Path Planning for Multi-Agent Jellyfish Removal Robot System JEROS and Experimental Tests

Donghoon Kim, Hanguen Kim, Hyongjin Kim, Jae-Uk Shin, and Hyun Myung

Abstract Over the recent years, the increasing influence of climate change has given rise to an uncontrolled proliferation of jellyfish in marine habitats, which has visibly damaged many ecosystems, industries, and human health. To resolve this issue, our team developed a robotic system to successfully and efficiently remove jellyfishes, named JEROS (Jellyfish Elimination RObotic Swarm). The JEROS consists of multiple USVs (Unmanned Surface Vehicles) that freely move in a marine environment to scavenge for and eliminate jellyfishes. In this paper, we propose a constrained formation control algorithm that enhances the efficiency of jellyfish removal. Our formation control algorithm is designed in consideration of the characteristic features of JEROS. It is designed to effectively work with the simple leader-follower algorithm. The leader-follower formation control does not work well if a reference path of the leader is generated without considering a minimum turning radius. In order to overcome such a limitation, a new path planning method - angular rate-constrained path planning - is proposed in this paper. The performance of the jellyfish removal function was tested at Masan Bay in the Southern coast of South Korea and formation control tests were conducted at Bang-dong Reservoir in Daejeon, South Korea.

Key words: Jellyfish removal robot, Unamanned surface vehicle, Path planning, Multi-agent robot

1 Introduction

Recently, the proliferation of jellyfish has emerged as a serious environmental issue that has threatened marine ecosystems and caused an enormous damage to marine-

Donghoon Kim · Hanguen Kim · Hyongjin Kim · Jae-Uk Shin · Hyun Myung (✉)
URL (Urban Robotics Lab.), KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, South Korea, 305-701,
e-mail: {dh8607, sskhk05, hjkim86, jacksju, hmyung}@kaist.ac.kr

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 USD per year in 2009 [2]. In particular, the fishery industries, seaside power plants, and oceanic tourism enterprises have taken the most serious hit. 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 strong venom that can lead people to death. To cope with this problem, a number of studies pertaining to jellyfish removal have been actively performed. A previous study has developed a system consisting of two trawl boats equipped with jellyfish cutting nets [5, 13]. Utilizing large ships and many human operators, the system has shown high performance in jellyfish removal, but was limited in terms of its difficulty to operate 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 [11]. Yet another system utilized a bubble generator and a conveyor device [10]. 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, named JEROS (Jellyfish Elimination RObotic Swarm) consisting of a USV (Unmanned Surface Vehicle) part and a jellyfish remover part, was presented in [7]. The USV is designed based on a twin-hull-type ship that is stable to external disturbances, and the remover part shreds jellyfish using a rapidly rotating blade. An electrical control system for autonomous navigation is embedded in the prototype of JEROS. The design of the ship, navigation and image processing algorithms, and feasibility tests for the algorithms and jellyfish removal were introduced.

In this paper, the enhanced design of mechanical and electrical systems and the multi-agent robot system of JEROS are presented. The robot system is extended to a multi-agent robot system composed of three USVs to enhance the efficiency of jellyfish removal, and the leader-follower scheme [1] is employed and enhanced to control the formation of multiple robots. In the enhanced scheme, each follower robot

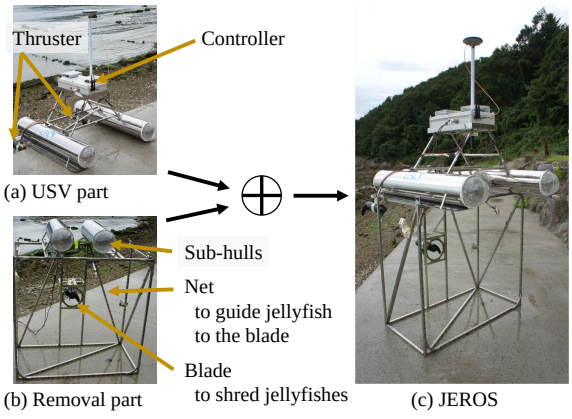


Fig. 1 JEROS consisting of a USV part and a remover part

follows not only its desired position for formation control, but also the speed and heading angle of the leader robot by utilizing the line-of-sight (LOS) guidance algorithm [4]. Additionally, for the autonomous navigation functionality considering formation control of the multi-agent robot system, a novel angular rate-constrained path planning algorithm based on the Theta* path planning algorithm [12] is introduced. Since the minimal turning radius and backward steering of JEROS are limited, the small turning radius or backward input to follower robots caused by sharp turn of leader robot can lead to a large formation error. The proposed algorithm generates a smooth path considering these constraints. Finally, we carried out field tests at Bang-dong Reservoir in Daejeon, South Korea to demonstrate the enhancement in the performance of formation control by using a path generated by the proposed path planning algorithm. The result of formation control with the generated path using the proposed path planning algorithm is compared with those using A* and Theta*. The performance of the jellyfish removal was also demonstrated through field tests in Masan Bay located in South Korea.

The paper is organized as follows: in Section 2, the design and implementation of JEROS, the formation control based on the leader-follower scheme, and the novel path planning algorithm based on Theta* are described; in Section 3, experimental results of field tests for formation control and jellyfish removal are presented. Finally, in Section 4, we summarize this paper and discuss future works.

2 Formation Control and Path Planning of JEROS

2.1 Design of JEROS

The JEROS is composed of two parts as outlined in Fig. 1. One is the USV part, and the other is the remover part. The USV part can be operated alone, independent of the attachment of the remover part. In the remover part, a funnel-shaped net guides jellyfish to be gathered near the blade. The total mass of the USV part of JEROS weighs about 45kg. Through some experiments at a fresh water tank, it is found that the maximum payload of USV is about 40kg for the robot to remain stable

Table 1 Components Used in Controller

Component	Model	Manufacturer
GPS	OEM Star	Novatel
IMU	EBIMU-9DOF	E2BOX
Global Network	3G Modem	Qualcomm
Local Network	EZBee-M100-EXT(Zigbee)	Chipsen
Computer	Core i7 SBC	Intel
Microprocessor	TMS320F2808	Texas Instrument
Thruster	Endura C2	Minnkota

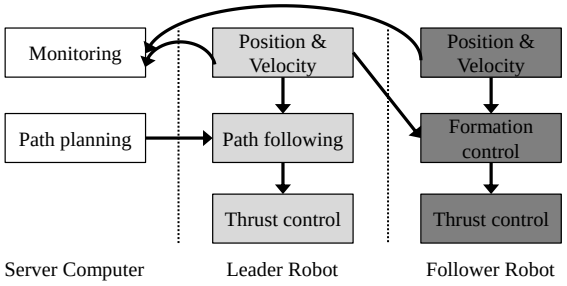


Fig. 2 Control scheme of JEROS

in sea-state 2. The sea-state is the condition index of sea with respect to wind and wave, and sea-state 2 is characterized as smooth wave. The payload is increased at the sea due to the larger specific gravity of seawater than that of fresh water. The remover part can float by itself by virtue of the sub-hulls. The USV is operated as a differential drive robot with two thrusters installed on the rear portion of the USV. The tested maximum speed is about 2m/s. The USV is designed as a twin-hull-type since it is more stable against waves than mono-hull-type and easy to increase the payload using large hull.

The components used to make the controller are listed in Table 1. The position and heading angle are measured by GPS and IMU. The high-level control algorithm such as path-planning and formation control is computed by the SBC (Single Board Computer), and the low-level control such as controlling the thrusts is performed by the DSP. The control scheme is illustrated in Fig. 2. For communications between the USVs and the server computer at a remote place, two kinds of networks are used. To monitor and operate manually, the 3rd Generation mobile network (3G) is used. For local communications between the USVs, the ZigBee wireless network is used.

2.2 Formation Control

In order to overcome the limited area coverage of a single robot system, we employ a multi-agent cooperation system. The application of a multi-agent robot system enhances the jellyfish removal capacity that is commensurate with the number of robot systems used.

For the formation control of the multi-agent robot system, the leader-follower method is employed due to its simplicity and low computation time. According to the intended scheme of the leader-follower method, the leader robot follows the reference path generated from the path planning algorithm using LOS guidance law, whereas the follower receives the location and velocity information of the leader robot and maintains the formation. The follower robot, for maintaining the formation, creates a waypoint based on the information of the leader robot, and is controlled to follow the waypoint. In order to properly perform formation control, it is imperative to accurately pursue the waypoint's location, heading angle, and the

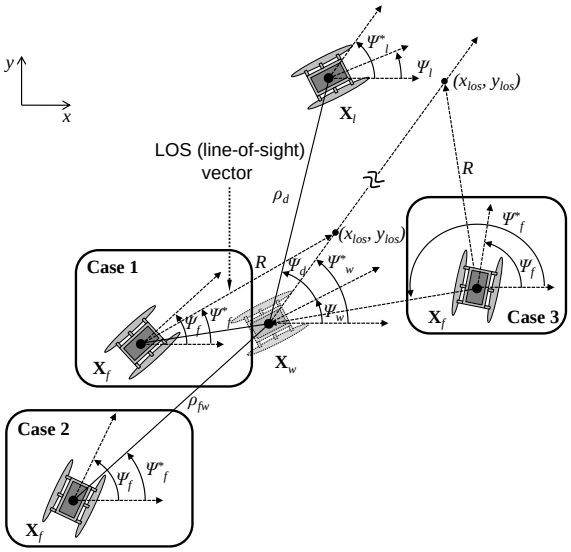


Fig. 3 Overview of the proposed formation control algorithm and explanation of notations

target speed at the location, but in marine environment faced with numerous disturbances, making the robot follow correct positions is not an easy task. In order to solve this problem, the follower robot is controlled as simply controlling a target its heading angle and a speed using the LOS guidance law in our system. The LOS guidance law provides target heading angle to asymptotically follow the position and target heading angle of the waypoint, which is case 1 as shown in Fig. 3. However, if the distance from follower robot to the waypoint is larger than R, the location of the waypoint is followed, which is the case 2 and 3 as shown in Fig. 3. The speed is controlled proportional to the distance from the follower robot to the waypoint.

The position of each robot is described by $\mathbf{X}_i = (x_i, y_i, \Psi_i)$ where x_i, y_i, and Ψ_i denote x, y coordinates, and heading angle of the robot, respectively. Ψ_i^* indicates target heading angle. The position of the leader and follower robots are denoted by $\mathbf{X}_l$, $\mathbf{X}_f$, and the waypoint which is the desired position of the follower robot is denoted by $\mathbf{X}_w$, as shown in Fig. 3. $\mathbf{X}_w$ is determined by the desired displacement and heading angle from $\mathbf{X}_l$, and is calculated as follows:

$$\mathbf{X}_w = \begin{pmatrix} x_l - \rho_d \cos(\Psi_d + \Psi_w) \\ y_l - \rho_d \sin(\Psi_d + \Psi_w) \\ \Psi_l \end{pmatrix}, \tag{1}$$

where ρ_d and Ψ_d denote the desired displacement and heading angle. Ψ_w and Ψ_w^* should be Ψ_l and Ψ_l^*, respectively. The desired speed of a follower robot at the location of the waypoint, v_w, is determined with consideration of the yaw rate of the leader robot, $\dot{\Psi}_l$, and the displacement of the follower robot as follows:

$$v_w = v_l + \rho_d \sin(\Psi_d)\,\dot{\Psi}_l. \tag{2}$$

The target heading angle of the follower robot, Ψ_f^*, is calculated as follows:

$$\Psi_f^* = \begin{cases} \tan^{-1}\left(\dfrac{y_{los}-y_f}{x_{los}-x_f}\right) & 0 \le \rho_c < R + k_\alpha v_w \,(\text{case 1}) \\ \tan^{-1}\left(\dfrac{y_w-y_f}{x_w-x_f}\right) & \text{otherwise}\,(\text{case 2,3}) \end{cases}, \tag{3}$$

where k_α is a constant gain. ρ_c is the distance between $\mathbf{X}_w$ and the LOS point, (x_{los}, y_{los}), in the direction of Ψ_w^*. In case 1 as shown in Fig. 3, the follower robot follows the waypoint and the leader robot's target heading angle by tracking a virtual path asymptotically using the LOS guidance law. The virtual path means a straight line from $\mathbf{X}_w$ in direction of Ψ_w^*. The LOS vector is a vector from $\mathbf{X}_f$ to the LOS point and its length is constant, R. The LOS point is a target point to track the virtual path. The case 2 and 3 indicate the large error state of the follower robot in the conditions of $\rho_c < 0$ and $\rho_c > R + k_\alpha v_w$, respectively. In these cases, Ψ_f^* is the angle from the follower robot to the waypoint and is calculated as shown in the second row of (3).

The target speed of the follower robot, v_f^*, is described as follows:

$$v_f^* = \begin{cases} v_{f,\max} - \dfrac{\left(v_{f,\max} - v_w\right)\rho_c}{R} & 0 \le \rho_c < R \\ v_w - \dfrac{v_w\left(\rho_c - R\right)}{k_\alpha v_w} & R \le \rho_c < R + k_\alpha v_w \\ \dfrac{v_{f,\max}\left(\rho_c - R + k_\alpha v_w\right)}{k_\alpha v_w} & R + k_\alpha v_w \le \rho_c \\ & \text{and}\, \rho_c < R + 2k_\alpha v_w \\ v_{f,\max} & \text{otherwise} \end{cases}, \tag{4}$$

where $v_f^* \in \left[0, v_{f,\max}\right)$ and $v_{f,\max}$ is the maximum speed of the follower robot. Equation (4) indicates a speed profile for the follower robot's waypoint tracking, which is designed as a linear interpolation with four inflection points with respect to ρ_c. The points are $\left(0, v_{f,\max}\right)$, (R, v_w), $(R + k_\alpha v_w, 0)$, $\left(R + 2k_\alpha v_w, v_{f,\max}\right)$. When the follower robot reaches near the waypoint, i.e., $\rho_c \simeq R$, v_f^* converges to v_w. If the robot reaches near the point of $\rho_c = R + k_\alpha v_w$ beyond the waypoint, v_f^* converges to 0. If $\rho_c > R + k_\alpha v_w$, the follower goes behind itself toward X_w with calculated speed. Otherwise, the follower heads for $\mathbf{X}_w$ directly with maximum speed.

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

2.3 Path Planning and Following

Based on [6, 8], we propose a new approach, named Angular Rate-Constrained Theta* (ARC-Theta*), to create paths with considerations of the formation state and the vehicle's performance. The proposed algorithm in this study is based on Theta*, which is similar to A* [12]. The basic Theta* selects a parent node by checking the LOS (Line of Sight). When connecting between the current search node s and its parent node $s.parent$, Theta* always checks the LOS, enabling it to search its neighbors $s_{neighbor}$ in any direction within a specified distance. One of the key concepts used in the proposed algorithm is to restrict the range of LOS and the angular rate by accommodating for the turning ability of the USV formation. Thus, the angular rate of each LOS is calculated when the node is expanded in Theta*. The angular rate r is defined as the ratio of the leader's speed V to the turning radius T_{total}, that is calculated from the leader's turning radius T_{leader} and the distance ρ_{lf} between the leader and the follower for a USV formation as follows:

$$\begin{aligned} r &= \frac{V}{T_{total}} \\ T_{total} &= T_{leader} + \rho_{lf}. \end{aligned} \tag{4}$$

Fig. 4 shows an example of the modified LOS, named as Angular Rate-Constrained LOS (ARC-LOS) that reflects the constraints. The ARC-LOS function always checks whether or not the current angular rate is greater than the maximum angular rate. If the current angular rate is greater than the maximum angular rate, the ARC-LOS function returns a false state. A function, *IsWalakble* checks the occupancy states around the current search node s on the weighted occupancy grid map. The occupancy states are calculated according to the vehicle orientation and size. If LOS is ensured, the ARC-LOS function returns the average occupancy cost according to the distance. Even so, the method cannot satisfy the arrival heading angle at the goal point because the ARC-LOS algorithm only verifies LOS between current

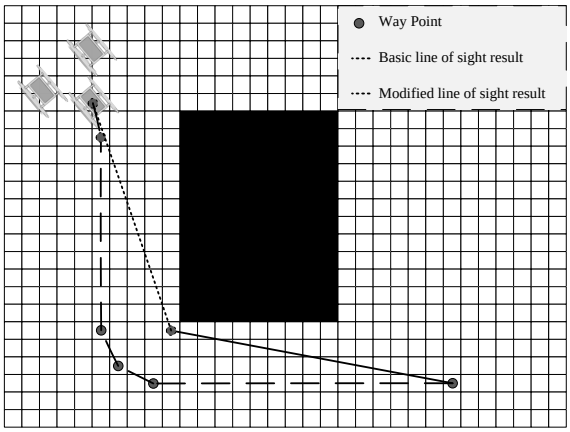


Fig. 4 Example of angular rate-constrained LOS result

Algorithm 1 ARC-Theta*(s_{start}, s_{goal})

```
1: s_start.Parent ← s_start
2: while open ≠ ∅ do
3:     s ← open.Pop()
4:     for each s_neighbor do
5:         if AngularRateConstrained LOS(s_neighbor, s) then
6:             s_neighbor.parent← parent
7:             open.Push(s_neighbor)
8:         end if
9:     end for
10: end while
11: if CreateDubinsCurves(s_start, s_goal) then
12:     return path found
13: else
14:     return no path found
15: end if
```

and previous nodes (explored node and its parent node). To mitigate this problem, Dubin's curve algorithm is applied at the start point and the goal point. Dubin's curve algorithm is used to get an optimal path under the rule that the vehicle turns left or right when the angular rate of the vehicle is given, with the assumption that the vehicle cannot reverse [9]. To apply Dubin's curve algorithm at the start and goal points, at first, a path from the start point to the goal point is created by Theta* with the ARC-LOS function. Then additional way-points are calculated using the Dubin's curve algorithm. This is possible because Theta* creates way-points using LOS so that the LOS between any two contiguous way-points is ensured. Thus, the influence of obstacles on the path can be avoided by creating Dubin's curve except in cases where the start angle or goal angle do not satisfy the LOS condition. If obstacles exist on the path generated with maximum angular rate at the start and goal points, the problem can be solved by decreasing the angular rate and re-calculating Dubin's curve. The pseudo code of the proposed path planning algorithm is shown in Algorithm 1. A function, *CreateDubinsCurves* generates the Dubin's curve at the start point and goal point after applying Theta* with the ARC-LOS function.

The navigation system calculates desired heading angles to follow given paths using the LOS guidance algorithm. The LOS guidance algorithm computes the LOS vector to calculate a control input to steer the vehicle. The LOS vector is formed by connecting the robot position to an intersecting point on the path at a distance of the tracking radius ahead of the robot.

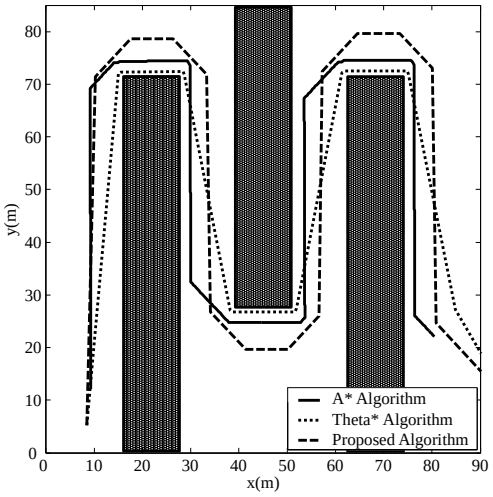


Fig. 5 Obstacle map denoted by dotted pattern and paths from A*, Theta*, and the proposed algorithms

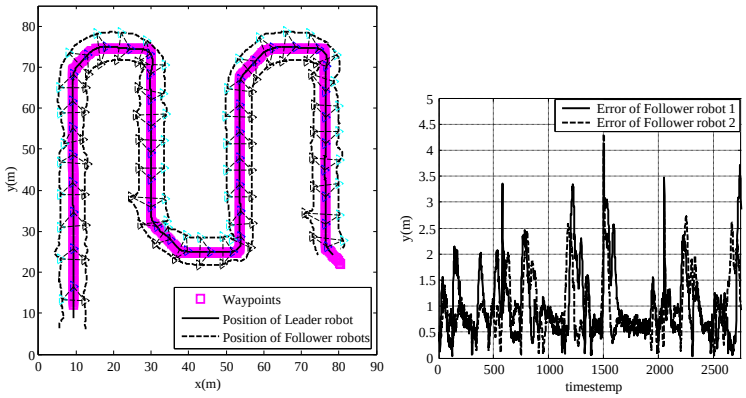
3 Experiments

3.1 Formation Control Tests

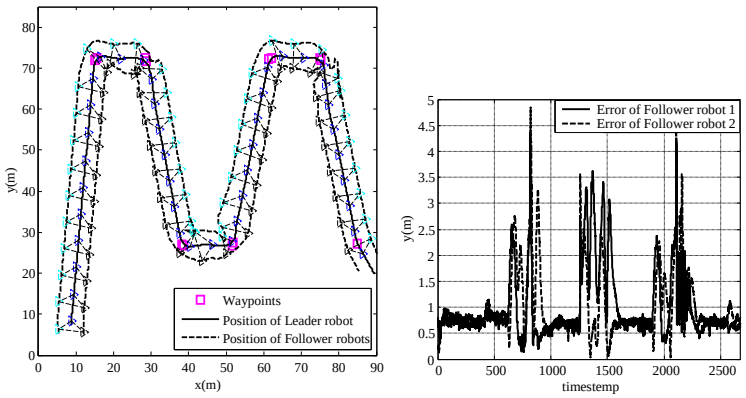
Field tests were conducted at Bang-dong Reservoir, Daejeon in South Korea to evaluate the feasibility of the formation control. Our multi-agent system consists of three JEROS prototypes (1 leader robot, 2 follower robots). Each robot is equipped with a GPS receiver (Novatel OEM-Star), which has 1.5m accuracy, and an IMU (EBIMU), which measures the robot's position, heading angle, and velocity, for conducting localization. The leader robot followed the paths of A*, Theta*, and the proposed path, while the two follower robots generated waypoints from the information received from the leader robot and followed those paths. Both lengths of LoS vector for path following guidance and formation control were set to 3m. The desired displacements and heading angles were set to $(\rho_{d1}, \Psi_{d1}) = (4, \pi/3)$ and $(\rho_{d2}, \Psi_{d2}) = (4, -\pi/3)$, respectively, 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 0.9m/s and 1.8m/s, respectively.

Fig. 5 shows the generated paths from A*, Theta*, and the proposed algorithm. To create the Zig-Zag paths in a specific area, we have assumed that the virtual obstacles were installed. In Fig. 6, the results indicate that A* and Theta* generate rapidly changing curves, in which some waypoints were generated behind the follower. However, it is clear that the proposed path, by virtue of its formation control function, successfully incorporated the restrictions in the turning radius, hence the previously observed problems did not occur.

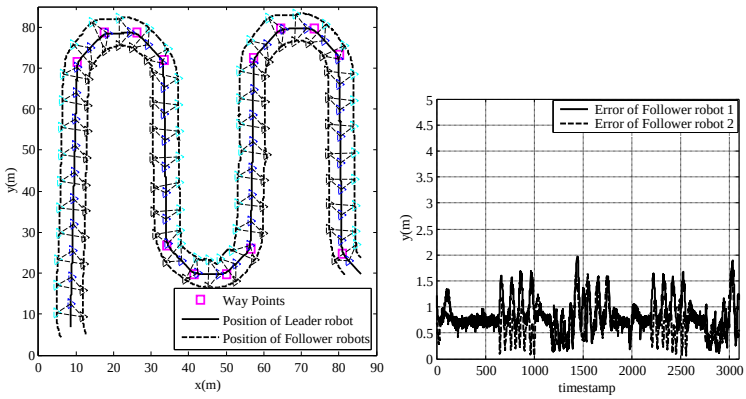
The position error between the waypoint and the two follower robots' is shown in right side of Fig. 6. In the case of A*, the zig-zag patterns shown on the straight path due to map resolution caused regular occurrence of large errors, even when the



(a) A* algorithm



(b) Theta* algorithm



(c) Proposed algorithm

Fig. 6 (a), (b), and (c) are resulting paths and following error of A*, Theta*, and proposed algorithm. The solid line and the dashed line denote the leader's path and the followers' path, repectively. The squares denote waypoints of each path.

Table 2 RMSE of the follower robots' position

	Follower robot 1 (m)	Follower robot 2 (m)
A*	1.01	0.97
Theta*	1.01	1.00
Proposed	0.83	0.72

leader was moving forward. In the case of Theta*, the generated path is composed of the waypoints on the inflection points. There were few errors when the robot was moving forward, but the event of a sudden curve resulted in a high error. The proposed path maintained a stable formation in comparison to A* and Theta*, and there was little error. The RMSE (Root-mean-squared error) results are listed in Table 2.

3.2 Jellyfish Removal Tests

The performance of the jellyfish removal function was tested at Masan Bay located in southern coast of South Korea. When the robot moved, the jellyfish swimming in the periphery of the robot's path were guided to the shredding blade by a net installed on the remover part. Then, the jellyfishes were cut into small pieces as shown in Fig. 7. In this experiment, it was verified that JEROS can remove approximately 36 *Aurelia aurita* jellyfish per 1 minute (approximately 3.6kg) when the speed of JEROS is 0.5 m/s and the entrance area of the remover part 1.44m^2. Fig. 7 visually shows the shredded remnants of many jellyfishes caught in the cutting blades.

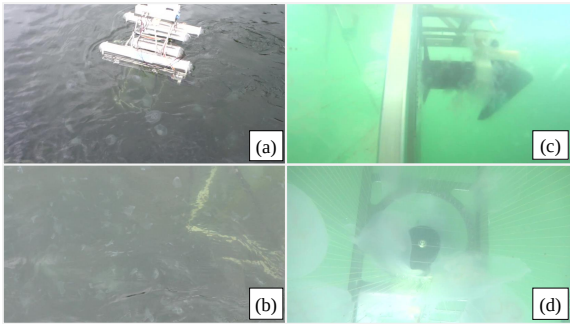


Fig. 7 Jellyfish removal test: (a) JEROS on the water (b) image of small pieces of shredded jellyfishes (c) image of small pieces of shredded jellyfishes in taken underwater (d) the jellyfishes guided to the blade

4 CONCLUSIONS

In this paper, we presented the leader-follower scheme-based formation control and a novel path planning algorithm based on Theta* for the multi-agent autonomous jellyfish removal robot system, JEROS. In addition, field tests were performed in order to assess the performance of formation control and jellyfish removal functionality. To enhance the performance of jellyfish removal, the robot system was modified compared to the previous version with respect to its dimensions, thrust force, and wireless communication method; and it was extended to a multi-agent robot system composed of three prototypes of JEROS. To accomplish the autonomous navigation of the multi-agent robot system, a leader-follower scheme was employed to control their formation. A novel path planning algorithm based on Theta* was employed to plan a path considering the formation state and the vehicle's constraints. The performance of the formation control was demonstrated through field tests in a reservoir in South Korea. The result of the test using a path generated by the proposed path planning algorithm showed smooth and stable formation control compared with the results using A* and Theta*. Additionally, the performance of jellyfish removal was estimated to be about 3.6kg per minute, on average, through field tests at Masan Bay located in South Korea. Future research will be focused on creating more advanced formation control algorithms and conducting more elaborate investigations of the efficiency of JEROS through various field tests.

Acknowledgements This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2013R1A1A1A05011746). Mr. H. Kim and Mr. H. Kim are supported by Korea Ministry of Land, Transport and Maritime Affairs (MLTM) as U-City Master and Doctor Course Grant Program.

References

1. Breivik M, Hovstein VE, Fossen TI (2008) Ship formation control: A guided leader-follower approach. IFAC World Congress
2. Choi H-S (2012) Scientists seek beneficial uses for jellyfish. The Korea Herald. http://www.koreaherald.com/view.php?ud=20120826000052. Cited 17 Jun 2014
3. Dunbabin M, Lang B, Wood B (2008) Vision-based docking using an autonomous surface vehicle. IEEE Int'l Conf Robot and Automation (ICRA)
4. Fossen T (2002) Marine control systems: Guidance, navigation and control of ships, rigs and underwater vehicles. Marine Cybernetics
5. Kim I-O, An H-C, Shin J-K, Cha B-J (2008) The development of basic structure of jellyfish separator system for a trawl net. J Korean Soc Fish Technol 44(2):99–111 (in Korean with English abstract)
6. Kim H, Lee T, Chung H, Son N, Myung H (2012) Any-angle path planning with limit-cycle circle set for marine surface vehicle. IEEE Int'l Conf Robot and Automation (ICRA)
7. Kim D, Shin J-U, Kim H, Kim H, Lee D, Lee S-M, Myung H (2013) Design and Implementation of Unmanned Surface Vehicle JEROS for Jellyfish Removal. J Korea Robot Soc 8(1):51–57

8. Kim H, Kim D, Shin J-U, Kim H, Myung H (2014) Angular rate-constrained path planning algorithm for unmanned surface vehicles. Ocean Eng 84:37–44
9. LaValle SM (2006) Planning algorithms. Cambridge university press
10. Lee J-H, Kim D-S, Lee W-J, Lee S-B (2006) System and method to prevent the impingement of marine organisms at the intake of power plants. Korean Patent 10-0558267-00-00
11. Matsuura F, Fujisawa N, Ishikawa S (2007) Detection and removal of jellyfish using underwater image analysis. J Visualization 10(3):259–260
12. Nash A, Daniel K, Koenig S, Felner A (2007) Theta*: Any-angle path planning on grids. National Conf Artif Intell (AAAI)
13. NFRDI (2005) Trends of overseas fisheries. Technical Report 2 (National Fisheries Research and Development Institute (NFRDI) of South Korea issued in Korean)