determined by the following linear equation with respect to R_w:

$$u_f^* = \begin{cases} u_{\max}, \text{if } R_w \le K_u(u_0 - u_{\max}), \\ -\dfrac{1}{K_u}R_w + u_0, \text{if } K_u(u_0 - u_{\max}) < R_w \le K_u u_0, \\ 0, \text{otherwise}, \end{cases} \quad (5)$$

where R_w is the distance from $\mathbf{X}_w$ to the LOS point, u_{max} is the maximum speed, K_u is a constant value for the deceleration slope, and $u_0 = R_{los}/K_u + u_w$ is an offset value calculated to set u_f^* to u_w at $R_w = R_{los}$. When the follower robot pursues from a far distance, its target speed is set to maximum for building the formation more quickly. Around the waypoint, the target speed linearly decreases as R_w increases. When the follower robot is close to the waypoint, i.e., $R_w \approx R_{los}$, the target speed is set as u_w. When the follower robot goes past the waypoint, it slows down until it stops. The target speed, u_f^*, and the target heading angle, Ψ_f^*, are used to control the speed and direction of the robot.

Since the leader-follower formation control does not consider performance limitation of the follower robot in generating waypoints, a path might be generated in such a way that the follower robot cannot follow. For example, the waypoint behind the follower robot can cause a serious problem. Therefore, constraints such as the minimum turning radius need to be considered when the leader's path is generated.

2.2. Extended ARC-Theta* algorithm

Our gridmap-based approach suggested in [23] describes how to create paths in real time by considering vehicle performance on a special Euclidean group $SE(2)$ occupancy gridmap using the Theta*-based path planning

algorithm. Among the various path planning algorithms, the reason for choosing the gridmap-based path planning algorithm is that it generates a path in a short computation time and can be quickly applied to real-world applications. Our goal is to extend the work of [23] for maintaining formation of a multi-agent robot system. The proposed path planning algorithm creates the resultant paths on the assumption that the robots are configured with the formation.

The ARC-Theta* algorithm is based on an any-angle path planner called Theta*, which selects a parent node by checking the LOS in $SE(2)$. The parent node is the prior search node that is connected to a present search node. Theta* constantly checks the LOS when connecting between the present search node and its parent node. Therefore, unlike the A* algorithm, Theta* creates waypoints at major turning points and it is possible to connect nodes in any direction (angle) over any distance if there are no obstacles. One of the key concepts of ARC-Theta* is that it restricts the range of LOS and angular rate by accommodating for the maximum speed and the turning ability of the robot. Thus, the angular rate of each LOS is calculated when the node is expanded in ARC-Theta*. We extend this concept of ARC-Theta* to maintain formation of multi-agent robots while they are moving. Fig. 2 shows the concept of the eARC (extended ARC)-LOS. The eARC-LOS considers the maximum angular rate of multi-robots calculated from the formation pattern and performance specification of each robot. The angular rate, r, of a formation consisting of multiple robots is defined as the ratio of the follower robot's maximum speed, $V_{f\max}$, to the turning radius R_t, which is calculated from the distance between the central moment of the formation, P_{cm}, and the farthest robot position, P_f, from the moment point as follows:

$$\begin{aligned} r &= \frac{V_{f\max}}{R_t}, \\ R_t &= P_{\text{cm}} + P_f. \end{aligned} \tag{6}$$

The reason why the angular rate r is calculated by using $V_{f\max}$ is that the leader robot should be slower than the follower robot's maximum speed to maintain the formation while the multiple robots are moving. Also, the turning radius R_t should take into account the robot with the largest turning radius. The pseudo code of the eARC-LOS algorithm is shown in Algorithm 1. The eARC-LOS function checks whether the angular rate between node n and neighbor node n_{neighbor} is larger than the maximum angular rate $r_{\max}$. Then $r_{\max}$ is calculated by using (5). $d(n_{\text{neighbor}}, n)$ is the Euclidean distance between the node n and its neighbor node, and 'IsWalkable(n_{current})' checks the occupancy states between the node and the neighbor node according to the formation orientation and size. In this paper, we use a nonuniform grid representation to calculate the total cost *costsum* obtained by the sum of the nonuniform gridmap cost *mapcost* and also use Dubins curve algorithm [28] at the start and goal points to avoid problems explained in [23]. This algorithm is used to get a path considering performance of the vehicle under the motion primitives such as turning left or right given the angular rate of the vehicle and the assumption that the vehicle cannot reverse. Fig. 3 shows an example path of eARC-Theta* algorithm that uses eARC-LOS to consider the formation of the multi-agent robot while the robots are moving. The pseudo code of the eARC-Theta* algorithm is shown in Algorithm 2. In Algorithm 2, node n is defined as $n = [x, y]$ and nodes n_{start} and n_{goal} are the start point and goal point, respectively. As with the A* algorithm, *open* checks whether a node has been already visited. 'CreateDubinsCurves' function creates the Dubins curve at the start and goal points after finding a path with the eARC-LOS algorithm. As the final outcome, the eARC-Theta* algorithm creates a path that maintains formation of the multi-agent robot system.

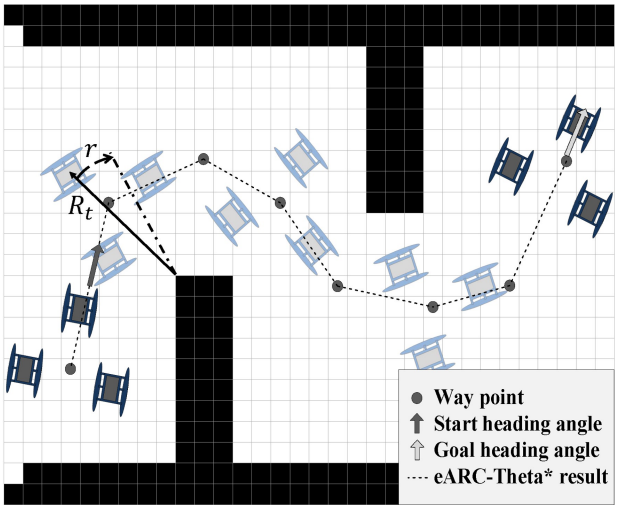


Fig. 3. Example of the eARC-Theta* algorithm result that considers the multi-agent robot formation while they are moving.

Algorithm 1 Extended ARC-LOS(n_{neighbor}, n)

```
if (n.θ − n_neighbor.θ) * d(n_neighbor, n)/V_fmax > r_max then
    return 'cannot connect n_neighbot'
end if
for each n_rmneighbor adjacent to n do
    n_current ← n_neighbor
    if !IsWalkable(n_rmcurrent) then
        return 'exist obstacles'
    else
        costsum+ = n_current.Mapcost
    end if
end for
n_neighbot.θ ← n.θ − n_neighbor.θ
return costsum and 'can connect'
```

Algorithm 2 Extended ARC-Theta*($n_{\text{start}}, n_{\text{goal}}$)

```
n ← n_start
n.parent ← n_start
while open ≠ ∅ do
  s ← open.Pop()
  for each n_neighbor do
    if eARC-LOS(n_neighbor, n) then
      n_neighbor.Parent ← n.parent
      open.Push(n_neighbor)
    end if
  end for
end while
if CreateDubinsCurve(n_start, n_goal) then
  return 'path found'
else
  return 'no path found'
end if
```

3. EXPERIMENTAL RESULTS

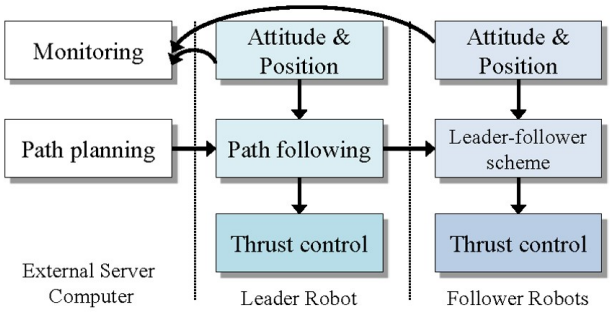
In this section, we describe experiments performed to evaluate eARC-Theta*. We begin by describing the JEROS system and its configurations. Then, we evaluate the eARC-Theta* algorithm with conventional gridmap-based path planning algorithms such as A* and Theta*. Following that, we examine the effectiveness of eARC-Theta* in more detail.

3.1. Multi-agent JEROS system

JEROS is a twin-hull-type USV that has dimensions of 1.5 m × 1.1 m × 0.8 m (length × width × height) and a mass of about 50 kg in air. Its sensor system for localization consists of a GPS (Global Positioning System) receiver and an IMU (Inertial Measurement Unit) that provides global position and attitude information, respectively. The actuator system of the JEROS is designed as a differential drive system using two thrusters installed at the rear of two parallel hulls. A laptop computer and a microprocessor are embedded for processing of proposed algorithms and for thruster control, respectively. The electrical parts of JEROS are listed in Table 2. JEROS has been extended to a multi-agent robot system composed of three JEROS prototypes, as shown in Fig. 4. One of the prototypes is assigned as a leader robot and others are assigned as follower robots for a leader-follower scheme. All robots and an external server computer are connected via a 3G/LTE network using TCP/IP communication, which provides unlimited communication range within the network. The robots share their status including position, attitude, and speed, and the external server computer transfers the information of jellyfish emergence to the leader robot. Then the leader robot takes the lead on the jellyfish removal mission while maintaining formation with follower robots; thus, the multi-agent robot system can easily be operated by controlling only the leader robot.

Table 2. Electrical parts of JEROS.

Device	Name	Manufacturer
GPS	OEMStar	Novatel
IMU	EBIMU-9DOF	E2BOX
Computer	Core i5 Laptop	Hansung
Microprocessor	ATmega128	Atmel
Thruster	Endura C2	Minkota



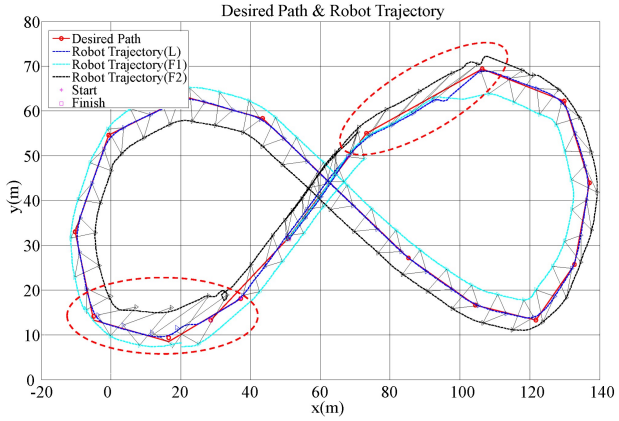
(a) JEROS system control scheme.

(b) Multi-agent JEROS system and its configuration.

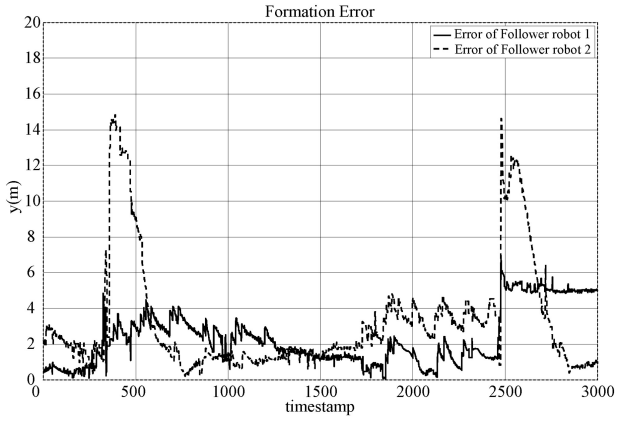
Fig. 4. Multi-agent JEROS system and its configuration.

3.2. Experimental results

To investigate the eARC-Theta* algorithm, field tests were conducted at Bang-dong Reservoir in Daejeon, South Korea. As mentioned in Section 2.1, the two follower robots follow the waypoints that were generated from information of the leader robot's tracked waypoints. Both lengths of the LOS vectors for path following guidance were set to 3 m. The desired translations and heading angles were set to $(\rho_d, \Psi_d) = (4, \pi/3)$ for follower robot 1 and $(\rho_d, \Psi_d = (4, -\pi/3)$ for follower robot 2 (unit: m, rad), which represented an equilateral triangle formation. The target speed of the leader robot, v_l^*, and the maximum speed of the follower robot, $v_{f\max}$, were set to 0.9 and 1.8 m/s, respectively. In the first experiment, we



(a) Experimental result of JEROS autonomous navigation with a manually generated figure-eight-shaped path.



(b) Euclidean distance error between the waypoint and the position of the follower robots.

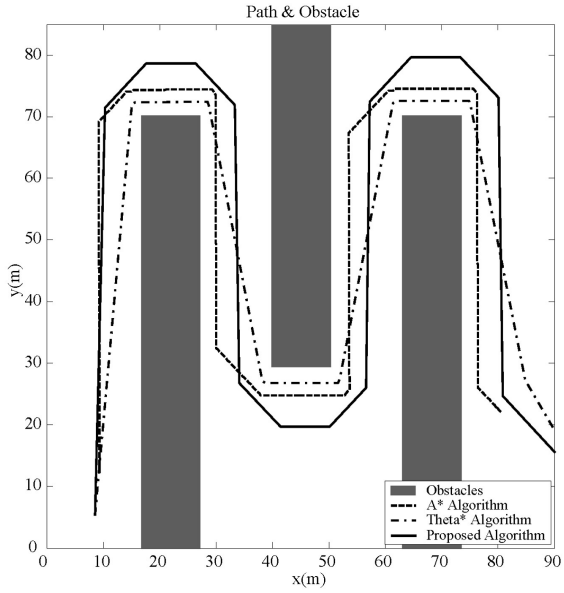
Fig. 5. Experimental results of a figure-eight-shaped path and its formation error.

generated a figure-eight-shaped path without a path planning algorithm to evaluate the feasibility of the leader-follower scheme. The size of gridmap is 500×500 pixels and the grid resolution is 0.18 m/pixel. The evaluation is performed by recording the Euclidean distance error between the waypoint and the position of the follower robot. Fig. 5(a) shows the experimental results of the autonomous navigation of the multi-agent JEROS for the manually generated figure-eight-shaped path with the formation of the follower robots overlaid. We found that the formation did not maintain well at the point where turning is commanded over turning ability of robots as indicated by dashed ellipses. In more detail, the formation error of the follower robots is over 10 m when the robots turn rapidly as indicated by the dashed ellipses in Fig. 5. The main reason for the large error in the dashed circle region is that follower robots take a long time to reconstruct a desired formation due to the performance limit of robots such as limited maximum velocity.

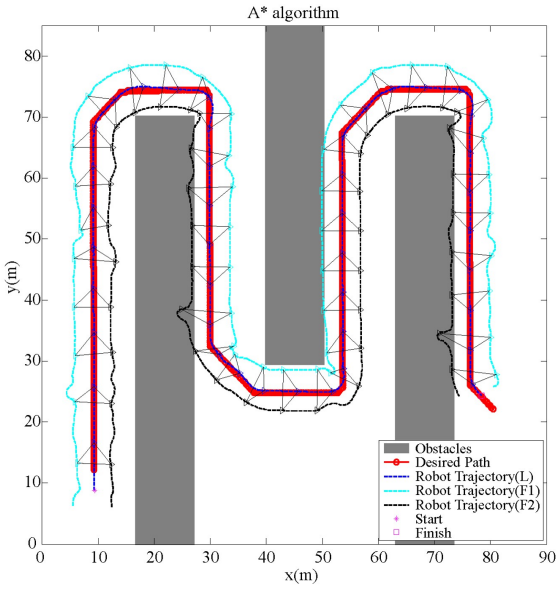
In the second experiment, we have tested the navigation using A*, Theta*, and eARC-Theta* algorithm. By comparing with A* and Theta*, we confirm how successfully eARC-Theta* maintains the formation of the JEROS system compared to the other gridmap-based path planning algorithms. A* and Theta* find a path with Euclidean distance-based cost function which is commonly used in path planning. And these algorithms can deal with collision avoidance of only a single robot. Although this problem can be resolved by somewhat complex modification of the algorithms, we have used original A* and Theta* algorithms for fair comparison. Assuming virtual obstacles were installed, we have created zig-zag-shaped paths to check how well formation of the JEROS system is maintained when the robots turn as shown in Fig. 6(a). The experimental conditions were the same as in the first experiment, and the experimental results and the formation error of the follower robots are shown in Figs. 6 and 7, respectively. In the case of A*, the step patterns resulting from the gridmap resolution caused frequent occurrences of large error, even when the leader robot was moving forward, as shown in Fig. 6(b). In the case of Theta*, the generated path is composed of the waypoints on the inflection points. Therefore, there were few errors if the LOS vector was constant when the robots were moving forward, but a sudden curvature resulted in large error, as shown in Fig. 6(c). In contrast, the eARC-Theta* maintained a stable formation of multi-agent JEROS system not only when the robots were moving forward but also when they were turning at the inflection points, as shown in Fig. 6(d). The formation error of the follower robots are shown in Fig. 7. In the case of A*, the formation error was over 2 m on a regular basis in all sections of the waypoints. Likewise, in case of Theta*, the formation error was over 2 m when the robot turned at the inflection points. The eARC-Theta* errors were less than 2 m in all sections of the waypoints. This result demonstrates how well eARC-Theta* maintains the formation of the multi-agent JEROS system even when GPS error is taken into account. The root-meansquared error (RMSE), maximum error, and mean error results are listed in Table 3. The performance of eARC-Theta* was consistently better than the other path planning algorithms. The eARC-Theta* algorithm was shown to create a path that enables multi-agent robots to better follow paths in an ocean environment.

4. CONCLUSION AND FUTURE WORK

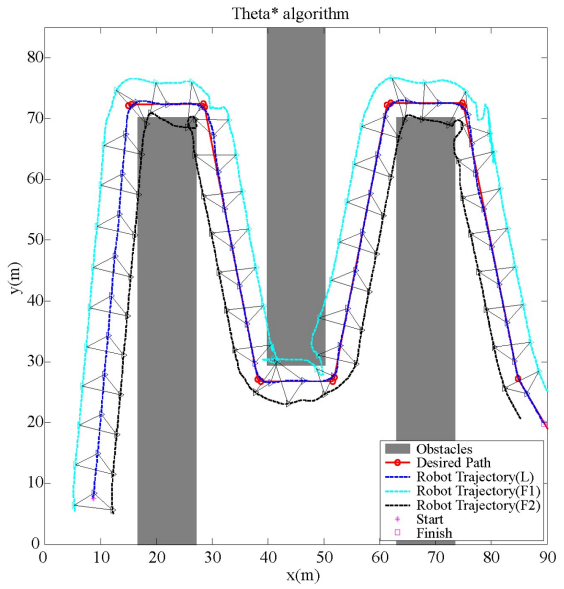
In this paper, to accomplish the autonomous navigation of the multi-agent robot system, a guidance-based leader-follower scheme was proposed to control their formation. We also proposed the extended angular-rate-constrained path planning algorithm (eARC-Theta*) for the multi-agent autonomous jellyfish removal robot system JEROS.



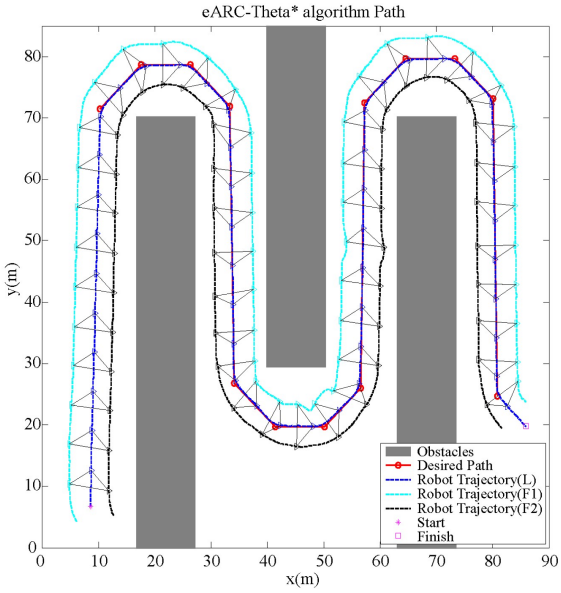
(a) Experimental paths from A*, Theta*, and eARC-Theta* algorithms (with virtual obstacles assumed to be installed).



(b) Result of A* algorithm, showing a large formation error owing to the zig-zag waypoints. Note that the desired path looks thick due to the densely generated waypoints.



(c) Result of Theta* algorithm, showing a stable formation except for the case of turning.



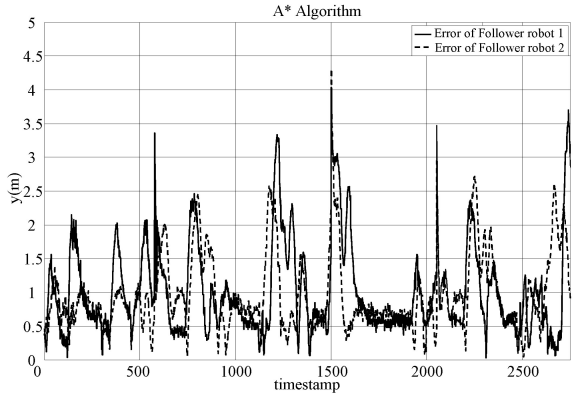
(d) Result of eARC-Theta* algorithm, showing a constantly stable formation while avoiding obstacles.

Fig. 6. Generated paths (a) and experimental results for (b) A*, (c) Theta*, and (d) eARC-Theta* algorithms.

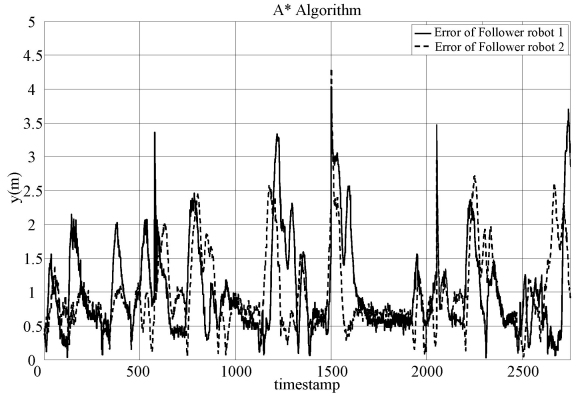
The eARC-Theta* algorithm was employed to plan a path considering the formation state and the vehicle's performance constraints. The performance of maintaining formation was demonstrated through field tests in a reservoir in South Korea. The results of the test using a path generated by the eARC-Theta* path planning algorithm showed a smooth and stable formation compared with the results using A* and Theta*. Future research will be focused on creating more advanced formation control algorithms and conducting more elaborate investigations on the efficiency of JEROS through various field tests.

Table 3. Formation error of the follower robots (RMSE, Maximum error, Mean error).

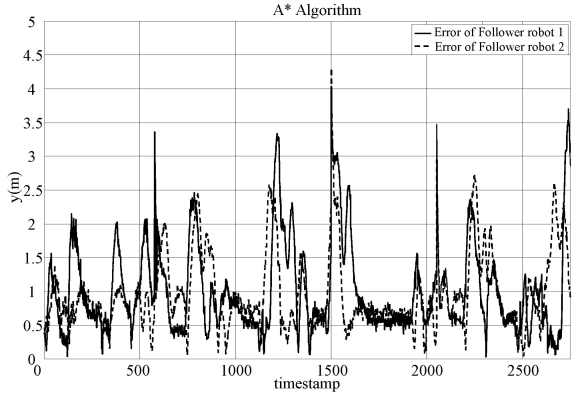
	Follower robot 1 (m)			Follower robot 2 (m)		
	RMSE	Max	Mean	RMSE	Max	Mean
Manual	2.34	6.84	7.90	3.27	14.61	21.17
A*	1.01	3.70	1.52	0.97	4.31	1.27
Theta*	1.01	4.84	1.51	1.00	4.52	1.41
Proposed	0.83	1.96	0.81	0.72	1.66	0.61



(a) Formation error for A*.



(b) Formation error for Theta*.



(c)) Formation error for eARC-Theta*.

Fig. 7. Formation errors for (a) A*, (b) Theta*, and (c) eARC-Theta* algorithms.

REFERENCES

[1] H.-S. Choi, "Scientists seek beneficial uses for jellyfish," *The Korea Herald*, http://www.koreaherald.com/view.php?ud=20120826000052, accessed 4 Dec. 2012.

[2] I.-O. Kim, H.-C. An, J.-K. Shin, and B.-J. Cha, "The development of basic structure of jellyfish separator system for a trawl net," *Journal of the Korean Society of Fisheries Technology* (in Korean), vol. 44, no. 2, pp. 99–111, 2008. [click]

[3] NFRDI, *Trends of overseas fisheries*, Technical Report, no. 2, 2005 (National Fisheries Research and Development Institute (NFRDI) of South Korea issued in Korean).

[4] F. Matsuura, N. Fujisawa, and S. Ishikawa, "Detection and removal of jellyfish using underwater image analysis," *J. Visualization*, vol. 10, no. 3, pp. 259–260, 2007. [click]

[5] J.-H. Lee, D.-S. Kim, W.-J. Lee, and S.-B. Lee, "System and method to prevent the impingement of marine organisms at the intake of power plants," Korean Patent 10-0558267-00-00, 2006.

[6] D. Kim, J.-U. Shin, H. Kim, D. Lee, S.-M. Lee, and H. Myung, "JEROS: Jellyfish removal robot sys- tem," *Proc. the Eighth Int'l Conf. Humanized System* (ICHS), pp. 336–338, 2012.

[7] D. Kim, J.-U. Shin, H. Kim, D. Lee, S.-M. Lee, and H. Myung, "Development of jellyfish removal robot system JEROS," *Proc. Int'l Conf. Ubiquitous Robots and Ambient Intell.* (URAI), pp. 599–600, 2012. [click]

[8] D. Kim, J.-U. Shin, H. Kim, D. Lee, S.-M. Lee, and H. Myung, "Experimental tests of autonomous jellyfish removal robot system JEROS," *Proc. Int'l Conf. Robot Intell. Technology* (*RiTA*), pp. 395–403, 2012. [click]

[9] D. Kim, J.-U. Shin, H. Kim, H. Kim, D. Lee, S.-M. Lee, and H. Myung, "Design and implementation of unmanned surface vehicle JEROS for jellyfish removal," *J. Korea Robot. Soc.*, vol. 8, no. 1, pp. 51– 57, 2013. [click]

[10] I. A. Ihle, J. Jouffroy, and T. I. Fossen, "Formation control of marine surface craft: A Lagrangian approach," *IEEE J. Oceanic Engineering*, vol. 31, no. 4, pp. 922–934, 2006. [click]

[11] F. Fahimi, "Sliding-mode formation control for underactuated surface vessels," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 617–622, 2007. [click]

[12] R. Cui, S. S. Ge, B. V. Ee How, and Y. S. Choo, "Leader-follower formation control of underactuated autonomous underwater vehicles," *Ocean Engineering*, vol. 37, no. 17, pp. 1491–1502, 2010. [click]

[13] M. Bibuli, G. Bruzzone, M. Caccia, A. Gasparri, A. Priolo, and E. Zereik, "Swarm-based path-following for cooperative unmanned surface vehicles," *Journal of Engineering for the Maritime Environment*, vol. 228, no. 2, pp. 192–207, 2014. [click]

[14] M. Breivik, V. E. Hovstein, and T. I. Fossen, "Ship formation control: A guided leader-follower approach," *Proc. IFAC World Congress*, pp. 16008– 16014, 2008. [click]

[15] T. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*, Marine Cybernetics, Trondheim, 2002.

[16] S. Garrido, L. Moreno, and P. U. Lima, "Robot formation motion planning using Fast Marching," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 675–683, 2011. [click]

[17] M. T. Lázaro, P. Urcola, L. Montano, and J. A. Castellanos, "Position tracking and path planning in uncertain maps for robot formations*," *Multi-vehicle Systems*, vol. 2, no. 1, pp. 7–12, 2012. [click]

[18] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. [click]

[19] F.-L. Lian, "Cooperative path planning of dynamical multi-agent systems using differential flatness approach," *International Journal of Control, Automation and Systems*, vol. 6, no. 3, pp. 401–412, 2008. [click]

[20] T. D. Barfoot, C. M. Clark, "Motion planning for formations of mobile robots," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 65–78, 2004. [click]

[21] A. Janchiv, D. Batsaikhan, B. Kim, W. G. Lee, S.-G. Lee, "Time-efficient and complete coverage path planning based on flow networks for multi-robots," *International Journal of Control, Automation and Systems*, vol. 11, no. 2, pp. 369–376, 2013. [click]

[22] A. N. Asl, M. B. Menhaj, A. Sajedin, "Control of leader-follower formation and path planning of mobile robots using Asexual Reproduction Optimization (ARO)," *Applied Soft Computing*, vol. 14, part c, pp. 563–576, 2014. [click]

[23] H. Kim, D. Kim, J.-U. Shin, H. Kim, and H. Myung, "Angular rate-constrained path planning algorithm for unmanned surface vehicles," *Ocean Engineering*, vol. 84, pp. 37–44, 2014. [click]

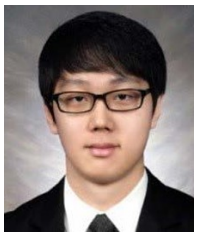
[24] H. Kim, B. Park, and H. Myung, "Curvature Path Planning with High Resolution Graph for Unmanned Surface Vehicle," *Proc. Int'l Conf. Robot Intell. Technology (RiTA)*, pp. 147–154, 2012. [click]

[25] A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," *Proc. the National Conf. Artif. Intell. (AAAI)*, pp. 1177–1183, 2007. http://www.aaai.org/Papers/AAAI/2007/AAAI07-187.pdf

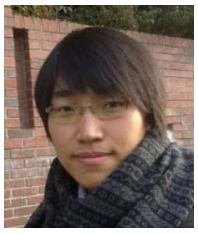
[26] H.-K. Lee, K. Choi, J. Park, and H. Myung, "Selfcalibration of gyro using monocular SLAM for an indoor mobile robot," *International Journal of Control, Automation and Systems*, vol. 10, no. 3, pp. 558–566, 2012. [click]

[27] Y. Wang, W. Yan, and W. Yan, "A leader-follower formation control strategy for AUVs based on line-of-sight guidance," *Proc. of IEEE International Conf. Mechatronics and Automation*, pp. 4863–4867, Aug. 2009. [click]

[28] S. M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.



Hanguen Kim received the B.S. and M.S. degrees from Kyung Hee University, Seoul, Korea in 2009 and 2011, respectively. He is currently working toward the Ph.D. degree at Urban Robotics Lab., KAIST (Korea Advanced Institute of Science and Technology), Daejeon, Korea. His research interests include mobile robot navigation, artificial intelligence, human robot interaction, and surface robot.



Donghoon Kim received the B.S. degree from University of Seoul, Seoul, Korea in 2009, and the M.S. degree from KAIST, Korea in 2011. He is currently working toward the Ph.D. degree at Urban Robotics Lab., KAIST, Daejeon, Korea. His research interests include computer vision and surface/underwater robot.

Hyungjin Kim received the B.S. degree from Kyung Hee University, Seoul, Korea in 2012, and the M.S. degree from KAIST, Korea in 2014. He is currently working toward the Ph.D. degree at Urban Robotics Lab., KAIST, Daejeon, Korea. His research interests include probabilistic robotics, robot navigation, and surface robot.

Jae-Uk Shin received the B.S. degree from Korea University of Technology and Education, Cheonan, Korea in 2010, and the M.S. degree from KAIST, Korea in 2014. He is currently working as a research engineer at Rastech, Inc., Daejeon, Korea. His research interests include robot design, robot control, and surface robot.

Hyun Myung received his Ph.D. degree in electrical engineering from KAIST, Daejeon, South Korea in 1998. He is currently an Associate Professor in the Dept. of Civil and Environmental Engineering, KAIST. He was a principal researcher in SAIT (Samsung Advanced Institute of Technology, Samsung Electronics Co., Ltd.), Yongin, Korea (2003.7-2008.2). He was a director in Emersys Corp. (2002.3-2003.6) and a senior researcher in ETRI (Electronics and Telecommunications Research Institute) (1998.9-2002.2), South Korea. His research interests include the areas of mobile robot navigation, autonomous underwater vehicle, autonomous surface vehicle, SLAM (Simultaneous Localization And Mapping), evolutionary computation, numerical and combinatorial optimization, and intelligent control based on soft computing techniques.