ENGINEERING JOURNAL

Article

Real Time Underwater Obstacle Avoidance and Path Re-planning Using Simulated Multi-beam Forward Looking Sonar Images for Autonomous Surface Vehicle

Thanapong Phanthong

Program in Physics, Faculty of Science and Technology, Songkhla Rajabhat University, Songkhla 90000, Thailand
E-mail: thanapong.phanthong@gmail.com

Abstract. This paper describes underwater obstacle avoidance and path re-planning techniques for autonomous surface vehicle (ASV) based on simulated multi-beam forward looking sonar images. The sonar image is first simulated and then a circular obstacle is defined and created in the field of view of the sonar. In this study, the robust real-time path re-planning algorithm based on an A* algorithm is developed. Our real-time path re-planning algorithm has been tested to regenerate the optimal path for several updated frames with a proper update frequency between the start point and the goal point both in static and dynamical environments. The performance of proposed method is verified through simulations, and tank experiments using an actual ASV. While the simulation results are successful, the vehicle model can avoid both single obstacle, multiple obstacles and moving obstacle with the optimal trajectory. For tank experiments, the proposed method for underwater obstacle avoidance system is implemented with the ASV test platform. The vehicle is controlled in real-time and moderately succeeds in its avoidance against the obstacle simulated in the field of view of the sonar together with the proposed position stochastic estimation of the vehicle.

Keywords: underwater obstacle avoidance, real-time path re-planning, A* algorithm, sonar image, ASV.

ENGINEERING JOURNAL Volume 19 Issue 1
Received 5 June 2013
Accepted 1 July 2014
Published 30 January 2015
Online at http://www.engj.org/
DOI:10.4186/ej.2015.19.1.107

DOI:10.4186/ej.2015.19.1.107

1. Introduction

In the field of underwater path planning and obstacle avoidance strategies, several literatures describing applications in this field that which many of the approaches are developed.

In [1], the dynamic potential method was used to solve the bottom navigation, avoidance and following problems of an under-actuated autonomous underwater vehicle (AUV). In [2], the application of a new nonlinear optimal control strategy based on the numerical approximation of solutions of the Hamilton–Jacobi–Bellman (HJB) equation for the station-keeping control of an underwater robotic vehicle in the horizontal plane has been presented. As similar to previous approach, in [3], they have studied the problem of steering a vehicle from its initial position in a 2D, time-varying, ocean flow field to a desired target position in minimum time, and focus on the case which the magnitude of the flow field exceeds the speed of the vehicle. In order to obtain trajectories, one solves the HJB equation for associated optimal feedback control law. In [4], simulated navigations of an AUV achieved by the minimum time guidance within undersea areas of current disturbances were presented. In this study, the concept of quasi-optimal navigation was presented and implemented by on-site updates of optimal guidance followed by the heading tracking control. In [5], they have presented numerical approaches for the problem of path planning and collision avoidance. The optimal control problem using control vector parameterization and single shooting methods have been solved. This method has solved the problem for a simple specific model of an underwater vehicle. In [6], a real-time optimal motion planning and an approximate analytical solution for the optimal control problem of a symmetric astable AUV with symmetric thruster configuration have been explained. The time-optimal path planning of autonomous vehicles in the presence of obstacles based on the dynamic programming (DP) approach has been undertaken in [7]. The minimum-time paths through a strong current region of position-dependent vector velocity (Zermelo's problem) of a surface vehicle have been presented in [8]. Alvarez et al. [9] propose a genetic algorithm (GA) for path planning of an AUV in an ocean environment characterized by strong currents and enhanced space-time variability. Their goal is to find a safe path that takes the vehicle from its starting location to a mission-specified destination, minimizing the energy cost. Chang et al. [10] have also proposed a framework of obstacle avoidance based on GA for AUVs to avoid moving obstacles that based on the real-time information of forward looking sonar (FLS). In [11], they describe a heuristic search technique based on the fuzzy relation products carrying out path planning and collision avoidance for AUVs. Tan et al. [12] present an integrated approach to the design of an obstacle avoidance system for AUVs, a type of incremental stochastic algorithm known as rapidly-exploring random tree (RRT) is merged with a maneuver automaton (MA) representation to form the motion planner. A fastest and efficient path planning method that is suitable for real-time missions based on a visibility graphs search is an A* algorithm [13]. Although the obtained path is near-optimal, however, it can search a suboptimal path extremely fast and most efficient with high performance. Therefore, in the field of underwater path planning, A* algorithm can be adopted especially in real-time missions.

With the recent underwater technologies of reliable, most obstacle avoidance platform used a high solution multi-beam FLS, particularly in [14] that describes a framework for segmentation of sonar images, tracking of underwater objects. This framework is still applied to the design of obstacle avoidance and path planning system for underwater vehicles based on a multi-beam FLS. Although their obtained paths on real sonar images were very smooth, could handle complex and changing workspaces. However, the path planning of this research was still performed in the vehicle frame and not in the world frame. Moreover, sonar servoing, real-time motion estimation and vehicle localization for actual vehicle of this research were not studied.

In this paper, we propose a two-dimensional underwater obstacle avoidance and path re-planning techniques for actual autonomous surface vehicle (ASV) based on simulated multi-beam FLS images, and the robust real-time path re-planning algorithm based on an A* algorithm is developed. The performance of proposed method is verified through simulations and tank experiments. For simulations, the vehicle model can avoid both single obstacle, multiple obstacles and moving obstacle with the optimal trajectory that are performed both in the vehicle frame and the world frame. To verify the effectiveness of the proposed method, the condition of environment with local minimum has also been tested. For tank experiments, an actual vehicle is automatically controlled in real-time and adequately succeeds in its avoidance against the static and moving obstacle simulated in the field of view of the sonar together with the proposed position stochastic estimation of the vehicle.

2. Target Application

Our specific target application of this research is to develop the proposed real-time underwater obstacle avoidance and path re-planning system that will be carried out with actual ASV and actual multi-beam FLS at the sea, while sonar image noises owing to sea environment has to be considered. Meanwhile, the position of the ASV will be measured by actual GPS receiver, therefore the position accuracy threshold of GPS must be also considered. We need to develop the simulation programs based on these applicabilities, and they must be easily adapted to the real-time programs and supports with actual instruments i.e. FLS and GPS.

3. Path Re-planning Algorithm

A famous and efficient of path planning algorithms is an A* algorithm. It is an optimization algorithm with modified best-first-search (BFS) strategy which uses heuristic cost estimation [13]. In our proposed method, we used the pixel-based representation of the obstacle and an optimal circular fitting algorithm for a given obstacle. Each obstacle in the sonar image frame is a constraint that the path generator must not cross the obstacle while minimizing the distance to goal. We have represented the obstacle as a circle in the field of view of the sonar, it is called the "sonar view" in this paper. Our optimal circular matching algorithm is applied to accomplish the represented obstacles. The start and goal pixels can be then assigned at any point within the sonar view. Begin at the start pixel, an A* algorithm chooses one of the adjacent pixels surrounding start pixel excluding obstacle pixels and previously moved ones. Anyhow, which pixel does it select, the answer is the one with the lowest *F(n)* cost. The key to determining which pixels to use when figuring out the path is the following equation [15]:

$$F(n) = G(n) + H(n). \tag{1}$$

G(n) is the cost to move from the starting pixel to a given pixel*(n)* within the sonar view. We have assigned *G(n)* cost of 1.0 (pixel) to each horizontal or vertical moved, and this cost of 1.4 (pixel) for a diagonal move, in the sonar view. *H(n)* is the estimated cost to move from that given pixel*(n)* to the goal pixel in the sonar view, *H(n)* cost is often referred to as the heuristic. Our path is generated by repeatedly going through the *open list* [15] and selecting the pixel with the lowest *F(n)* cost. In our assignment, each angle of the movement cost is 45 degrees that is called the *4-geometry* configuration [16]. For real-time path re-planning with the *4-geometry*, the running time for computation is so fast, even though the path is only suboptimal. However, it can be acceptable in our system. In the near future, we will develop our system to the *8-geometry* configuration [16] that each node has 24 related neighbors. However, we have to consider a good compromise between performance and computing time.

The main aim of development of our path re-planning algorithm is to particularly perform with the moving obstacle. The briefly flow chart of our algorithm shows as Fig. 1. Before discussing this algorithm, the concept of transformations between the world and the vehicle frames should be presented. Figure 2(a) shows the notation for goal-following control on a horizontal plane. X-Y is the world frame and Xv-Yv is the vehicle frame. Let **X** be the position vector of the vehicle on the world frame. S is the start point, **GW** is the position vector of the goal with respect to S, both on the world frame. The angle ψ, ψ_b and ψ_r can be given as follow (their subscripts will be explained later):

$$\psi = \psi_b - \psi_r, \tag{2}$$

G is the goal on the world frame, for simplification we have defined G on the Y-axis, thus ψ_r approaches zero. Additional details in our path re-planning algorithm (Fig. 1) are explained as following steps:

1: Define the goal and start point on the world frame.

2: Calculate the net vector of the **GW** – **X**, that is **GV** on the world frame, see Fig. 2(a).

3: Calculate the angle ψ, in this case; $\psi_r \rightarrow 0$.

4: Calculate the position vector of the goal on the vehicle frame (the sonar view); **GV** by this equation:

$$\mathbf{GV} = R^T (\mathbf{GW} - \mathbf{X}), \tag{3}$$

where R^T is a transpose of 2D rotation matrix, that is:

$$R^T = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix}. \tag{4}$$

5: Simulate the obstacle and the protection circle on the world frame.

6: Send a center position of obstacle(s) from the world frame to the vehicle frame.

7: Simulate the obstacle(s) and the protection circle on the Image Processing Library (IPL) image (500×866 pixels) on the sonar view.

8: Define the region of the sonar view, which has a wide beam 120 degrees in a fan-shaped area, and 30 meter range. The region of the sonar view is a constraint for an A* algorithm to search the path within this region.

9: Set the goal and start points for an A* search within the sonar view.

10: Search the optimal path by an A* algorithm within the sonar view. If the path is found then display that path in the sonar view.

11: If the path is found, send path information from the vehicle frame to the world frame. At the same time, define the distance interval between the waypoints (the found path) on the world frame as shown in Fig. 2(b).

12: On the world frame, if the path is found, calculate the position vector of the waypoints (P_i) for tracking that is **PW** by this equation, also see in Fig. 2(b):

$$\mathbf{PW} = \mathbf{X} + \mathrm{R}\,\mathbf{PV}, \tag{5}$$

where **PV** is the position vector of the waypoints on the vehicle frame . **X** is the current position vector of the vehicle on the world frame, and R is a 2D rotation matrix, that is:

$$\mathrm{R} = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix}. \tag{6}$$

12.1: Count the number of the waypoints.

12.2: Save the position of the *current waypoints* in the name of the *previous waypoints* (see in Fig. 1).

12.3: Show the position of the *current waypoints*.

12.4: If the path is not found:

12.4.1: Set the position of the *previous waypoints* to be the position of the *current waypoints* (Fig. 1).

12.4.2: Show the position of the *current waypoints*.

13: Show the position of the obstacle(s) and the vehicle on the world frame.

4. Simulations

4.1. Software Architecture

The operating system of main computer is Windows XP™. The application software for the graphic user interface (GUI) and dynamic controls of the ASV model are implemented with Visual C++, Microsoft™ Visual Studio 2005. The software architecture consists of three programs:

High Level Program; it simulates a dynamic motion of the vehicle model. It mainly consists of the Proportional (P) controller and our path re-planning algorithm adapted from A* algorithm. By the model-based simulation in our work, the P controller was tuned to easily perform the controlled response with sufficient stability, without a derivative or an integral compensation. Our path re-planning algorithm updates every 1.6 s or more than that depends on the system optimization. As the applicability in target application, the latitude and longitude of the vehicle model have been simulated and transformed to X and Y coordinates (world frame) and displayed in our GUI program.

Mid Level Program; this part is shared memory segments between the *High Level Program* and the *Low Level Program*.

Low Level Program; this part consists of several threads that run with multi-threads in real-time. Firstly, the *DeltaT Thread* that simulates the obstacle(s) in the sonar view and saves sonar images in JPEG format.

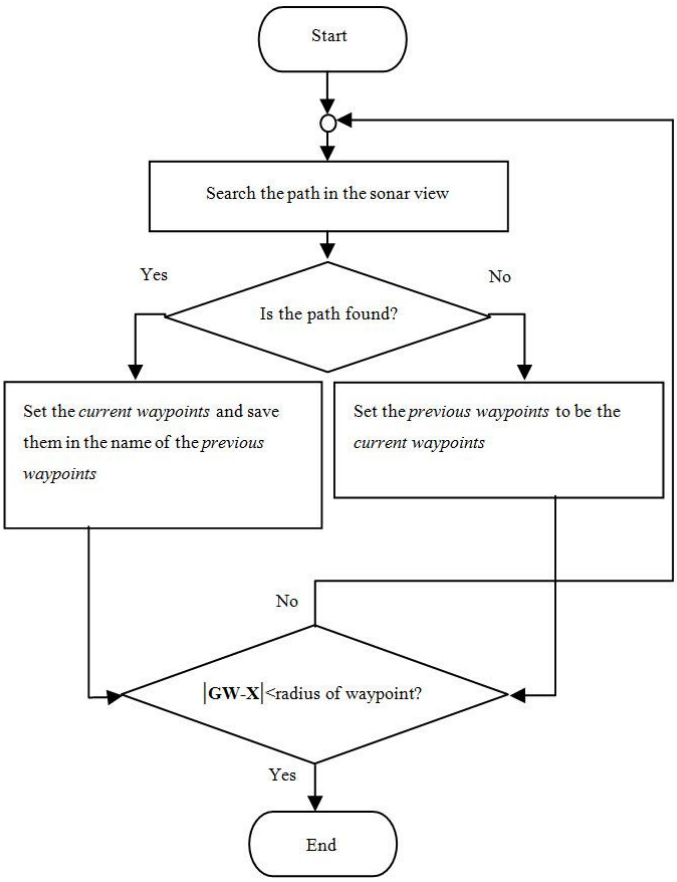

Fig. 1. Path re-planning algorithm.

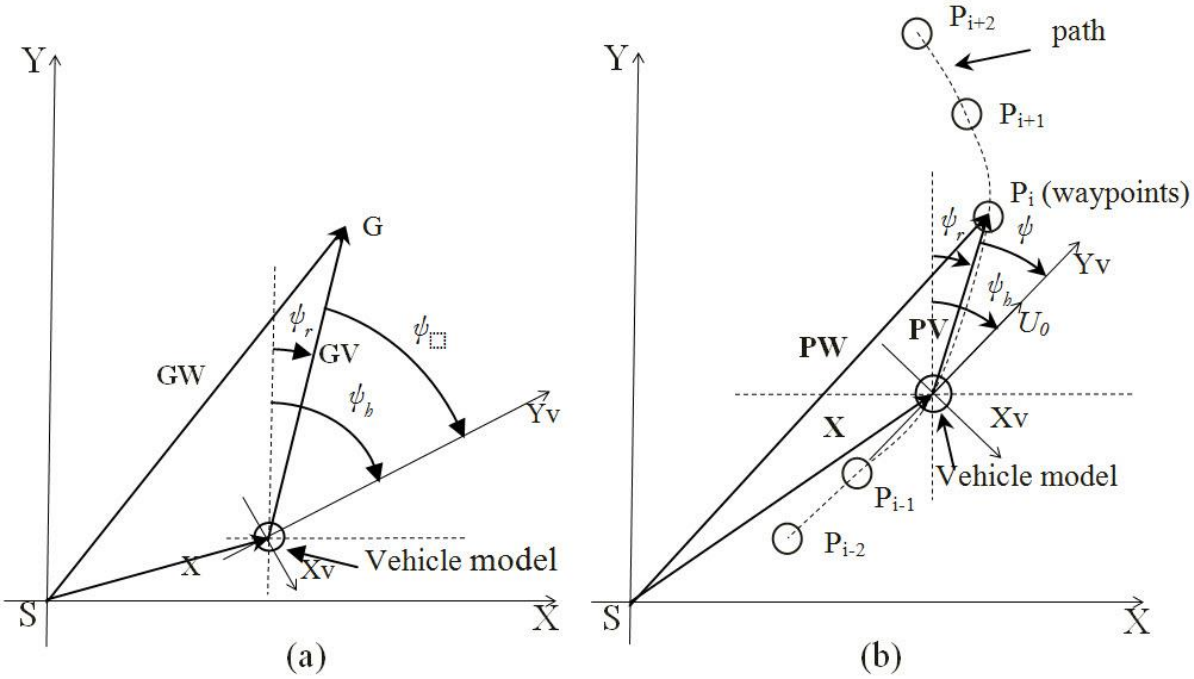

Fig. 2. The goal on the transformed coordinates (a) and traced waypoints on the transformed coordinates (b).

This thread is ready to adapt for the real-time program in the sea trials as our target application. The *Path Finder Thread* uses an A* algorithm to generate path between the start point and the goal point within the

sonar view. The *AHRS Thread* simulates the heading of the vehicle model. Lastly, the *Thruster Thread* simulates thrust commands to control the vehicle model.

4.2. Waypoints Tracing

To track waypoints, the vehicle controls its dynamic motion with surge and heading control as shown in Fig. 2(b). To avoid an underwater obstacle(s), the vehicle independently controls its motion in 2-degrees of freedom; surge and heading. It is assumed that roll and pitch motions are stable. For surge motion, we have no distance measurement equipments, both simulation and tank experiments, therefore we have defined the surge reference (*SurgeRef*) to control the vehicle without sensory feedback data that it has been controlled by an open-loop. Our objectiveness is that the surge speed should be constant, therefore controller for surge motion is set by this relative:

$$SurgeRef = \text{constant}. \tag{7}$$

Indeed, *SurgeRef* is set to be an integer (see in section 6.3 for details). As for the heading control system, ψ_r is defined as the angular displacement reference of the ahead waypoint with respect to the Y-axis in the world frame as Fig. 2(b). If the position of each waypoint on trajectory does not change, it is so called "*planning mode*", therefore a set of angles ψ_r of waypoint P_{i-2}, P_{i-1}, P_i and so forth is constant. If the position of each waypoint on trajectory does change that is caused by our re-planning algorithm, so we call "*re-planning mode*", thus the set of angles ψ_r of those waypoints is redefined every 1.6 s. Therefore, the heading angle of the vehicle is controlled to make the heading to follow ψ_r that is:

$$\psi_r = \begin{cases} \text{constant}, & \text{planning mode}, \\ \text{redefined}, & \text{re-planning mode}. \end{cases} \tag{8}$$

The deviation ψ from ψ_r is calculated from (2) where ψ_h is the heading of the vehicle that calculated from the angle between the Y-axis and the forward speed of the vehicle (U_0) in the world frame; see Fig. 2(b). The heading reference (*HeadRef*) is generated by the following equation [17] where *Kh* is proportional gain;

$$HeadRef = Kh \times \psi. \tag{9}$$

4.3. Trajectory of the Vehicle Model

Although the A* generated path that have been dealt on the pixel-based map, the main idea of Fig. 3 that tries to explain the conceptual scheme of our obstacle avoidance and path re-planning that is prepared for positioning applicability in target application. In simulations; we have simulated the circular obstacle, the protection circle and the circular ASV model that its radius is r_{obs}, r_{pro} and r_{ASV}, respectively. In addition, the multi-beam FLS has been simulated that is equipped in front of the ASV model, and scans forward which 120° beam width and 30 meters range.

As the applicability of positioning in our target application, in Fig. 3, it is not necessary to measure the curve distance interval (C) in the centimeter level for the world frame by GPS receiver. To easily adapt with actual GPS, all pixels (1 pixel = 6 cm) from the A* generated path that have been represented by some points to be a set of waypoints P_{i-2}, P_{i-1}, P_i and so forth, then the distances C(s) between the waypoint P_{i-1} and P_i and so forth that are adequate long (these distances interval are optimal with the accuracy threshold of GPS receiver), and the angle θ (s) are quite wide. Therefore, the center of the vehicle model will enter the protection circle, or the actual trajectory will be generated inside of the protection circle as seen in Fig. 3. It seems that the actual trajectory as the result is not equal to the planned path at a certain time, however, this concept to increasingly smooth the obtained paths, and still reduces a memory and time consumption. On the other side, if all pixels from the A* generated path are taken to be a set of waypoints P_{i-2}, P_{i-1}, P_i and so forth, then the distances C(s) between the waypoint P_{i-1} and P_i and so forth that are more short (they are not optimal with the accuracy threshold of GPS receiver), and the angle θ (s) are very small. Therefore, the actual trajectory will not be generated on the inside of the protection circle. However, we have approximately defined the distance C between each waypoint as shown in Fig. 3. Hence, we can approximately calculate the angle θ by the definition of radian as:

$$\theta \cong \frac{C}{2r_{pro}}. \tag{10}$$

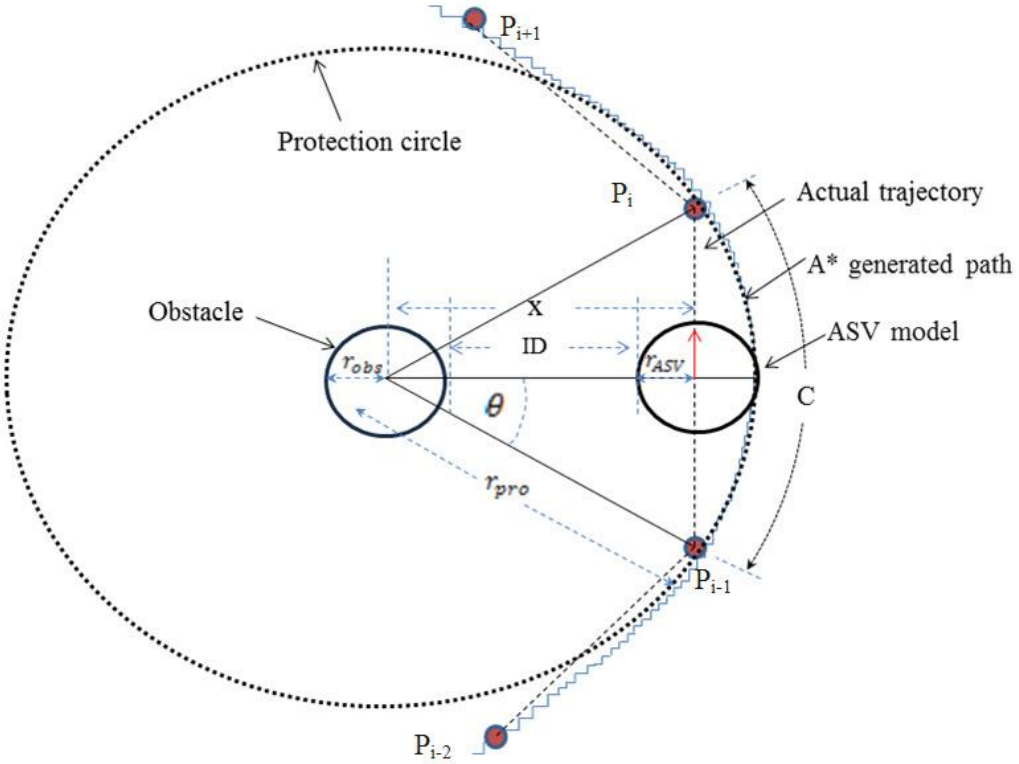

Fig. 3. The conceptual scheme of obstacle avoidance and path re-planning.

Also, we have a cosine function that:

$$x \cong r_{pro} \cos \theta; x \leq r_{pro}. \tag{11}$$

The inoffensive distance (ID) in a case of both the centre of ASV model is on the inside and the outside of the protection circle can approximately be calculated by:

$$ID \cong \begin{cases} r_{pro}\cos\theta - r_{obs} - r_{ASV} & ; x \leq r_{pro} \\ x - r_{obs} - r_{ASV} & ; x > r_{pro}. \end{cases} \tag{12}$$

4.4. Obstacle Avoidances

The simulated static circular obstacles are shown in the sonar view as Fig. 4(a), their radii are 0.50 m (denoted as red spots), and also show the protection circles with light-green circular regions, and their radii of the protection circles are 3.5 m. An A* algorithm has generated the path denoted as a yellow line linked between the start and the goal point in each sonar view.

For simulation in the world frame, we have defined the start and goal point at (X, Y) = (0, 0) and (X, Y) = (0, 20 m), respectively. We have defined a circular waypoint, and assign its radius of each waypoint of 0.50 m. Then, we have simulated a static circular obstacle that its centre is located at (X, Y) = (0, 10 m), its radius is 0.50 m. As the target application, the position of an ASV will be measured by GPS. Therefore, the simulated latitude and longitude of an ASV model have been transformed to X and Y coordinates (the world frame). We have approximately defined the distance *C*: 1.3 m (see Fig. 3). Figure 5(a) has shown the first waypoints that have been generated in the world frame, at 4 s. The ASV model is denoted as a blue circle (r_{ASV}= 0.5 m).

In Fig. 4(b), at 26 s, in the sonar view, the new position of the goal and the obstacle with respect to the vehicle is computed and set. The result is that they have been moved forward the sonar head with respect to the vehicle frame. At the same time, in the world frame as seen in Fig. 5(b), the (new) *current waypoints* are shown on the right side of the obstacle that unlike the *previous waypoints* that have been shown on the left side of the obstacle. The vehicle tracks those current waypoints by right turning to avoid the obstacle with the heading 27.3 degree.

In Fig. 4(c), in the sonar view, at 36 s, an A* algorithm attempts to find the new path, but the new path cannot be found; because the vehicle model's present position is within the region of the protection circle. The start point in the sonar view is concealed by the protection circle area, then an A* algorithm does not know its own start point, the result is the new path does not exist. In the world frame, at the moment, in Fig. 5(c), as there is no path information, therefore the *previous waypoints* have been set to be the position of the *current waypoints* as our path re-planning algorithm that they are shown as magenta circles.

In Fig. 4(d), in the sonar view, having been released from the region of the protection circle, there is no obstacle. At 95 s, an A* algorithm has generated a very short path, it means that the vehicle has arrived the goal point already. At the moment as Fig. 5(d), in the world frame, the vehicle also has reached the goal at (X, Y) = (≈ 0 m, 19.68 m) when compare with its defined goal (X, Y) = (0.0 m, 20.00 m). It has a position error less than 0.5 m (radius of waypoint); because of the waypoint hit condition that has been defined as the equation in [18]. The trajectories of the vehicle model have been depicted as shaded blue lines.

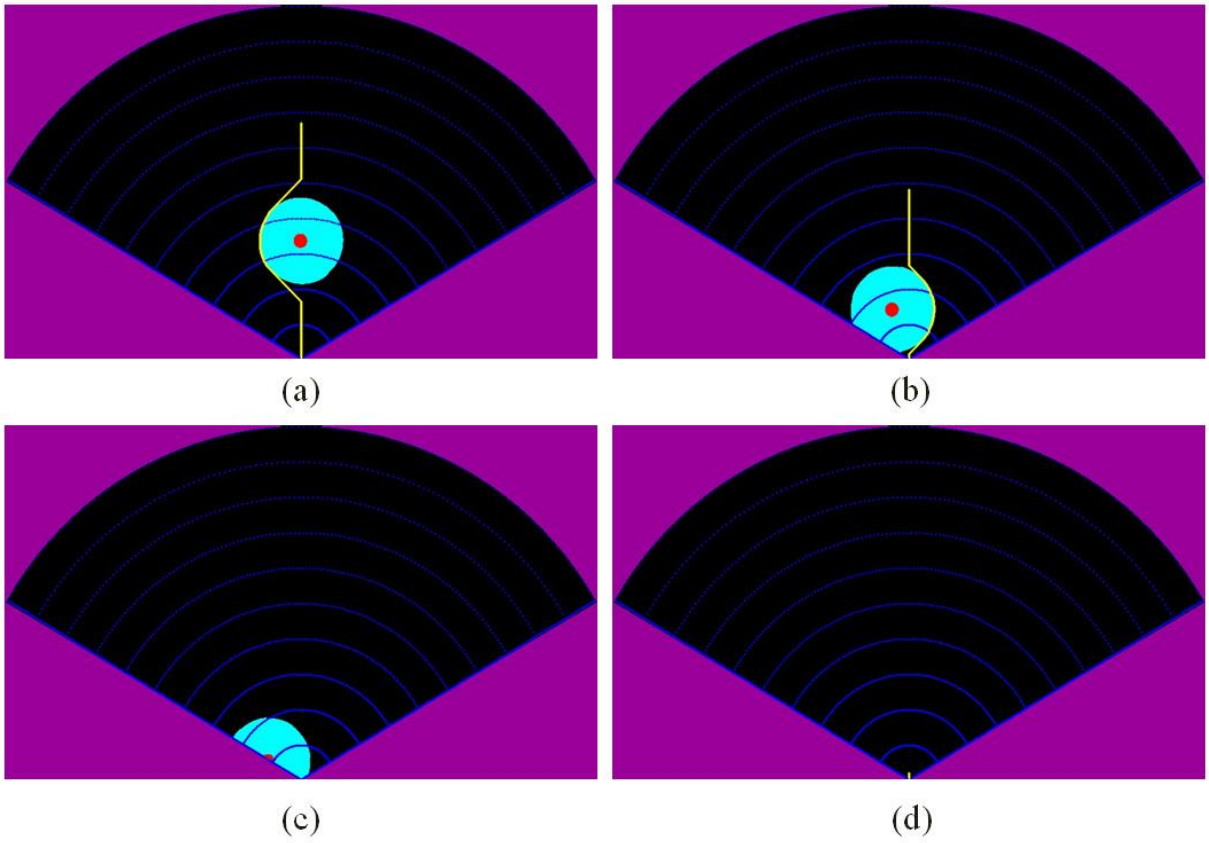

Fig. 4. Paths generated to avoid the static obstacle in the sonar view. (a) At 4 s, (b) at 27 s, (c) at 36 s, there is no path for this view and (d) at 95 s; it means the vehicle reaches the goal.

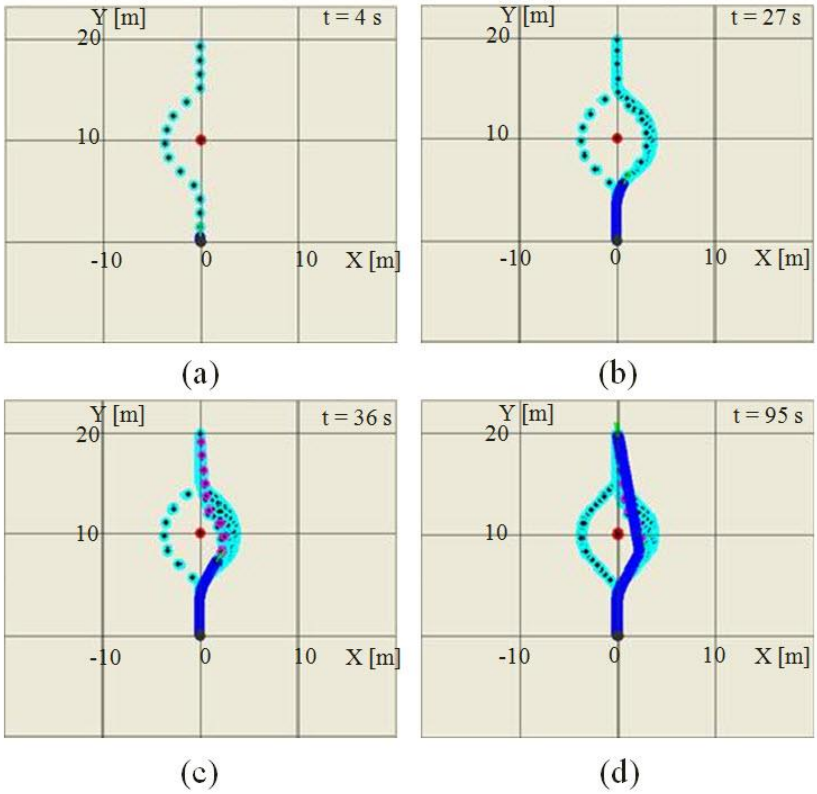

Fig. 5. Waypoints and trajectories generated to avoid the static obstacle in the world frame. (a) At 4 s, (b) at 27 s, (c) at 36 s and (d) at 95 s; the vehicle arrives the goal.

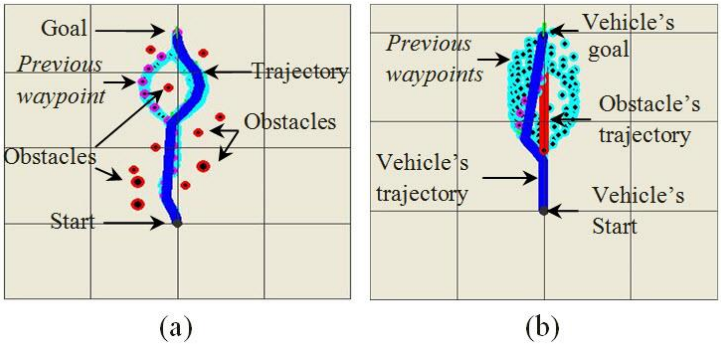

Fig. 6. Avoidance trajectories of the vehicle with multiple obstacles (a) and moving obstacle (b).

To enhance the effectiveness of the proposed algorithm, we have dealt avoidances with multiple static obstacles and moving obstacle. Figure 6(a) shows its overall avoidance trajectory; the red circles represent obstacles consisting of several their radii. Figure 6(b) shows the trajectories of the vehicle model (shade of blue) and a moving obstacle (shade of red). To clarify the limitation and applicability of the proposed path re-planning method: the time history of the inoffensive distance (ID) of the different size of static obstacles has been added for the simulation result. Figure 7(a) shows the time history of the ID that calculated from (12) for the different radii of the static obstacles; those are 0.2 m, 0.5 m, 1.0 m and 1.5 m, while the vehicle's positions at the start and goal point, the positions of the obstacles, radii of protection circles (3.5 m) and vehicles (0.5 m), and the average surge speed of the vehicle (0.22 m/s) are specified as before.

DOI:10.4186/ej.2015.19.1.107

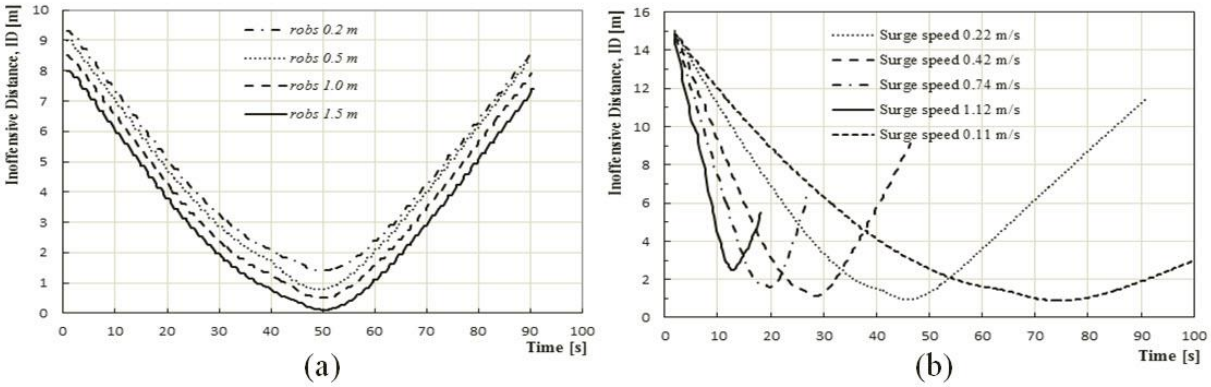

Fig. 7. ID with different sizes of obstacles (a) and ID with different speeds of the vehicle.

With these assignments, if the radius of obstacle (*robs*) is enlarged to 1.5 m, the vehicle will increase the risk of obstacle collision, especially at (approx.) 50 s. Figure 7(b) shows the time history of the ID for the different average surge speeds of the vehicle; such as 0.11 m/s, 0.22 m/s, 0.42 m/s, 0.74 m/s and 1.12 m/s, meanwhile the obstacle head-on moves to the vehicle with the average constant speed (0.11 m/s). Also, radii of the obstacle, the protection circle and the vehicle are specified as 0.5 m, 3.5 m and 0.5 m, respectively, and the vehicle's position at the start and goal point are specified as before. It shows that the minima ID(s) have been less affected by the surge speed variation at low speeds; approximately from 0.11 m/s to nearly 0.74 m/s. However, at high surge speed (approx. 1.12 m/s), the heading of the vehicle has been more fluctuated, and therefore, it does have an effect on the stability and performance of the vehicle, this is the limitation. To verify the effectiveness of the proposed method with a challenge, the environment with the local minimum has been carried out. In the sonar view, Fig. 8(a), the protection circles have been adapted to the protection ellipses, the positions of obstacles have been assigned at close range, and then their protection ellipses are overlapped and formed to be the local minimum region, also the goal point of the vehicle is assigned in the local minimum region. Figure 8(b) shows its overall avoidance trajectory of the vehicle for the local minimum environment in the world frame.

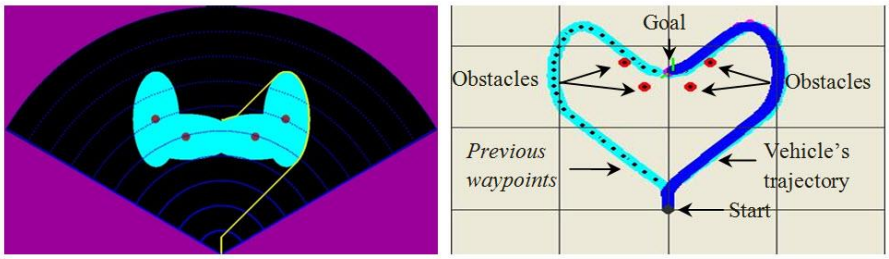

Fig. 8. Protection ellipses are overlapped to build the local minimum region in the sonar view (a) and the avoidance trajectory for the local minimum environment in the world frame (b).

5. Sonar Images Processing and Noises

Although we have no chance to test the underwater obstacle avoidance of the vehicle with actual multi-beam FLS, however, before now, we collected the raw sonar image with a full-resolution ping matrix captured from the wall of the cubic pool (each side of a cubic was 8 m long) that were taken by the Imagenex™ DeltaT sonar [19]. It has the specification such as; number of beams: 120 and 500 range bins per beam (high resolution); sector: 120° ; vertical beam-width: 3°. The intensities of beams data (yellow shades) of the pool's wall were shown as Fig. 9(a) (sonar scanned downward; range of sonar: 10 m). A common segmentation for sonar image consists of filtering and thresholding [14].The filtering is in charge of the Imagenex™ DeltaT sonar; in this sonar, there are various filter settings could be used that depend on the bottom type or in the water column, e.g. rocks, wrecks, fish schools, piers. For instance, the Remove Short Outliers filter [19]; short outliers are unwanted targets above the bottom and in the water column that they should be removed. For thresholding; the fixed threshold technique was used in our sonar images,

such as the intensities of beams have been represented or fixed with red ellipses on the segmented image; see Fig. 9(b). Then, protection ellipses have been built as boundaries (light blue shades) enclosing red ellipses (obstacles) on the segmented image as Fig. 9(c). Although the pool's wall has been used to represent obstacles, however, these steps just want to exhibit our sonar image processing maneuver. However, noises still remained in the pool test, therefore the sonar head was tested to horizontally scan a cylindrical obstacle (piling) in the lake to reduce short outliers (noises) above the bottom, and the obtained segmented sonar image was very clearance as Fig. 9(d) (range: 10 m).

6. Tank Experiments

6.1. The ASV

After testing in simulations, in order to verify the performance of the proposed method, a tank test is carried out using an actual ASV. Currently, we have no chance to test with actual multi-beam sonar.

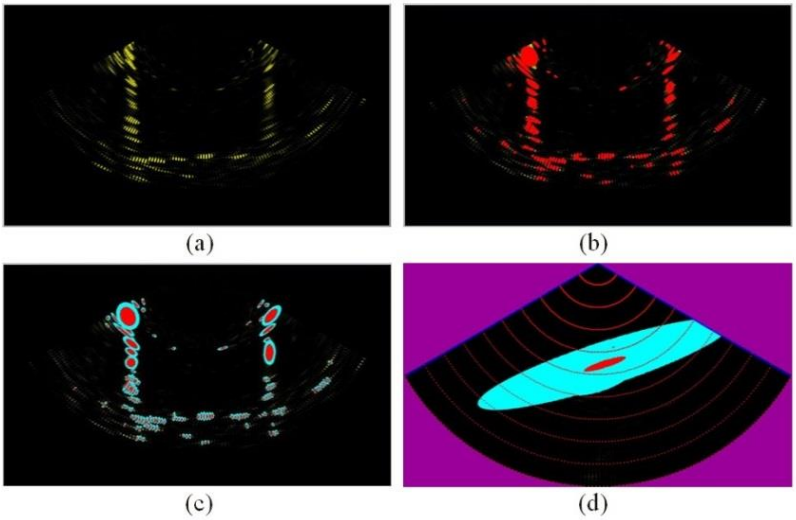

Fig. 9. The raw sonar image with a full-resolution ping matrix captured from the pool (a), several segmented sonar images that show our fixed threshold technique (b); protection ellipses (c); and example of image of piling taken from the lake (d).

Therefore we have to simulate the obstacle in the sonar view as mentioned in simulations. The specifications and the appearance of our ASV are shown in Table 1 and Fig. 10(b). Surge and yaw motions of our ASV can be independently controlled by two thrusters. The ASV consists of a hull for computers, electronics devices and batteries. Its heading angle is measured by Attitude Heading Reference System (AHRS). Its onboard avoidance system performs basic waypoint-following maneuvers in real-time with our position estimation that does not use a Doppler Velocity Log (DVL) or Fiber Optical Gyro (FOG). This is the lack of high precisely positioning equipments, because of budget limitation. However, this is our challenge under the lack of high technology apparatuses.

6.2. Hardware and Software System

The core of the ASV hardware is the main computer interfaced with the wireless LAN adapter *via* the Ethernet (10 Base-T) using TCP/IP and a switching hub. The wireless LAN is used to send a start command to the ASV. A thruster controller box connected to the main computer *via* USB cables. Figure 10(a) shows the hardware diagram of the ASV. For software, we still use the concept of three programs as previously simulation to control an actual vehicle. We have adapted in the *Low Level Program* to test with actual ASV. Therefore, two threads have been adapted for tank test that are: for the *AHRS Thread*, it has been adjusted for Crossbow™ NAV440CA-202 (AHRS) to measure the actual heading of the vehicle in real-time, and the *Thruster Thread* has been adapted to control actual thrusters of the vehicle to track the waypoints in real-time.

DOI:10.4186/ej.2015.19.1.107

6.3. Control Scheme in Tank Experiments

To track waypoints (generated path) for the obstacle avoidance of actual ASV in tank test, the actual vehicle is controlled in 2-degrees of freedom (surge and heading) independently. The P controller controls heading angle by *HeadRef* in (9), also see in (2) and Fig. 2(b). The angular feedback data from Crossbow™ NAV440CA-202 (AHRS) are added into heading control as shown in Fig. 11(a) to guarantee that the vehicle can follow its generated path (waypoints). Figure 12(a) shows an example of control result of heading control. For the thruster control scheme of the vehicle, we have defined thrust command (integer numbers; 0-255) of each thruster that is calculated by the following equations:

$$f(Th_L)=12.8\sqrt{\frac{SurgeRef + HeadRef}{2}}+128, \quad (13)$$

$$f(Th_R)=12.8\sqrt{\frac{SurgeRef - HeadRef}{2}}+128, \quad (14)$$

where $f(Th_L)$ and $f(Th_R)$ is the thrust command of the left and the right thruster, respectively. *SurgeRef* is set as a constant in real experiments as (7). *HeadRef* is deviated in real experiments as (9). *SurgeRef* and *HeadRef* are both integers that have been defined a range of -100 to 100. To avoid square roots of negative numbers that are $\frac{SurgeRef \pm HeadRef}{2}<0$, then they have been redefined as $f(Th_{L,R}) = -12.8\sqrt{-\left(\frac{SurgeRef \pm HeadRef}{2}\right)}+128$. Thrust commands have been transformed to the percentage of duty cycles of pulse-width modulation, or (PWM) technique control for 2 thrusters, such that thrust commands; 0-126 request 0% to 49% duty cycles (propeller rotates anticlockwise to propel the vehicle backward when viewed from astern, see Fig. 10(b), 129-255 request 51% to 100% duty cycles (propeller rotates clockwise to drive the vehicle forward). For optimization in practice, if the thrust command is set the range of 127 and 128, the propeller will be stopped. For instance, we define *SurgeRef* = -45 and *HeadRef* =+55, then $f(Th_L)$=156(156.62) (left propeller rotates clockwise) and $f(Th_R)$=37(37.49) (right propeller rotates anticlockwise), and the result is the vehicle will be turned to right with constant surge speed. Indeed, our vehicle can move backward by setting the range of *SurgeRef* from -1 to -100, however, the vehicle has been tested only a forward motion. Example of thrust commands from avoidance experiment is shown as Fig. 12(b). However, kinematic performance of the vehicle from a viewpoint of thrust force can be considered from the non-linear interrelation of the thrust command and thrust force for each thruster as shown in Fig. 13. Notice that this interrelation is gathered from total force (kgf) acting on each thruster in the water, each thrust command, measured by seizing each thruster with a hook of spring balance apparatus. Normally the vehicle at the surface of the water will drift, however, in our case the vehicle has been ballasted by payload masses approx. 95 kg, and others disturbances such as waves, currents and winds are controlled, therefore it is very less affected. The vehicle dynamics in tank is based on the proposed position stochastic estimation (not a high accurate positioning system), also precise bottom-referenced surge velocity measurement equipment such as Doppler velocity log (DVL) that has not been used, and therefore, it is inconvenient to use a braking force in our vehicle.

Table 1. Specifications of the ASV.

Size	2.04 m (L) x 1.0 m (H) x 1.20 m (W)
Mass	95 kg (with payloads)
Max. speed	1.0 m/s
Duration	4 hours
Actuators	Minn Kota™ 120 W thruster x 2
Power	Ni-Cd 25.2 V 20 Ah
Processor	Intel™ Core 2 Duo 2.66 GHz (3.4 GB RAM)
OS	Windows ™ XP Professional
Communication	Wireless LAN
Heading, Roll and Pitch	Crossbow ™ NAV440CA-202 (AHRS)
Multi-beam sonar	Not available, now

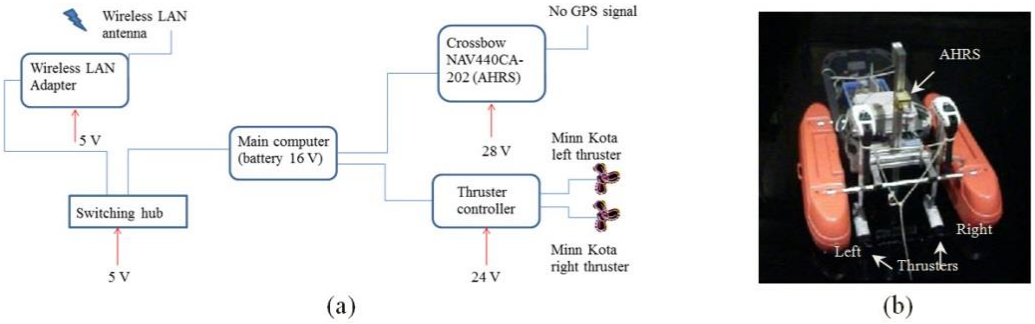

Fig. 10. Hardware diagram of the ASV (a) and the configuration of our ASV in tank experiments (b).

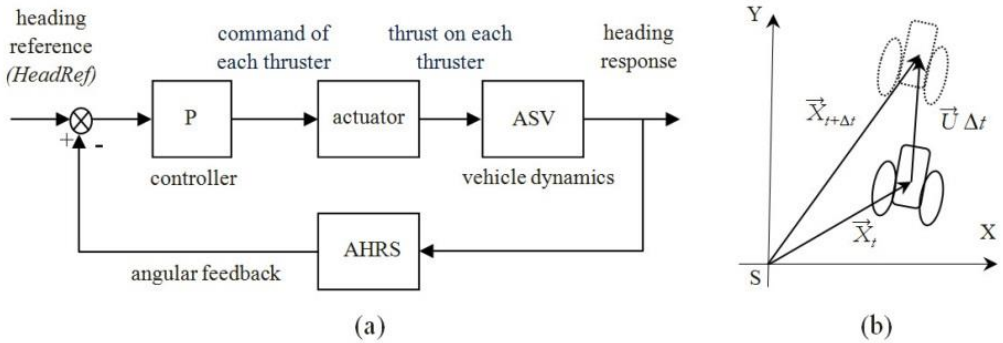

Fig. 11. Block diagram of heading control of actual ASV (a) and the estimated position of the ASV (b).

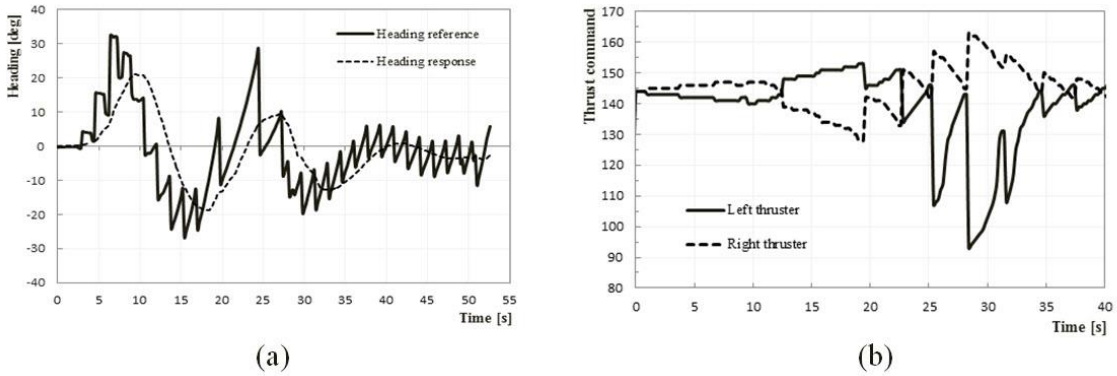

Fig. 12. Example of heading response with feedback control (a) and example of thrust commands acting on each thruster (b), remark: results in these two figures are not the same experiment.

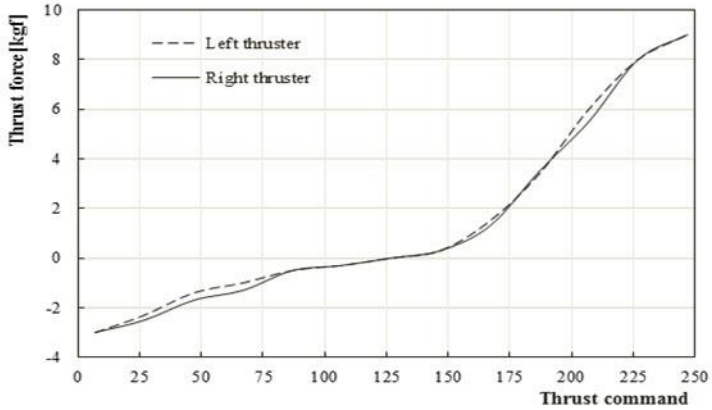

Fig. 13. Interrelation of thrust command and thrust force.

DOI:10.4186/ej.2015.19.1.107

6.4. Position Estimation

Now, the GPS module still has not been used because it cannot receive signals from satellites owing to the composition of tank's top part. However, we need to develop our real-time codes for the positioning of actual ASV that prepares for the applicability of target application as previously mentioned, and therefore, other range sensors; such as a laser range sensor still has not been used now. Then, we have addressed this problem based on the estimated positioning to overcome it, this method is similarly a dead reckoning process. Hence, the estimated position vector of actual vehicle is expressed as:

$$\vec{X}_{t+\Delta t}=\vec{X}_t+\vec{U}\,\Delta t, \tag{15}$$

where $\vec{X}_{t+\Delta t}$ is the estimated position vector of the vehicle at time $t+\Delta t$, $\vec{X}_t$ is the estimated position vector of the vehicle at time t, $\vec{U}$ is the estimated velocity of the vehicle, and Δt is a time interval; we used 0.2 s. The initial condition for this estimation is that $\vec{X}_t$ must be known at the initial time ($t = 0$). These vectors forms can be shown in Fig. 11(b). The scalar components of the estimated position vector of the vehicle at time $t+\Delta t$ can therefore be shown as:

$$X_{t+\Delta t}=X_t+Ux\,\Delta t \tag{16}$$

$$Y_{t+\Delta t}=Y_t+Uy\,\Delta t, \tag{17}$$

where $X_{t+\Delta t}$ and $Y_{t+\Delta t}$ is the scalar component of the estimated position vector of the vehicle of X-axis and Y-axis at time $t+\Delta t$, respectively. X_t and Y_t is the scalar component of the estimated position vector of the vehicle of X-axis and Y-axis at time t, respectively. Ux and Uy is the component of the vehicle estimated velocity in the water of X-axis and Y-axis, respectively.

Practically, the speed of the vehicle; U is proportional to thrust command; $f(Th)$. The approximately relation is often denoted as:

$$U \propto f(Th), \tag{18}$$

and then;

$$U \cong k_0\, f(Th), \tag{19}$$

where k_0 is so-called the proportionality constant. Therefore, the components of the vehicle estimated velocity in the water of X-axis and Y-axis can be expressed as:

$$U_x \cong k_0 f_x(Th), \tag{20}$$

$$U_y \cong k_0 f_y(Th). \tag{21}$$

We have assigned that $f_x(Th)$ and $f_y(Th)$ relate with the surge reference: *SurgeRef* and heading reference: *HeadRef* by these relations;

$$f_x(Th) = SurgeRef \times \sin(HeadRef), \tag{22}$$

$$f_y(Th) = SurgeRef \times \cos(HeadRef). \tag{23}$$

The scalar components of the estimated position vector of the vehicle at time $t+\Delta t$ can therefore be redefined as:

$$X_{t+\Delta t} \cong X_t + k \times SurgeRef \times \sin(HeadRef)\,\Delta t, \tag{24}$$

$$Y_{t+\Delta t} \cong Y_t + k \times SurgeRef \times \cos(HeadRef)\,\Delta t. \tag{25}$$

SurgeRef is surge reference assigned as a constant as in (7). *HeadRef* is heading reference as defined in (9). Then, k relates with k_0 and approximately relates with a drag force, density of the water, vehicle cross section area and velocity of the vehicle that it is practically determined from tank experiments. The heading angle of the vehicle is measured by the actual AHRS, therefore there is no estimated equation for the heading angle of the vehicle in tank experiments.

6.5. Obstacle Avoidances in Tank Experiments

Before this, the vehicle was tested with straight-line motion that it meant no any simulated obstacle was assigned in the sonar view. The start and goal of the vehicle were set in the estimated frame at (X, Y) = (0 m, 0 m) and (X, Y) = (-0.25 m, 8.5 m), respectively. The result was the position of actual goal of the vehicle was not drifted from the estimated goal, therefore we could conclude that the vehicle was not affected by any force for this phenomenon. After that, the static circular obstacle has been simulated in the sonar view (radius of obstacle: 0.5 m; radius of protection circle: 1.5 m), the start and goal of the vehicle are set as

before. Then, some examples of scene of the vehicle dynamic motions for simulated static obstacle avoidance experiment in tank are shown in Fig. 14(a), and the position of the static obstacle has been assumed and it is denoted as a red spot. A comparison of the estimated and actual avoidance trajectories of the vehicle for static obstacle is shown in Fig. 15(a). The position of actual goal of the vehicle has been drifted from the estimated goal. As comparison of straight-line motion and simulated obstacle avoidance motion of the vehicle that we are confident that the main influence of this dead reckoning error (drifted) is caused by the lift force [20] that acts at the vehicle while it is turning on a horizontal water plane to avoid the simulated obstacle.

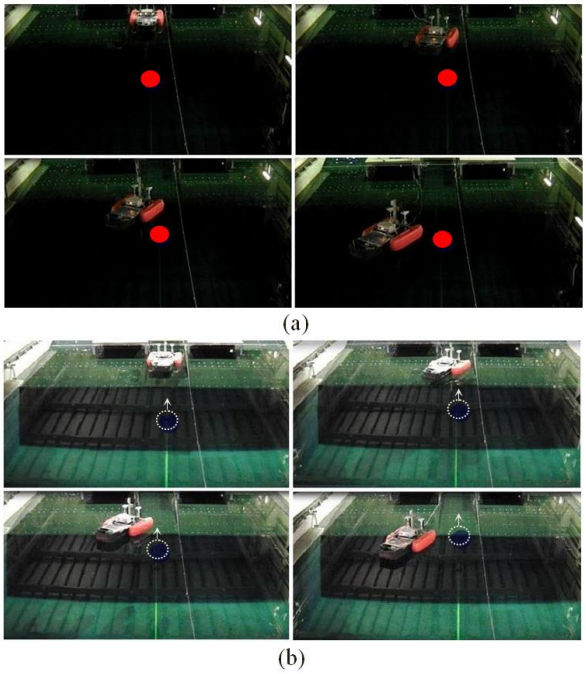

Fig. 14. Some examples of scene of the vehicle dynamic motions in tank experiments for simulated static obstacle (a) and simulated moving obstacle (b) avoidances.

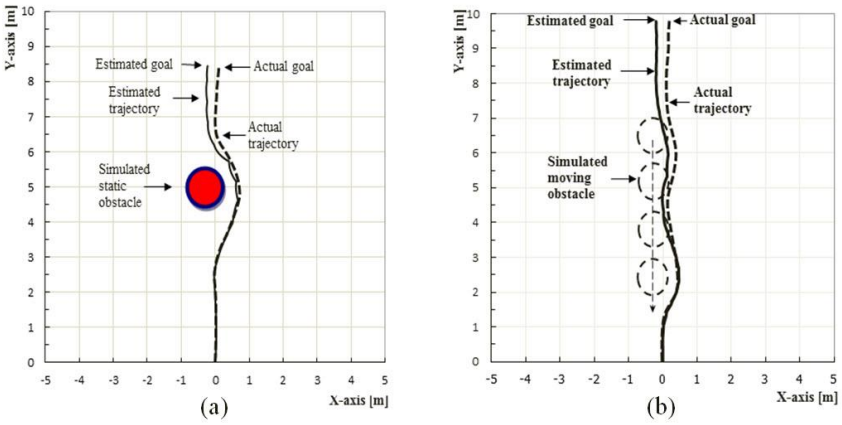

Fig. 15. Comparisons of the estimated and actual trajectories of the vehicle for simulated static obstacle (a) and moving obstacle (b) plotting on the actual position of the tank.

For the moving obstacle on the estimated frame, the start of simulated moving obstacle is set at (X, Y) = (-0.25 m, 6.5 m), and represented with a dotted circle, their radius and radius of protection circle are the same as before. The goal of the moving obstacle is assigned at (X, Y) = (-0.25 m, 0 m). The obstacle moves with a straight-line path to its goal with the estimated average speed 0.11 m/s. The vehicle starts at (X, Y) = (0.0 m, 0.0 m) moves with the estimated average speed 0.22 m/s, having avoided the simulated moving obstacle, it arrives the goal point at approx. (X, Y) = (-0.25 m, 9.75 m) as shown in Fig. 14(b). Some

examples of scene of the vehicle dynamic motions for simulated moving obstacle avoidance experiment in tank are shown in Fig. 14(b), the position of the moving obstacle has been assumed and it is denoted as a dotted circle and its estimated speed vector shows as an arrow. Notice that the vehicle is tied with long rope for the reason of safety and convenience. Then, in Fig. 15(b), comparison of the estimated and actual avoidance trajectories of the vehicle for moving obstacle is shown. As the result, the position of actual goal of the vehicle has been drifted from the estimated goal, which is approximately similar to the experiment of the static obstacle avoidance. In Fig. 16, vehicle heading (heading response) in time sequence for static and moving obstacle avoidances have been measured by Crossbow™ NAV440CA-202 (AHRS) to exhibit that the position errors of actual trajectories are not mainly due to the errors of heading measurement, but they are caused by the effects of the lift force. The order of Reynolds number of our vehicle is sufficiently large, thus the lift and drag forces depend solely with the angle of attack, also this effect still has taken place with R-One AUV [21].

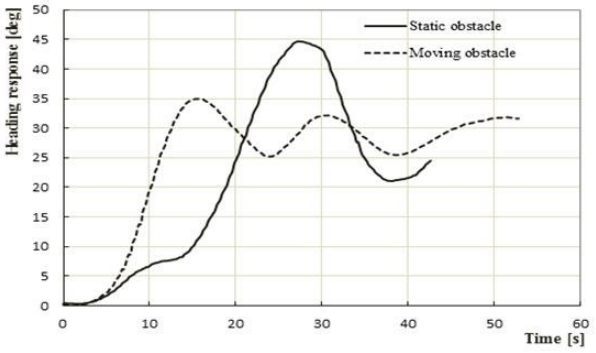

Fig. 16. Time sequence of the vehicle headings (heading response).

However, our vehicle will be able to precisely align its heading by controlling the left/right thrusters while it is drifted owing to the lift force, if the position of the vehicle has been measured by a high accurate positioning system, or precise bottom-referenced surge velocity measurement equipment such as DVL. Anyway, the dead reckoning error occurred owing to DVL that was still large as shown in [17]. Therefore, it is more difficult to precisely control the thrusters while the vehicle is acted by the lift force in the estimated positioning system. However, now, our position estimation of the vehicle is still adequate for the overall performance for underwater obstacle avoidances in tank experiments. As our target application, the position of the vehicle will be measured by actual GPS at the sea, when that time comes, even if the vehicle drifts owing to the lift force, it will be able to cruise almost perfectly (less position error) that based on an accurate GPS measurement. Therefore, currently, we have decided not to deal the lift force parameters into equations of estimated position for the vehicle.

7. Conclusion and Future Work

In this paper, an A* algorithm has been adopted to show that it can be applied in frameworks for performing 2D real-time underwater obstacle avoidance and path re-planning for ASV based on simulated multi-beam forward looking sonar images that it is successfully completed in simulations. For tank experiments, the proposed method is implemented in actual ASV. The vehicle moderately succeeds in its operations by means of the proposed position stochastic estimation and the simulation of circular obstacle in the sonar view. As our target applications, we are developing our system to have the ability of undersea obstacle avoidance using actual multi-beam FLS. However, there are several types of obstacles in the sea such as piers, rocks, or others. Therefore, an ellipse feature is optimal that should be used for obstacles representation for segmentation in a sonar image processing, because an ellipse is more flexible and adaptable, and compound objects can be easily constructed using ellipse shapes.

Acknowledgements

The author wish to thank Professor Tamaki Ura for his contribution in the experimental apparatuses, Associate Professor Toshihiro Maki for his help in software architectures, Dr. Kangsoo Kim for his

consulting in optimal control theories, Mr. Takashi Sakamaki for his help in tank experiments, and Ura laboratory members, at Institute of Industrial Science, Komaba Research Campus, The University of Tokyo, Japan. The author also thanks Professor Hayato Kondo for his invitation, Dr. Jin-Kyu Choi; postdoctoral fellow for his help in sonar image processing techniques, staff and students at Kondo laboratory, Tokyo University of Marine Science and Technology, Japan for the first part of the research.

References

[1] J. Gao, D. Xu, N. Zhao, and W. Yan, "A potential field method for bottom navigation of autonomous underwater vehicles," *Intelligent Control and Automation*, Chongqing, 2008, pp. 7466–7470.
[2] T. W. McLain and R. W. Beard, "Successive Galerkin approximations to the nonlinear optimal control of an underwater robotic vehicle," in *Proceedings of the 1998 IEEE international conference on robotics & automation*, 1998, pp. 762–767.
[3] B. Rhoads, I. Mezic, and A. Poje, "Minimum time feedback control of autonomous underwater vehicles," in *Proceedings of Decision and Control*, Atlanta, Georgia, USA, 2010, pp. 5828–5834.
[4] K. Kim and T. Ura, "Optimal guidance for autonomous underwater vehicle navigation within undersea areas of current disturbances," *Advanced Robotics*, vol. 23, pp. 601–628, 2009.
[5] I. Spangelo and O. Egeland, "Path planning and collision avoidance for underwater vehicles using optimal control," *IEEE Journal of Oceanic Engineering*, vol. 19, pp. 502–511, 1994.
[6] R. P. Kumar, A. Dasgupta, and C. S. Kumar, "Real-time optimal motion planning for autonomous underwater vehicles," *Ocean Engineering*, vol. 32, pp. 1431–1447, 2005.
[7] N. Sadegh, "Time optimal motion planning of autonomous vehicles in the presence of obstacles," *American Control Conference*, Seattle, Washington, USA, 2008, pp. 1830–1835.
[8] A. E. Bryson and Y. C. Ho, "Optimization problem for dynamic system,"*Applied Optimal Control*, London, UK: Taylor & Francis, 1975, ch. 2, sec. 2.7, pp. 77-80.
[9] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE Journal of Oceanic Engineering*, vol. 29, pp. 418–429, 2004.
[10] Z. H. Chang, Z. D. Tang, H. G. Cai, X. C. Shi, and X. Q. Bian, "GA path planning for AUV to avoid moving obstacles based on forward looking sonar," in *Proceeding of the Forth International Conference on Machine Learning and Cybernetics*, Guangzhou, 2005, pp. 1498-1502.
[11] Y. I. Lee, Y. G. Kim, and L. J. Kohout, "An intelligent collision avoidance system for AUVs using fuzzy relational products," *Information Sciences*, vol. 158, pp. 209–232, 2004.
[12] C. S. Tan, R. Sutton, and J. Chudley, "An integrated collision avoidance system for autonomous underwater vehicles," *International Journal of Control*, vol. 80, no. 7, pp. 1027–1049, 2007.
[13] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of A*," *Journal of ACM*, vol. 32, pp. 505–536, 1985.
[14] Y. Petillot, I. T. Ruiz, and D. M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *IEEE Journal of Oceanic Engineering*, vol. 26, pp. 240–251, 2001.
[15] P. Lester. 2005. *A* Path finding for Beginners*. http://www.policyalmanac.org/games/AStarTutorial.htm.
[16] G. E. Jan, K. Y. Chang, S. Gao, and I. Parberry, "A 4-geometry maze router and its application on multi-terminal nets," *ACM Transactions,* vol. 10, pp. 116–135, 2005.
[17] H. Kondo and T. Ura, "Navigation of an AUV for investigation of underwater structures," *Control Engineering Practice*, vol. 12, pp. 1551–1559, 2004.
[18] T. Maki, H. Mizushima, H. Kondo, T. Ura, T. Sakamaki, and M. Yanagisawa, "Real time path planning of an AUV based on characteristics of passive acoustic landmarks for visual mapping of shallow vent fields," in *Proceeding of MTS/IEEE OCEANS2007*, Aberdeen, 2007, pp. 1–8.
[19] Imagenex Technology Corp., "DeltaT Multi-beam Sonar System Model 837/A/B," Port Coquitlam, British Columbia, Canada, 2011.
[20] A. E. Bryson and Y. C. Ho, "Parameter optimization problems,"*Applied Optimal Control*, London, UK: Taylor & Francis, 1975, ch. 1, sec. 1.2, pp. 8-9.
[21] K. Kim, "Optimal guidance and tracking control of autonomous underwater vehicle under environmental disturbances," Ph.D. thesis, The University of Tokyo, Tokyo, 2003.