Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 6, No. 2, May 2021, Pp. 169-176

On the experiment of path planning using multi-way points with A* algorithm for autonomous surface vehicle

Bayu Erfianto*[1], Adysti Adrianne[2], Ramzy Rashaun Arif[3]
Department of Information Technology, Universitas Telkom, Indonesia[1]
Department of Informatics, Universitas Telkom, Indonesia[2,3]

Article Info

Keywords:
Autonomous Surface Vehicle, Path Planning, A*, Way-point

Article history:
Received: April 23, 2021
Accepted: May 18, 2021
Published: May 31, 2021

Cite:
Erfianto, B. (2021). On the Experiment of Path Planning Using Multi-way Points with A* Algorithm for Autonomous Surface Vehicle. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(2). https://doi.org/10.22219/kinetik.v6i2.1244

*Corresponding author
Bayu Erfianto
E-mail address:
erfianto@telkomuniversity.ac.id

Abstract

Commonly, surveillance activities on lake waters is mostly carried out by using a surface vehicle as special-designed vehicle, especially to conduct water quality measurements, underwater surveys, and bathymetry mapping. However, conventional survey and monitoring still involves humans on the site. If a survey is conducted during strong wind conditions, it could jeopardize surveyor's safety. Therefore, a vehicle must have several criteria, e.g., it must be pretty spacious and comfortable to carry surveyors, free from engine vibrations, stabilized and easy to maneuver, and the surveyor's safety can be guaranteed. This paper discusses preliminary research aiming to develop an Autonomous Raft Vehicle (ARV), a type of autonomous unmanned surface vehicle. The ARV is equipped with autonomous control based on multi-way-points with an A* algorithm. Thus, a user only requires giving a command once initially during path planning. A* algorithm over multi-way-point could improve ARV navigation when there are obstacles along the predetermined trajectory. Hence the predetermined trajectory will be maintained throughout the mission. It is a significant contribution to this paper.

1. Introduction

Conventional water quality monitoring directly involves humans as surveyors. However, if monitoring is carried out during strong wind conditions or around the dangerous lake area, it could jeopardize the surveyor's safety. For that reason, the boat must have several criteria, such as it must be comfortable for surveyors, closed and free from engine vibrations, stabilized, and easy to maneuver at low speeds. The surveyor's safety during activities must be guaranteed, for example, to carry out bathymetry mapping equipment and monitor water surface conditions [1][2].

Several studies on the Autonomous Surface Vehicle (ASV) have been carried out to monitor water quality and other missions. The ships' design is generally using mono-hull or double-hull designs [3]. Besides, the criteria for ship design must accommodate tools and several types of equipment, not only surveyors. As the solution, several researchers propose autopilot technology for ASV integrated with autonomous control systems [4][5][6] or using long-distance radio remote control [7]. Using ASV, surveyors can monitor water conditions on the lake, especially in difficult or dangerous areas to explore.

The problem of the research is as follows. For trajectory tracking or trajectory control, the way-point technique is commonly used for path planning algorithms, such as in UAV, UGV, and ASV. Suppose the path planning is implemented in ASV. The surveyors only require entering the way-point coordinate as a reference along with the desired path. However, this method still exploits higher cross-track errors due to external disturbance, such as GPS signal error, wind, or waves. Several studies on automated path planning use an A* algorithm due to its less complexity and ease of implementation in embedded UAV controllers. A* is a heuristic search algorithm to find the shortest path using heuristic approaches, which give better performance automated path planning [8]. In this research, the existing path planning, which is commonly deployed in the autopilot controller software, will be integrated with A* Algorithm to better cross-track error. The following paragraph explains the contribution of the paper.

This paper presents an experiment of an Autonomous Raft Vehicle (ARV) prototype that is built for an unmanned surface vehicle. ARV is equipped with an autonomous control system, including hardware and software. We then developed an A* algorithm and integrated it into the way-point path planning in the existing controller. The intention is to improve ARV navigation by reducing cross-track error, especially during obstacle avoidance maneuvers. Using the A* algorithm in multi-way points can improve ARV trajectory convergence when many obstacles exist in the predetermined path. Hence, the predetermined trajectory is always maintained.

2. Research Method

2.1 Related Works

Various types of unmanned surface vehicle have been developed for military purposes, seafloor geographical observations, and water quality monitoring. Kaizu [9], in his research, uses USV to measure and analyze water quality. Meanwhile, Ferreira [10] uses commercial USV to create three-dimensional maps of habitats using GPS-navigated USV. Liu [5] developed a vehicle to mimic an amphibian creature controlled remotely to measure the mud layer's depth and the quality of water used for the paddy field area.

An autonomous Surface Vehicle (ASV) is an unmanned vehicles that operates on the water's surface. ASV can be a low-cost alternative to monitoring water areas with a high danger level because ASV can be operated unmanned or called Unmanned Surface Vehicle or USV, or autonomously (ASV) [11]. Several studies related to the development of low-cost ASV include research using ASV to measure the sea's depth [12]. ASV from jet-powered kayaks is used to take high-quality oceanographic data samples from shallow and dangerous waters that cannot be traversed by traditional boats [13] [14]. Several studies using USV have been carried out to map the waters' depth around the coast, such as those conducted by Fuad [15] and Arulmozhiyal [16].

In terms of USV, the vehicle is not fully autonomous. An autonomous surface water vehicle or autonomous USV, is then known as the Autonomous Surface Vehicle (ASV) that allows the ship to carry autonomous navigation system and measuring devices as a payload equipped with real-time online communication capabilities. The latest survey on unmanned vessels in environmental monitoring was carried out [17]. Current research on the use of robotics technology in the use of ASV is carried out by [18], [10], and [12].

ASV is generally designed with a mono-hull body with low friction and good maneuver stability in average weather. However, mono-hull design is challenging to operate in bad weather conditions or heavy water currents [12]. ASV with a double-hull design can carry more payload because it has more space, more stable, and better ability to withstand wind and waves. In his research, Podnar [7] uses ASV to monitor water quality using wireless water sensors. In his study, ASV was designed using a double-hull design by merging two RC ships containing a motor with a propeller.

Hitz [19] and Ferri [1] use ASV to monitor seawater quality in real-time. In their study, ASV is designed with a double-hull frame with a motor with a propeller on each hull. In [20], ASV was built for shallow water research. ASV is designed with a double-hull aluminum chassis and equipped with two thrusters. In [21], ASV was made using one brushless DC motor with one servo motor. The autopilot controller is based-on ArduPilot Mega 2.6 (APM 2.6) using an external U-Blox Neo6M GPS. In that study, the average radius error obtained was 2.2 meters, with a speed of 2.5 m/s. The research and experiments conducted by Suja [22] present the advantages of applying robot technology for bathymetry mapping applications. Exploitation with ASV dramatically increases sampling resolution, such as shallow water areas that are difficult for standard boats to pass. Explicitly, this research and experiment discuss the advantages of using ASV compared to using a manned boat to do bathymetry mapping.

Path planning is essential in developing Autonomous Surface Vehicles (ASV). ASV is very dependent on route planning because ASV has an autonomous feature where ASV can run without user control, so choosing the path to be passed by ASV is very important. Many researchers focused on route planning methods, among them are studies on the comparison of Rapidly exploring Random Tree (RRT), RRT* [23], Greedy and Dijkstra [24], and A* [25].

In [24], experiments have been conducted to develop route planning on an Unmanned Aerial Vehicle (UAV) that is used in agricultural areas. This study compares the Greedy algorithm and the Dijkstra algorithm, which is used to find the shortest path in visiting the watering area. A trial comparison of the two algorithms was carried out using Matlab with five case studies. This comparison shows that the total distance generated by the Dijkstra algorithm is smaller or equal to the total distance generated by the Greedy algorithm in the five cases. To implement path planning in the UAV, Noreen [24] uses the Greedy algorithm, which is implemented in Matlab with several polygons as a watering area.

The obstacle avoidance function is one of the functions that must be available in ASV. Many papers discussing obstacle avoidance use A*, RTT, and PRM for autonomous [26] [27] [28]. The experiment of these three methods is done by comparing the performance using simulation. Simulation results show the success of autonomous navigation involving all three methods. The smoothest path is achieved by the A* method. PRM can also closely follow way-points, but ASV tends to move towards one side of the predetermined path. For the RTT method, ASV cannot follow the way-point precisely but can still demonstrate the ability to navigate without collisions.

2.1 Design of Autonomous Raft Vehicle

The Autonomous Raft Vehicle (ARV) frame is built using a double-hull design, which is constructed with a 4-inch PVC pipe as the boat's hull. The two PVC pipes are joined with two aluminium frames to make a raft-like boat. On top of each part of the aluminium frame, BLDC motors are installed with 10 inches propellers used to drive ARV. If the propeller is placed under the hull, it may cause the propeller stuck if the boat is operated in shallow water.

A waterproof controller box is installed between the two aluminium frames containing a microcontroller, batteries, autopilot module, and GPS (see Figure 1. ARV's net weight is measured at about 5.6 kg, and ARV can move at a maximum of 2.5 m/s. The body of our ARV is depicted in Figure 1.

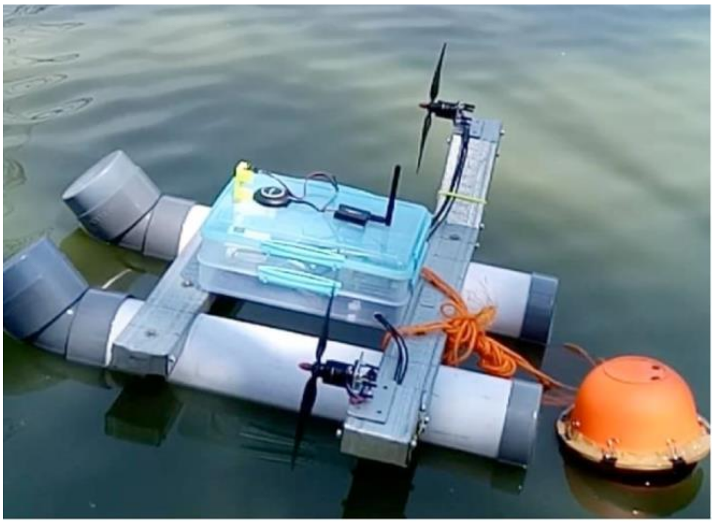


Figure 1. ARV with Double-Hull and Aluminium Frame. Orange Ball is External Sensors to Measure Water Quality

For the autonomous control module, ARV uses ArduPilot Mega 2.8 (APM 2.8) hardware. The choice of APM 2.8 hardware and firmware because it supports way-point navigation. APM 2.8 also supports the Inertial Measurement Unit (IMU), a 3-axis gyro and 3-axis accelerometer, GPS, and magnetometer. Regarding the datasheet, the GPS accuracy is about 2.5 meters. APM 2.8 can only use two modes available in APM 2.8 firmware: manual and auto mode. In manual mode, the ARV is controlled directly using Radio Control, while the auto mode will run autonomously from one way-point to another way-point determined by the user. Auto mode runs closed-loop control the ARV stays on the path between two way-point [28]. However, when the final destination is reached; the ARV will automatically switch from auto mode to hold mode, then the ARV mode automatically changes directly to manually operated. For obstacle avoidance, ARV can directly use some features that already available in APM 2.8 firmware.

The path planning method developed in this research is a modified way-point, i.e. by integrating A* algorithm. Way-points are a series of coordinates, in latitude and longitude, that govern a path used by vehicle to travel from the initial point to a destination in a physical space. The purpose of using the way-point system is that ARV can recognizes and calibrates positions and directions, make corrections in the direction of motion, and improve accuracy in reaching the destination of the route that has been predetermined [29]. In APM 2.8 with Mission Planner software, way-point are executed by entering the earth coordinates, consists of latitude and longitude, to be used as reference trajectory or desired trajectory.

A* is classified as a heuristic search algorithm. This algorithm is based on the Dijkstra and BFS (Best First Search), so this algorithm can be used to find the shortest path that is part of the Dijkstra algorithm function and search for paths using the heuristic function, which is the primary function of BFS [8]. This algorithm is one of the most popular algorithms for solving path planning problems because A* is an optimal algorithm for less complexity and has a lightweight implementation in embedded system controllers [25].

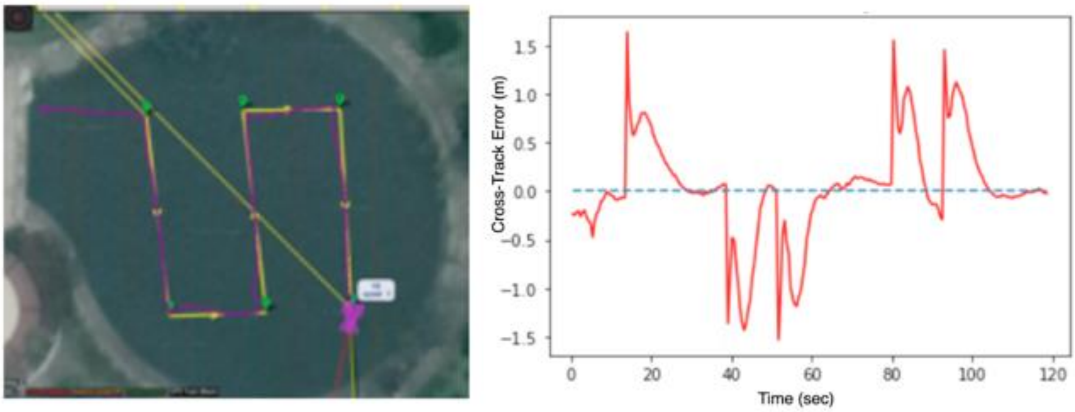
During traveling from an initial position to a goal position, the obstacle avoidance mechanism is crucial to prevent ARV from crashing into objects in front of it. Thus, A* algorithm can also be used to avoid obstacles along with the predetermined path. Implementing A* as an obstacle avoidance is no different from implementing A* for path planning. So A* needs to find a way to reach the goal point while avoiding obstacles. The trajectory search results will then be mapped using latitude longitude and then used as way-points, visualized in the Mission Planner application.

3. Results and Discussion

The experiment aims to test the design of ARV frame and ARV autonomous control during navigation and maneuvering. In general, there are two experimental scenarios, the first is to run standard way-point path planning, and the second scenario is to run the integrated way-point with A* path planning. For these two scenarios, a mission is to reach a goal with six way-points exist in the path, which is to test the ability to maneuver by observing the actual ARV trajectory's error against the desired trajectory. Each scenarios has several parameters that will be adjusted to produce optimal maneuvering.

3.1 Experiment of Standard Way-point Path Planning

The parameters that were adjusted during the experiment are proportional parameters of PID (Proportional Differential Integral controller) steering rate, which is 0.1 and 0.7 for APM 2.8 software environment. We only adjust the proportional parameter since this parameter is related to the fast response of controller, while the error of trajectory is corrected using way-point with A*. The first experiment was carried out by using a proportional gain value = 0.1. This experiment's results can be seen in Figure 2 left side, which is taken from the APM Mission Planner application. In that figure, the yellow line shows the predetermined path while the purple line indicates the actual trajectory of ARV. Based on the trajectory, it can be seen that ARV experiences unstable maneuvers, indicated by the oscillated trajectory when ARV is turning.

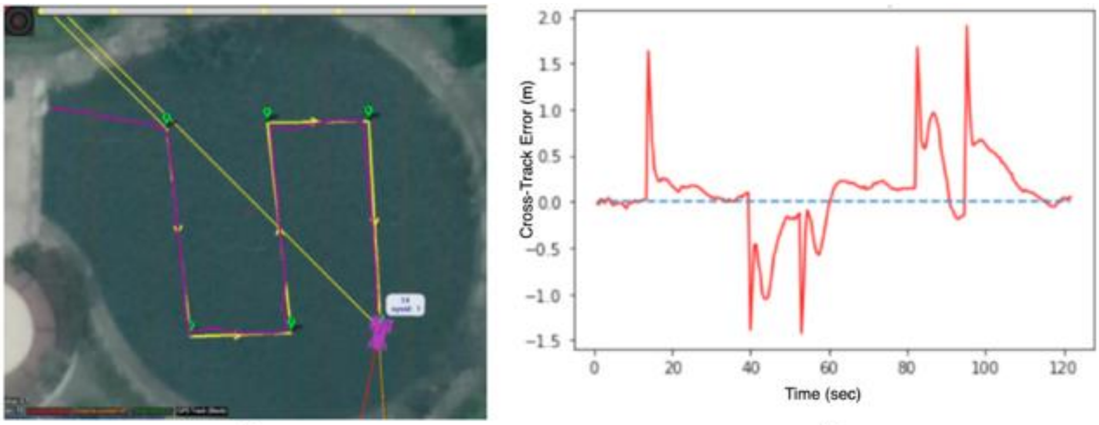


(a) (b)

Figure 2. Left: Trajectory of 6 way-point with Proportional Steering Constant = 0.1. Right: Cross Track Error of Trajectory

The cross-track error parameter is observed to see whether the actual trajectory is aligned with the desired path. Cross-track error is the vehicle's distance (in this case is ARV) to the straight line of the desired path between two way-points. The time-varying cross-track error can be seen in Figure 2 right side, which is plotted using Pyplot. The blue dash line shows the desired trajectory as the reference path. Regarding the plot, it can be inferred that the most significant cross-track error value is 1.636 m, with an average error rate 0.348 m.

The second experiment result with proportional gain value = 0.7 can be seen in Figure 3 (left side). Regarding the plot, it can be seen that the oscillation decreases when the ARV is turning. The cross-track error plot of the second experiment is depected Figure 3 (right side). This red-line shows that the most significant cross-track error value is 1.90 m with an average error rate 0.285 m. Regarding the experiments, the oscillated trajectory when ASV is turning can be reduced by decreasing the steering rate's of proportional controller.



(a) (b)

Figure 3. Left: Trajectory of 6 way-point with Proportional Steering Constant = 0.7. Right: Cross Track Error of Trajectory

3.2. Experiment of Way-point with A*

For this second scenario, the proportional gain value is set to 0.1 experiment uses additional 50 predetermined way-points to perform A* algorithm. The comparison of the result is depected in Figure 4. We also recorded the GPS tracking data for further analysis purposes. Figure 4 left side shows the actual trajectory of ARV using an standard way-point, while Figure 4 right side shows the actual trajectory of the way-point with A* path planning.

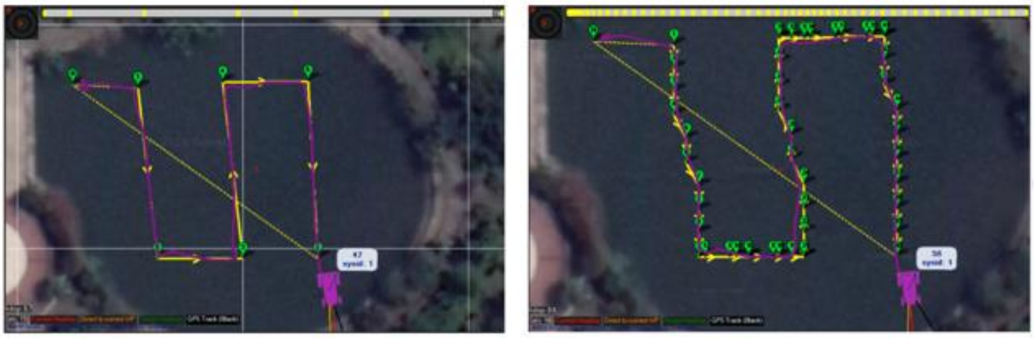

(a) (b)

Figure 4. Trajectory of ASV: (a) original way-point Algorithm (b) way-point with A Algorithm*

In Figure 5 and Figure 6, the black dots indicate the way-points location in latitude and longitude, the blue lines indicate the desired path of the ARV, and the red lines indicate the actual trajectory of ARV. Regarding the recorded data and visualization in Figure 5 (left side), we compute that the desired path length is 117.87 m, while the actual path length of the ARV is 119.28 m. Figure 6 (right side) shows the actual trajectory of ARV using modified way-point with A* algorithm with 50 points. As compared to the standard algorithm in Figure 5 (left side), the desired path of way-point with A* of ARV is 119.50 m, while the actual trajectory length of the ARV is 120.50 m.

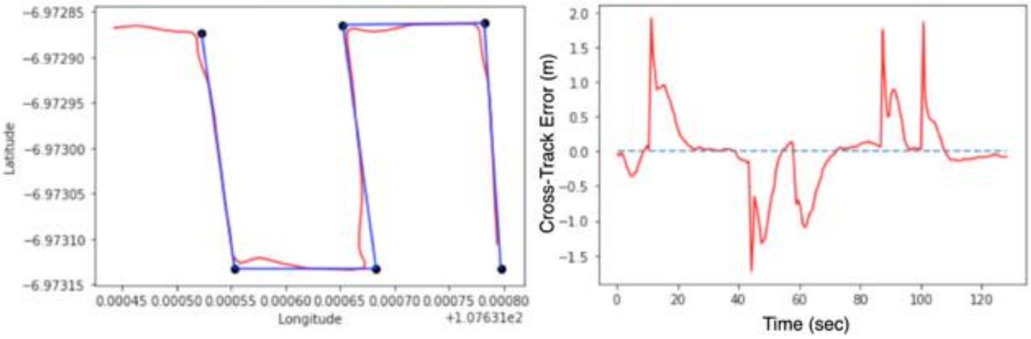


Figure 5 Left: Trajectory of ASV using way-point Algorithm. Right: Cross Track Error

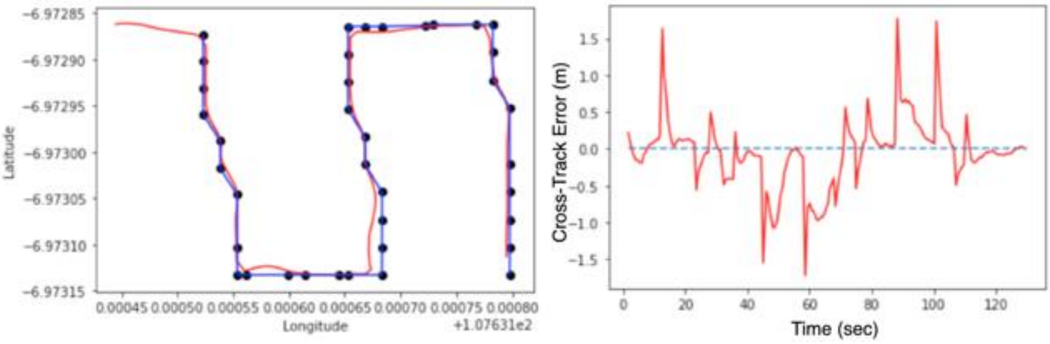


Figure 6 Left: Trajectory of ASV using way-point with A Algorithm. Right: Cross Track Error*

The comparison results of both sanrdard way-point and way-point with A* algorithm can be seen in Table 1**.** Regarding tha Table, it can be inferred that integrating A* into the existing way-point algorithm can reduce the cross-track error of trajectory planning when the vehicle moving from initial position to goal position. Our implementation of way-point with A* algorithm is embedded in APM 2.8 software development kit for autopilot navigation.

Table 1. Comparison of Both Algorithm

Method	No. of Way-point	Desired Path Length (m)	Actual Path Length (m)	Cross-track Error (m)
Without A*	6	17.88	19.28	0.32
With A*	50	19.50	20.50	0.2

Cite: Erfianto, B. (2021). On the Experiment of Path Planning Using Multi-way Points with A Algorithm for Autonomous Surface Vehicle. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(2).* *https://doi.org/10.22219/kinetik.v6i2.1244*

3.3 Experiment of Obstacle Avoidance in Multi Way-point with A*

ARV is equipped with a Maxboatix sonar sensor to detect the obstacle in front of it with a predetermined threshold minimum 2m. Experiments are carried out by placing an obstacle in between two way-points. The obstacle diameter is about 1m. An obstacle avoidance experiment was carried out by using two different scenarios. In the first scenario, ARV performs obstacle avoidance using the standard way-point method without A*, and in the second scenario, ARV performs obstacle avoidance using the way-point with A* algorithm.

The experiment for obstacle avoidance is placing the obstacle in between two way-point. In this experiment, a threshold distance of 2m is given to determine the closest distance between ARV and the obstacle. When the distance between ARV and obstacle reaches at least 2m, the ARV starts to execute an obstacle avoidance algorithm. During the experiments, the ultrasonic sensor reading varies from about 1.03 m to 1.97 m between the ASV and the obstacle. However, using a threshold at 2.5 meters is very risky because there is a possibility that the ultrasonic sensor does not read an obstacle but somewhat interpretes water surface instead. The result of obstacle avoidance using the way-point integrated with A* algorithm can be seen in part (a) of Figure 7.

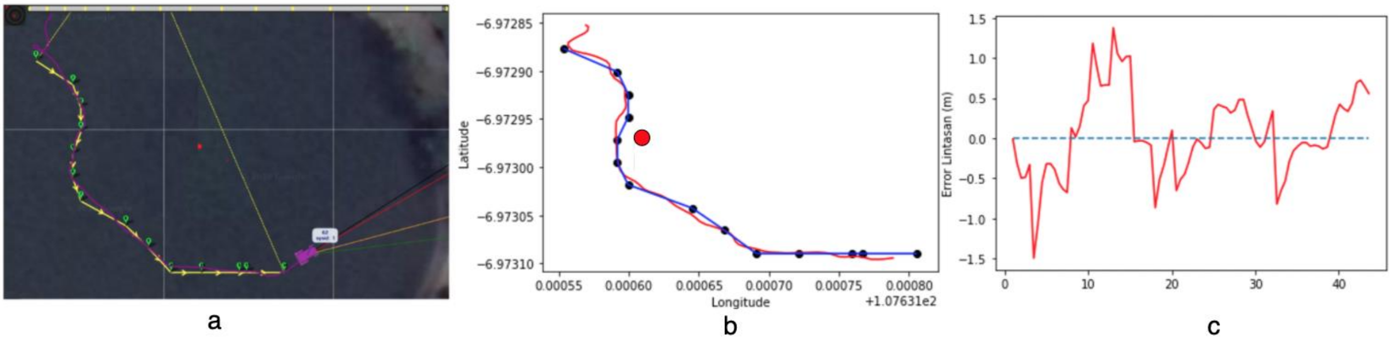


Figure 7 (a): Trajectory of Obstacle Avoidance, Visualized in Mission Planner Software. (b) Trajectory Plot in Python. (c) Cross Track Error of way-point with A Algorithm*

Figure 7 part (c) shows the plot of measurement of cross-track errors when the ARV is traveling with obstacle avoidance algorithm using way-points with A* algorithm. The blue line shows the desired path, and the red line shows the actual path ARV travelled. Regarding trajectory depicted in Figure 7 (a) and (b), the minimum error is about 0.02m, with the maximum error is 1.45m, and the average error is 0.45 m. If obstacle avoidance algorithm is executed using standard way-point, the minimum error is a bit larger, i.e. 0.3m, with a maximum error of 1.9 m and an average error of 0.56m. The results of the obstacle avoidance experiment can be seen in Table 2. Regarding the actual trajectory, the way-point with A* has a smaller actual path length. The average cross-track error of way-point with A * is smaller than the standard way-point regarding the error measurement results. It can be inferred that the way-point method with A* algorithm converges faster to the desired trajectory than the standard way-point after executing obstacle avoidance.

Table 2. Comparison of Path Planning Eith Obstacle Avoidance

Method	No. of Way-point	Desired Path Length (m)	Actual Path Length (m)	Average Cross-track Error (m)
Without A*	6	11.2	20.26	0.56
With A*	15	11.2	18.60	0.45

4. Conclusion

The research has successfully developed an ARV prototype, the Autonomous Raft Vehicle, that functions as an autonomous surface vehicle, especially in lake waters. From the experiment results, when the ARV maneuvers, the average cross-track error on the straight path is 0.088m, while on the curved path, the cross-track error is obtained with an average value of 0.285 m. The most significant cross-track error is obtained when the ASV was turning, which is 1.90m. This research also succeeded in implementing the way-points with A* algorithm for ARV path planning. In general condition, the experiment results show that the original way-point has better performance concerning the path length than the way-point method with A* algorithm. It is because the original way-point implementation is straightforward than the way-point method with A*. However, all experiments show that the integration of A* algorithm could reduce cross-track error than the original way-point. The ARV can avoid obstacles with the closest distance between the ARV, and the obstacle is 1.43m. The experiments also show that obstacle avoidance using an integration of way-point with A* is faster in terms of path length than standard way-point. In terms of cross-track error, integration of way-point with A* also has less cross-track error than the original way-point. Therefore, integrating A* into way-point is better for path planning correction by reducing cross-track error when the vehicle is doing obstacle avoidance.

References

[1] G. Ferri, A. Manzi, F. Fornai, F. Ciuchi, and C. Laschi, “The hydronet asv, a small-sized autonomous catamaran for real-time monitoring of water quality: From design to missions at sea,” *IEEE Journal of Oceanic Engineering*, Vol. 40, No. 3, Pp. 710–726, 2014. https://doi.org/10.1109/JOE.2014.2359361

[2] S. Siyang and T. Kerdcharoen, “Development of unmanned surface vehicle for smart water quality inspector,” in *13th International conference on electrical engineering/electronics, computer, telecommunications and information technology (ECTI-CON)*, Pp. 1–5, 2016. https://doi.org/10.1109/ECTICon.2016.7561370

[3] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, “Unmanned surface vehicles: An overview of developments and challenges,” *Annual Reviews in Control*, Vol. 41, Pp. 71–93, 2016. https://doi.org/10.1016/j.arcontrol.2016.04.018

[4] H. C. Brown, L. K. Jenkins, G. A. Meadows, and R. A. Shuchman, “Bathyboat: An autonomous surface vessel for stand-alone survey and underwater vehicle network supervision,” *Marine Technology Society Journal*, Vol. 44, No. 4, Pp. 20–29, 2010. https://doi.org/10.4031/MTSJ.44.4.5

[5] Y. Liu, N. Noguchi, and T. Yusa, “Development of an unmanned surface vehicle platform for autonomous navigation in paddy field,” *IFAC Proceedings*, Vol. 47, No. 3, Pp. 11 553–11 558, 2014. https://doi.org/10.3182/20140824-6-ZA-1003.00616

[6] M.W. L. Thein and M. Renken, “Comparative study of heading and speed control techniques for small autonomous surface vehicles,” in *IEEE OCEANS 2018*, Pp. 1–10, 2018. https://doi.org/10.1109/OCEANS.2018.8604777

[7] G. Podnar, J. M. Dolan, K. H. Low, and A. Elfes, ”Telesupervised remote surface water quality sensing,” in *IEEE Aerospace Conference*, Pp. 1–9, 2010. https://doi.org/10.1109/AERO.2010.5446668

[8] M. Lin, K. Yuan, C. Shi, and Y. Wang, “Path planning of mobile robot based on improved A* algorithm,” in *29th IEEE Chinese Control And Decision Conference (IEEE CCDC)*, Pp. 3570–3576, 2017. https://doi.org/10.1109/CCDC.2017.7979125

[9] Y. Kaizu, M. Iio, H. Yamada, and N. Noguchi. “Development of unmanned airboat for water-quality mapping,” *Biosystems Engineering*, Vol. 109, No. 4, Pp. 338–347, 2011. https://doi.org/10.1016/j.biosystemseng.2011.04.013

[10] H. M. Ferreira, C. Almeida, A. Martins, J. M. Almeida, A. Dias, G. Silva, and E. Silva, “Environmental modeling with precision navigation using roaz autonomous surface vehicle,” in *IEEE International Conference on Intelligent Robots and Systems*, Pp. 1–6, 2012. http://dx.doi.org/10.13140/RG.2.1.1213.0325

[11] Volker Bertram, “Unmanned surface vehicles-a survey”, *Skibsteknisk Selskab Copenhagen Denmark*, Vol. 1, pp. 1–14, 2008. https://doi.org/10.1.1.462.1894

[12] Mohamad Hilmi Bin Mat Idris, Muhammad Al Azhari Bin Che Kamarudin, Muhammad Irsyad Sahalan, Zulkifli Bin Zainal Abidin, and M.M. Rashid, “Design and development of an autonomous surface vessel for inland water depth monitoring,” in *International Conference on Computer and Communication Engineering (ICCCE)*, Pp. 177–182, 2016. https://doi.org/10.1109/ICCCE.2016.47

[13] D. Machado, A. Martins, J. M. Almeida, H. Ferreira, G. Amaral, B. Ferreira, A. Matos, and E. Silva, “Water jet based autonomous surface vehicle for coastal waters operations,” in *IEEE 2014 Oceans-St. John’s*, Pp. 1–8. http://dx.doi.org/10.1109/OCEANS.2014.7003169

[14] P. Ramos, N. Cruz, A. Matos, M. V. Neves, and F. L. Pereira, “Monitoring an ocean outfall using an AUV,” in IEEE OCEAN Conference and Exhibition, Vol. 3, Pp. 2009–2014, 2001. http://dx.doi.org/10.1109/OCEANS.2001.968154

[15] M. A. Z. Fuad, A. B. Sambah, A. Isdianto, and A. Andira, “Pemetaan batimetri sebagai informasi dasar untuk penempatan fish apartment di perairan Bangsring Kabupaten Banyuwangi Jawa Timur,” *DEPIK Jurnal Ilmu-Ilmu Perairan, Pesisir dan Perikanan*, vol. 5, no. 3, 2016. https://doi.org/10.13170/depik.5.3.5655

[16] R. Arulmozhiyal and K. Baskaran, “Implementation of a Fuzzy PI Controller for Speed Control of Induction Motors Using FPGA,” *Journal of Power Electronics*, Vol. 10, Pp. 65-71, 2010. http://dx.doi.org/10.6113/JPE.2010.10.1.065

[17] Bandini, D. Olesen, J. Jakobsen, C. M. M. Kittel, S. Wang, M. Garcia, and P. Bauer-Gottwein, “Bathymetry observations of inland water bodies using a tethered single-beam sonar controlled by an unmanned aerial vehicle,” *Hydrology and Earth System*, Vol. 22, No 6, Pp. 4165-4181, 2017. http://dx.doi.org/10.5194/hess-2017-625

[18] M. Dunbabin and L. Marques, “Robots for environmental monitoring: Significant advancements and applications,” *IEEE Robotics & Automation Magazine*, Vol. 19, No. 1, pp. 24–39, 2012. https://doi.org/10.1109/MRA.2011.2181683

[19] G. Hitz, F. Pomerleau, M.-E. Garneau, C. Pradalier, T. Posch, J. Pernthaler, and R. Y. Siegwart, “Autonomous inland water monitoring: Design and application of a surface vessel,” *IEEE Robotics & Automation Magazine*, Vol. 19, No. 1, Pp. 62–72, 2012. http://dx.doi.org/10.1109/MRA.2011.2181771

[20] Denny, T. O’Brien, E. Bergeron, D. Twichell, C. Worley, W. Danforth, B. Andrews, and B. Irwin, “Advances in shallow-water, high-resolution seafloor mapping: Integrating an autonomous surface vessel (asv) into nearshore geophysical studies,” in *AGU Fall Meeting Abstracts*, 2006.

[21] A. Tuncer and M. Yildirim, “Dynamic path planning of mobile robots with improved genetic algorithm,” Computers & Electrical Engineering,” Vol. 38, No. 6, Pp. 1564–1572, 2012. https://doi.org/10.1016/j.compeleceng.2012.06.016

[22] M. J. J. Suja, S. R. Sulistiyanti, and M. Komarudin, “Sistem navigasi pada unmanned surface vehicle untuk pemantauan daerah perairan. Electrician”, Vol. 11, No. 1, Pp. 32–43, 2017. https://doi.org/10.23960/elc.v11n1.2013

[23] M. Bibuli, G. Bruzzone, M. Caccia, E. Fumagalli, E. Saggini, E. Zereik, E. Buttaro, C. Caporale, and R. Ivaldi, “Unmanned surface vehicles for automatic bathymetry mapping and shores’ maintenance,” *in IEEE OCEANS 2014-TAIPEI*, Pp. 1–7, 2014. https://doi.org/10.1109/OCEANS-TAIPEI.2014.6964440

[24] I. Noreen, A. Khan, and Z. Habib, “A comparison of rrt, rrt* and rrt*-smart path planning algorithms,” *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 16, No. 10, Pp. 20, 2016. http://dx.doi.org/10.5772/56718

[25] K. Sinha, “Path planning for a uav in an agricultural environment to tour and cover multiple neighborhoods,” *MSc. Thesis*, Virginia Tech, 2017.

[26] H.Pan, C.Guo, and Z. Wang, “Research for path planning based on improved A* algorithm,” in *4th IEEE International Conference on Information, Cybernetics and Computational Social Systems (ICCSS),* pp. 225–230, 2017.

[27] A.J. Dalpe and M.-W. Thein, “Obstacle avoidance strategies for autonomous surface vehicles,” in IEEE OCEANS-Anchorage, pp. 1–8, 2017.

[28] A.J. Dalpe, A.E.Cook, M.-W.Thein, and M.Renken, “A multi-layered approach to autonomous surface vehicle map-based autonomy,” in IEEE *OCEANS 2018 Charleston*, Pp. 1–8, 2018. https://doi.org/10.1109/OCEANS.2018.8604602

[29] S. Park, J. Deyst, and J. How, “A new nonlinear guidance logic for trajectory tracking,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, p. 4900. 2004. http://dx.doi.org/10.2514/6.2004-4900

Cite: Erfianto, B. (2021). On the Experiment of Path Planning Using Multi-way Points with A Algorithm for Autonomous Surface Vehicle. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(2).* https://doi.org/10.22219/kinetik.v6i2.1244