Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi
Vol. 11, No. 3, Juli 2022

ISSN 0216 – 0544
e-ISSN 2301– 6914

DEVELOPMENT OF AUTONOMOUS UNDERWATER VEHICLE (AUV) BASED ON ROBOTIC OPERATING SYSTEM FOR FOLLOWING UNDERWATER CABLE

[a]Ronny Mardiyanto, [b]Haidar Semadhi Aji, [c]Astria Nur Irfansyah, [d]Heri Suryoatmojo

[a, b, c, d] Department of Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology, Sepuluh Nopember Institute of Technology (ITS)
Surabaya, East Java, Indonesia. Postal code: 60111
E-mail: ronny@elect-eng.its.ac.id, haidarsemadhi99@gmail.com, irfansyah@ee.its.ac.id

Abstract

In addition to satellites, the Marine Cable Communication Channel (SKKL) which is located under the sea is also one of the backbones of the communication network to connect from one island to another. However, there are often irresponsible parties who damage or commit acts of vandalism. This action resulted in the disruption of the communication process on the submarine cable network. So, it is necessary to periodically check the condition of the underwater communication network cables. Regular checking of underwater cables is very risky, so we propose an underwater robot to handle it. This paper presents the development of Autonomous Underwater Vehicle (AUV) based on Robotic Operating System (ROS) for Following Underwater Cable to monitor the condition of the cables automatically. The AUV is equipped with a camera, Nvidia Jetson Nano, Arduino, Flight Controller, ESC, and brushless DC motors that used to assist the tracking process on the cable. The camera is used as the main visual sensor. Visual image processing methods are carried out using thresholding and contours detection methods, then the obtained data are processed to drive the motors on the AUV so that they can move on the direction of the cable. The experiment results show that the object detection method can be used under conditions with light intensity more than 25 lux. It works optimally at speeds of 0.27 m/s to 0.42 m/s. In the horizontal motion control test, the overshoot parameter value is ±60%, rise time is 2s, settling time is 16s, and steady state error is ±20%. The AUV can track on a straight and winding path of 2 meters with a bright light intensity of ±493 lux, a dim light of ±107 lux and a dark light intensity of ±25 lux with the help of an LED beam with a light intensity of ±773 lux. The percentage of success of scoping experiments on a straight track and a winding track with three trials is 75%. This performance shows that the developed AUV works well to follow underwater cable.

Key words: Marine Cable Communication Channels; AUV; thresholding; contours detection.

INTRODUCTION

The world has experienced a communications revolution since 1988 when the first transoceanic fiber optic cable was installed connecting the UK, the United States and France. This is the result of a tremendous technological push and market appeal. Currently, network connectivity is highly dependent on underwater fiber optic cables, which later became something very important in our lives, given our social and economic dependence on the internet. Therefore, cable damage is very important to be detected immediately. Statistics show that about 70% of the total damage to underwater fiber optic cables is the result of external aggressions which are mainly related to human activities. Damage to underwater fiber optic cables occurs in shallow water depths of less than 200 m, due to fishing activities and ship shipping. Apart from that, the damage caused by natural disasters amounted to less than 10% of all damage. The damage occurs in both deep and shallow waters [1], [2].

The potential for damage to submarine cables is even greater if they are at sea with heavy ship traffic. As many as 75% are caused by illegal ship activities, one of which is fishing and stops in the submarine cable area [3]. Number of fish fishermen still use the C*antrang* method to find fish. This method has the potential to damage submarine cables because the depth of the C*antrang* that is spread is sometimes the depth near the underwater communication cable [4]. In addition to fishing activities, theft of submarine cables is also still common and becomes an obstacle. based on Law no. 36/99 concerning Telecommunications, the sanction given to the perpetrator is a criminal sanction of imprisonment for 5 years [5]. In addition, the danger of laying anchors carelessly also endangers other underwater cables such as gas pipelines and electricity pipelines [6].

The existence of traditional fishing boats, which are difficult to trace when crossing the SKKL, makes it difficult to detect possible damage to their location and whereabouts. Therefore, this project was created as a tool to carry out maintenance of the Marine Cable Communication Channel (SKKL) that stretches in the Indonesian sea. This tool is in the form of an Autonomous Underwater Vehicle (AUV) which is able to track cables automatically and is able to detect damage that occurs through the camera. The AUV is equipped with a camera sensor that is able to detect underwater cables, which are then processed to be able to move the AUV motor automatically to follow the direction of the cable. And then the damage that occurs can be detected by the camera which then the data is sent directly to the maintenance ship [7], [8].

The main objective of the proposed system is developing AUV for regularly check underwater cables automatically based on visual sensor. The proposed system has advantages on the performance for following underwater cable automatically.

The following are researches that have been carried out related to the development of AUV technologies.

Simple vision tracking of pipelines for an autonomous underwater vehicle has been developed by ref [9]. It used 2D vision-based pipeline tracking system for Autonomous Underwater Vehicles (AUV) using a model-based recognition approach. By extracting a square image from a 2D camera and with the help of sonar input and a compass that is used to guide the AUV to track the pipeline.

Wire Recognition in Image Within Aerial Inspection Application has been developed by ref [10]. To determine the position of the power line and estimate the orientation of the cable in the image, it used the Gaussian elliptical function. This stage is followed by canny edge detection. The simulation results show that the estimated range of allowable angular error is about $\pm 12^{\circ}$.

High speed Automatic Power Line Detection and Tracking for a UAV-Based Inspection has been developed by ref [11]. The Hough transform is used to extract line segments from power lines. According to the power line characteristics of the aerial captured image, they used K-means in the Hough space to group and filter the straight lines, and then detect the power lines. Lastly, a Kalman filter is used to track power lines in the Hough space, based on the continuity of the video sequence. The experimental results show that the proposed approach is effective in detecting and automatically tracking power lines in complex environments.

Power Line Recognition and Tracking Method for UAVs Inspection has been developed by ref [12]. The power line is improved by bilateral linear filtering. Then, the Hough transform is used to detect power lines. Finally, in the sequential figure, the state of the power line is estimated, and the power line is tracked with that estimate. Tracking can reduce computation time. To verify the performance of the proposed method, experiments were carried out on real image data taken from the UAV. The results show that the proposed method is computationally time efficient but has a low success rate of power line recognition.

Robust Real-Time UAV Based Power line Detection and Tracking has been developed by ref [13]. it used a new method that can effectively adjust the threshold in the detection algorithm. First, a Region of Interest (RoI) was selected for further analysis and tracking of power lines. Then the kernel improved on edge detection and hysteresis thresholding in the Canny approach. Then each point in the image is mapped to the appropriate physical space. Through this mapping, the UAV can obtain the corresponding coordinates for all patches. Next, the Hough Transform is used to filter out irrelevant linear structures. Then, a new power line model is reconstructed based on the power lines and power poles detected in all patches. Finally, move to the next frame and repeat the same process from the beginning. The results show that the method in this paper has a better performance than the manual Canny and Hough transformation, especially on complex backgrounds.

Extra Matters Recognition of Transmission System Based on Hough Transform has been developed by ref [14]. The transmission line detection system in the image is based on the Hough Transform and on the detection is analyzed if there are additional problems on the line. Before the detection of the transmission line, there are several stages of image processing. First, the RGB image color is converted to gray. After that, a spatial smoothing filter is used to improve the image quality. Tests were carried out on 150 samples. And as a result, the identification error rate is less than 15%.

Power Line Detection Via Background Noise Removal has been developed by ref [15]. The background noise is reduced by classifying the Convolution Neural Network (CNN) before extracting the power line. The proposed method operates in three steps: 1) extracting power line edge features from the image, 2) using CNN classifier to remove background noise, 3) using Hough-Transform (HT) to locate power lines. In this paper, the test is carried out by comparing the proposed method with 2 other methods that have been widely used, namely the Gaussian Model (GM) and Spatial Contexts (SC). The test involves 3 performance indices, namely Precision Rate (PR), Missing Rate (MR) and False Alarm Rate (FAR). As a result, the proposed method is superior to the index precision rate and false alarm rate compared to the GM and SC methods.

Automated Guided Vehicle Using Robot Operating Systems has been developed by ref [16]. It used an embedded controller integrated with the Robotic GUI operating system, which can be used to carry industrial goods and materials, public services, etc. AGV (Automated Guided Vehicle) is built with the latest sensors such as Oblu which is an IMU sensor, Kinect and software for parallel localization and mapping. This robot software section is mainly implemented using Python programming language and software frameworks, such as Robot Operating System (ROS), Open-CV, Open-Ni, etc. The control system in question includes self-localization, automatic search for accurate paths, and motion control.

The difference between the research that has been done with this research is on the object to be detected. Other studies detect objects such as pipelines and powerlines. While in this study is detecting an underwater communication. In addition, the robot that used is also different, in other studies is using UAV while in this study is using AUV. These two tools are clearly very different in their use. UAV is used for purposes in the land sector while AUV is used in the underwater sector. The cable detection method that used is also different. The other studies is using canny edge detection method, Kalman filter, Hough transform and some are using the Convolution Neural Network (CNN) method. In this study, the method is used a combination of thresholding and contour detection methods.

MATERIAL AND METHODS

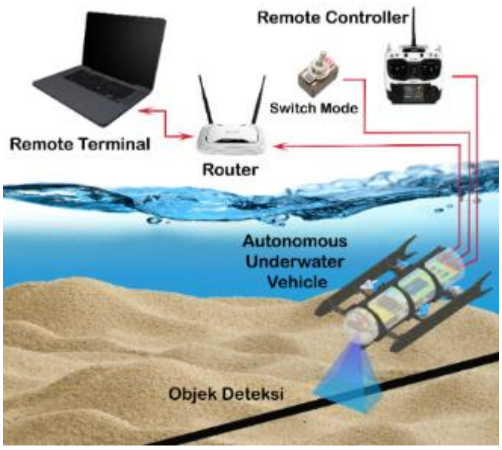


Fig 1. Schematic and illustration of AUV with cable tracking system

The following functional specifications of the AUV:

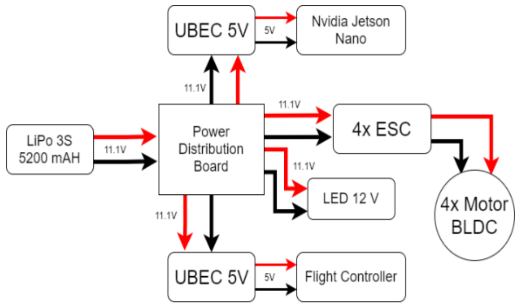
1. The AUV dives manually using a remote control, the AUV is controlled manually

until it finds the starting point of the underwater cable tracking automatically.

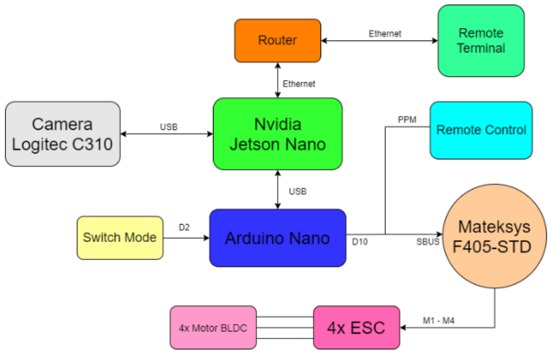
2. After the point is found, the Switch mode is switched to autonomous mode so that the AUV can move automatically along the underwater cable.

Electronic and Mechanical System Design

The AUV is designed to be able to dive dynamically, automatically detecting and tracking underwater cables. This AUV design is included in the Observation Class AUV category which is specifically designed for light use only with a propulsion system to carry camera and sensor packages to where useful images or data can be taken [17]. This drone uses a flight controller with the type MatekSys F405 – STD. Before being used, MatekSys F405 – STD needs to be set in advance the firmware used and several other parameters using the BetaFlight Configurator software. The firmware is adjusted to the type of drone used, namely ArduSub SimpleROV4 specifically for the use of ROV 4 motors (Thruster). The flight controller is used to read signals from Arduino Nano and Remote Control in the form of PPM signals, then give a signal to the ESC to control the speed of each motor. Arduino Nano is used to input PPM signals which are controlled serially using the ROS system found on the Nvidia Jetson Nano. This AUV design has two PPM input signal sources, sourced from Arduino Nano for automatic mode and Remote Control for manual mode. The two signals are connected in parallel to the flight controller. So that the two signals do not collide with each other, Switch Mode is used which serves to determine which signal is needed to enter the flight controller. This AUV uses a 3S 5200 mAH LiPo battery as its main power source with a voltage of 11.1 V. LiPo batteries are used to supply 4 ESCs, 2 12V LEDs, and 2 5V UBECs. The first 5V UBEC is used to supply the Nvidia Jetson Nano and Arduino Nano microcontrollers, while the second 5V UBEC is used to supply the Flight Controller. To find out the electronic scheme on the AUV, it can be seen in Figure 2.



(a)



(b)

Fig 2. Electronic schematic on AUV, (a) power distribution, (b) Data connection

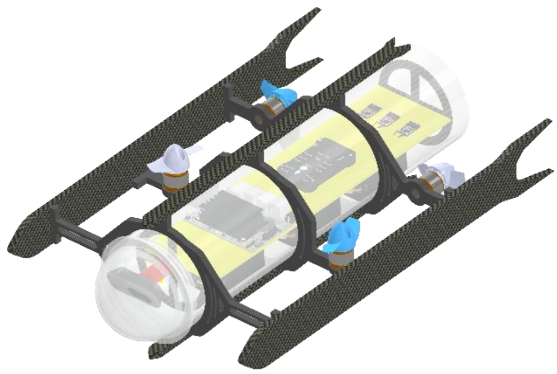


Fig 3. 3D modeling design using autodesk inventor software

The electronic system on this AUV is equipped with an underwater cable detection system using a Logitech c310 webcam camera. Images from the webcam are processed using the OpenCV library, an open-source library that focuses on simplifying programming related to digital images.

The visual image processing data is then published into the ROS system which will later be used to obtain output data in the form of a PPM signal issued by the Arduino to the flight controller in automatic mode.

In accordance with the type and classification, this AUV has specifications that are specifically used only for Observation. This AUV has a simple body designed specifically for light use only with a propulsion system to carry the camera and sensor packages to places where useful images or data can be captured. This AUV body only has one main component, namely an acrylic tube in which there is an electronic system package including sensors and cameras. On the right and left there are frames that function as holders for the four thrusters, the frame is also designed like a fish fin which aims to have good aerodynamics so that the AUV is able to dive and move dynamically and balanced in the water.

In the thruster section, it uses a brushless waterproof type of motor with a low KV so that the motor has a high torque, very suitable for underwater use so that it can drive a propeller that requires high torque to break up water currents to get a strong thrust. The motor is combined with the CW CCW 2 blade propeller, the two blades propeller has the characteristics of producing high speed with low torque, so it is very suitable for observation class AUVs that require agility with light weight only carrying a camera and sensor package so it does not require high torque to move.

This AUV can be operated manually using a remote controller using a PPM signal that is connected directly using a cable to the flight controller, while the automatic operation of the AUV is controlled based on the results of visual image processing obtained from camera captures in the form of Bisyal data which is then processed using the OpenCV library. then the data is processed again in the ROS system. The data is then published serially to Arduino to get output in the form of a PPM signal which is sent to the flight controller via Arduino digital pins. So that the PPM signal from the Arduino and the remote controller do not collide with each other, a switch is added which functions to regulate when the PPM output from the Arduino needs to be issued. This switch is called a mode switch. This switch is in the form of a toggle switch that is connected to the Arduino digital pin, whose data is published to the ROS system for processing so that it can be used to regulate when the PPM signal from the Arduino needs to be issued. The following is a flowchart of the work system on the AUV.

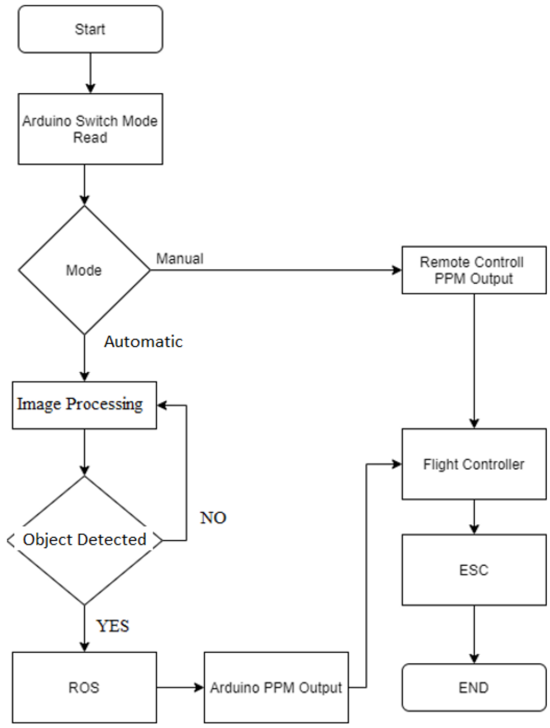


Fig 4. Control system flowchart

Detection System Design

Image processing is done using a Logitech C310 camera. This camera has a resolution of 1.3 MP and video with a maximum resolution of 720p at 30fps. In this image processing process, the incoming image is set so that it has a resolution of 640 x 480 due to lighten the processing load carried out by the Nvidia Jetson Nano.

After changing the image resolution, the next process converts the RGB color space format captured by the camera into a gray format. This is because the RGB color space of each of its constituent components (Red, Green, and Blue) is affected by the amount of light hitting the object, making the color selection process more difficult. By using the gray image becomes a black and white bar so that the thresholding process can be carried out more easily, the color of the image becomes easier to select. To further refine the image, the blurring process is carried out to get a smooth image [18].

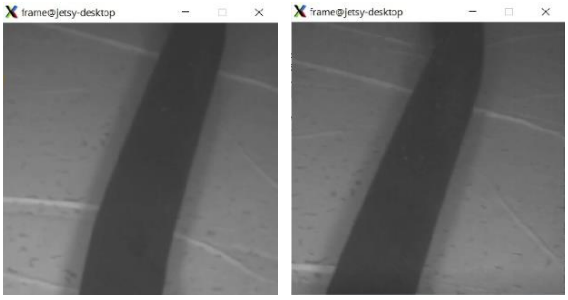


Fig 5. The results of the process of converting RGB to gray format (left) and the results of the blurring process (right)

Next, the color thresholding process is carried out on the image that has been converted into gray format to separate the desired color from the background image. The result of this process is a binary format image where pixels that have the same color as the threshold value will be worth 255 (white) while the others will be worth 0 (black) as shown in the following image [19].

Fig 6. Thresholding process results

After the threshold is obtained, the next process is to find the contour. Contour is a collection of points that represent a curve that has the same color/intensity in the image. How to represent this curve in OpenCV with a vector where each entry in the vector encodes information about the location of the points that make up the curve. Looking for a contour is like looking for a white object on a black background. So that in order to easily find a contour in an image, the image needs to be converted first into a binary image which is the result of the thresholding process [20].

After the contours data is found, the data is re-selected to get the area with the maximum contours value (the widest area) using one of the Contours features, namely cv2.contourArea. In addition to the contour area feature, the contour function has a moments feature (cv2.moments) with this feature we can find the midpoint data of the widest area obtained from the x-axis and y-axis.

In this case, the data needed is x-axis data because the robot only needs data values to the right and left of the midpoint. The data can be used for the robot navigation process. From the object detection process, 2 parameters are obtained, namely the status of the object which represents the presence or absence of the object and the position of the midpoint of the contour area on the x-axis and y-axis.

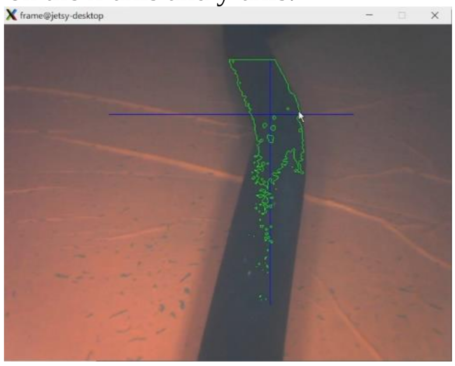


Fig 7. The final result of the cable detection process

ROS Framework Design

The main framework used in this robot uses ROS (Robot Operating System) which is a software framework in which there is a collection of programs and libraries that are integrated with each other which really helps to simplify programming [21]. In this AUV, the ROS is used to communicate between programs, from the detection program to the PPM output program issued by the Arduino. In this ROS framework, there are 4 nodes and 4 topics that have different functions. Here is the ROS map used in this system.

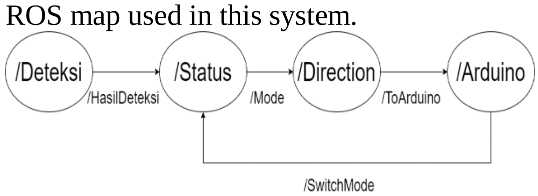


Fig 8. ROS map diagram

The first node is the /Deteksi node which contains an object detection publish program that publishes its data through the /Deteksi Result topic to the /Status node. Next is the /Status node, this node contains the publish and subscribe programs that function to read then combine and publish data to the next node. This node subscribes to data from two topics, namely /ResultDeteksi which comes from the /Deteksi node and subscribes data from the /SwitchMode topic which comes from the /Arduino node. The program algorithm on this node is if the switch reads 1 then the node will publish data in the form of an integer obtained from the /Detect node, then if the switch reads 0 then the node will publish data in the form of an integer, in this case the author programmed the node to publish the number 1000 so that later the data easy to compare and process at the next node. The following is a programming algorithm flowchart on the /Status node.

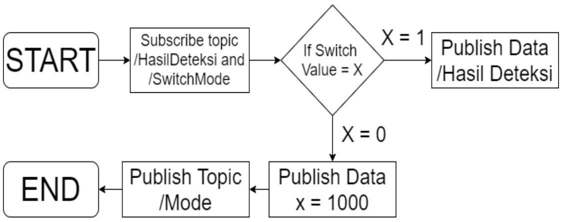


Fig 9. Node /status algorithm

The next node is the /Direction node. This node contains subscribe and publish programs that serve to simplify and determine the direction of the robot's motion. This node subscribes to the topic /Mode from the previous node, namely /Status. The following is the program algorithm on the /Direction node.

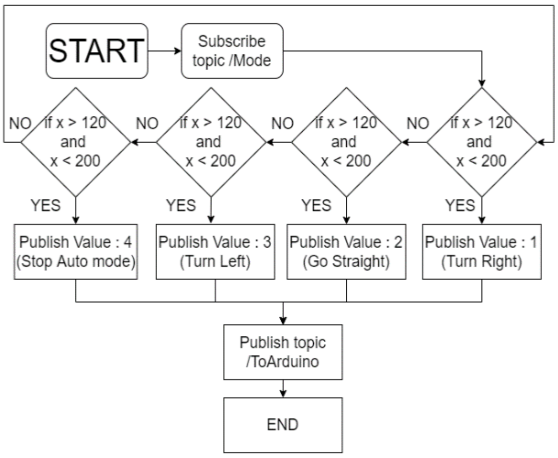


Fig 10. Node /direction algorithm

The /Direction node then publishes data to the /Arduino node with the /ToArduino topic. The /Arduno node also contains subscribe and publish programs. The /Arduino node subscribes to the /ToArduino topic whose data is then processed and used to determine the PPMout signal output value. In addition to subscribing, Arduino also publishes data from reading switches with the topic /SwitchMode whose data is used to determine whether the robot is in manual or auto mode. The complete Arduino system will be discussed in the next sub-chapter

Arduino System Design

The main component of the AUV automatic motion system is Arduino. Arduino is tasked with issuing an output in the form of a PPM signal to the flight controller. The PPM output that is issued is controlled based on the input signal originating from a node in the ROS system which has been discussed in the previous sub-chapter. In this case, Arduino acts as a subscriber that receives data and then it is processed into an output signal that is issued via Arduino digital pins.

Apart from being a subscriber, the Arduino ROS system also serves as a publisher where Arduino publishes reading data from the mode switch to be used as an indicator of whether the robot operates manually or automatically. Here is the Arduino programming algorithm as a Publisher and Subscriber.

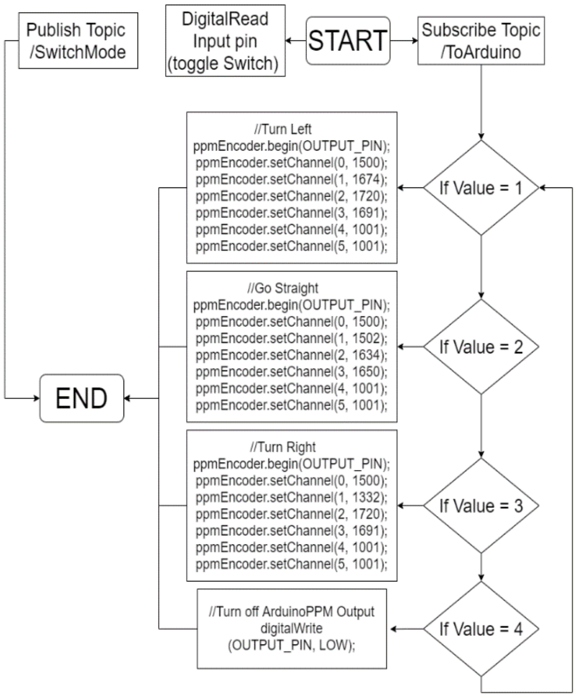


Fig 11. Arduino program algorithm

RESULT AND DISCUSSION

In this chapter, some experiments that have been carried out and their results will be presented which aims to determine/measure the ability of the AUV made. The experiments carried out include Object Detection Test based on AUV motion speed, Object detection test based on light intensity, Object detection test based on level of water turbidity, Object detection test based on water current disturbance, etc.

Object Detection Test

This test is carried out to observe the object detection process with visual image processing methods in several conditions that may occur when the AUV dives in the automatic tracking process. This aims to determine the reliability of the method in the process of detecting objects from optimal to non-optimal conditions.

Figure 12 shows the results of the object detection process properly and perfectly with thresholding and contours detection methods. The selection of the right color threshold value in the gray image and good and stable lighting causes object detection results to run optimally. The position of the object can also be known accurately.

Fig 12. Detection of objects under optimal conditions

Furthermore, the object detection process in conditions that are not optimal. This condition is caused by poor lighting conditions, so that the threshold is not able to distinguish black and white in the gray color format.

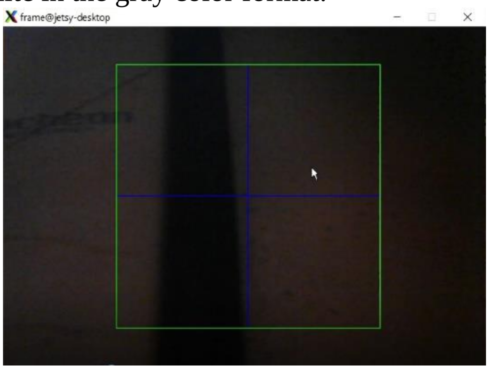


Fig 13. Detection of objects in non-optimal conditions

In this experiment, the accuracy level of the object detection system was also observed with 4 different lighting conditions, namely when there is enough light, low light, dark light and dark lighting conditions with the help of an LED highlight. The three conditions were tested at the same place at different times to obtain different light intensities from the sun. The test was carried out in an outdoor pool with an area of 2 meters x 3 meters with a depth of 1 meter. This test is carried out by running the robot automatically at a depth of approximately 50 cm.

Here is the first test, namely conditions when there is enough light. This test was carried out at 16.00 WIB, when the sun was still bright. The results of this test, object detection looks very accurate. Threshold is able to distinguish black and white colors well.

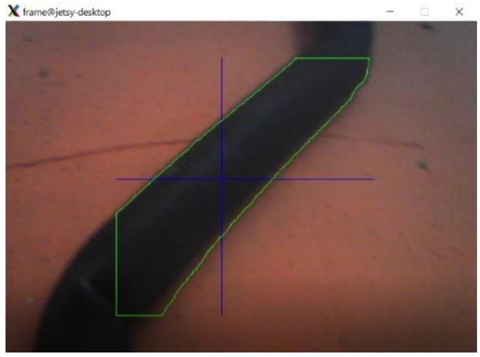


Fig 14. The results of testing the detection program with sufficient lighting parameters

Next is testing when conditions are low light. This test was carried out at 17.00 WIB where the sunlight began to dim. The results of this test object detection look less accurate, there is a lot of noise around the object, this is due to the threshold starting to have difficulty distinguishing black and white colors.

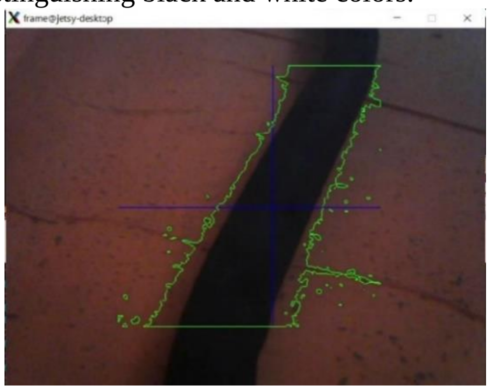


Fig 15. Test results of detection programs with low lighting parameters

The third test is a test when the light conditions are dark. This test was carried out at 18.00 WIB where the sun had dimmed. The results of this test object can no longer be detected. Threshold is no longer able to distinguish between black and white.

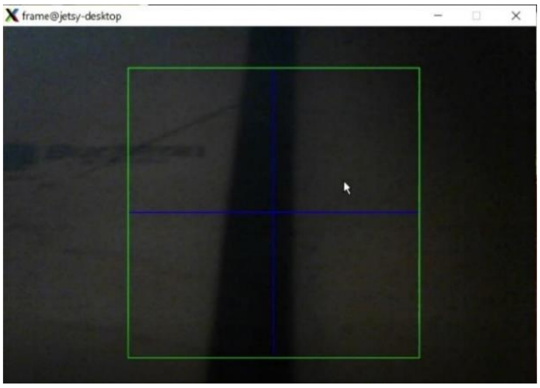


Fig 16. Test results of detection programs with dark lighting parameters

The last test is a test when the light conditions are dark with the help of a 12 V LED lighting beam. This test is carried out at 18.00 WIB where the lighting conditions around are dim. The results of this test can detect objects again, but there is a lot of noise and parts of objects that are close to the 12 V LED cannot be detected. This is because the LED lighting is too bright, so the threshold is difficult to distinguish between black and white. However, at the border of the 12V LED range, it can be detected, this is because the object gets the right lighting, not too little and not too much.

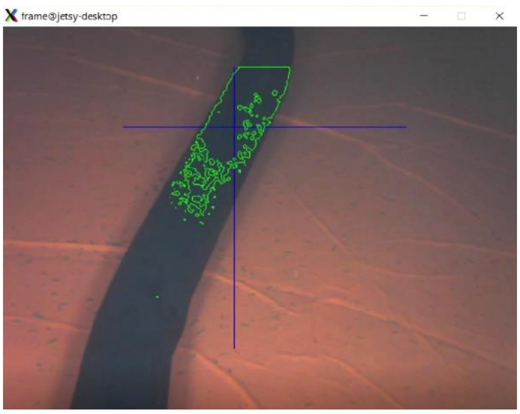


Fig 17. The results of testing the detection program with dark lighting parameters with the help of a 12 V LED light

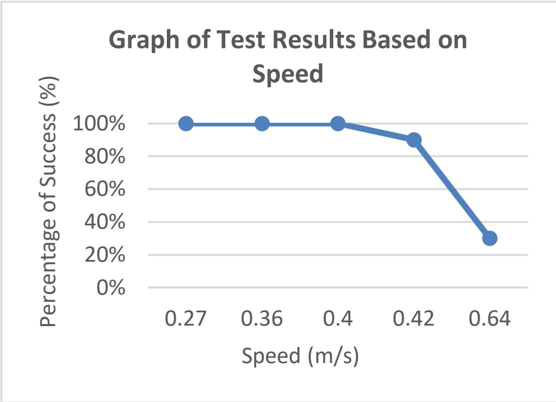


Fig 18. Object detection test results based on AUV motion speed

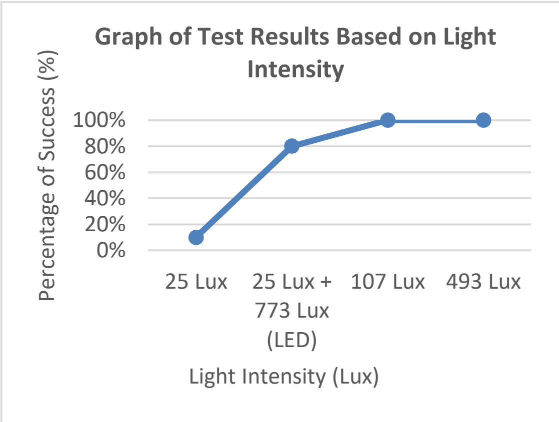


Fig 19. Object detection test results based on light intensity

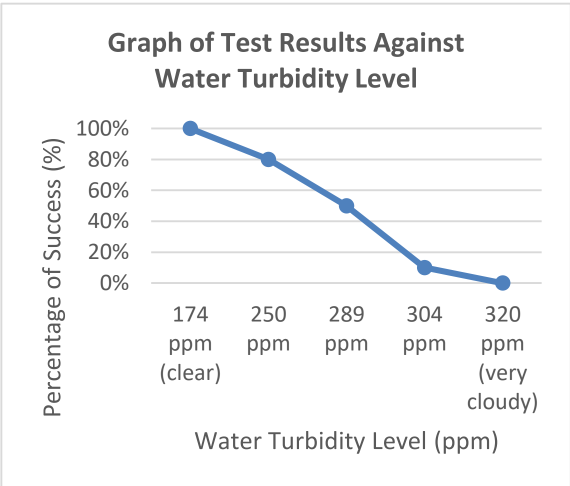


Fig 20. Graph of test results on the level of water turbidity

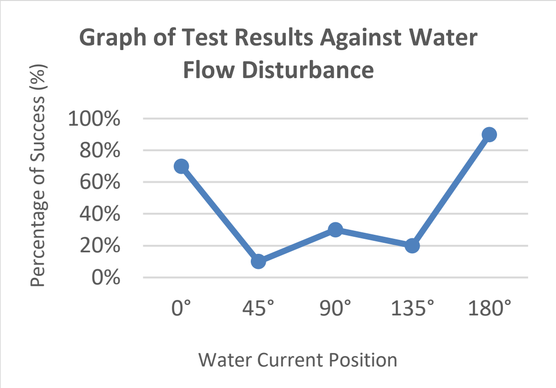


Fig 21. Graph of test results against water current disturbance.

Overall System Test

Table 1. AUV test results data on a straight line

Condition	Test 1	Test 2	Test 3	Average time
Bright light (493 Lux)	26 second	27 second	29 second	27,33 second
Low Light (107 Lux)	24 second	25 second	22 second	23,66 second
Dark Light (25 Lux)	-	-	-	-
Dark Light (25 Lux) + LED (773 Lux)	**23 second**	**24 second**	**30 second**	25,66 second

Condition	Test	Straight cable tracing process
Bright light (493 Lux)	1	Succeed
	2	Succeed
	3	Succeed
Low Light (107 Lux)	1	Succeed
	2	Succeed
	3	Succeed
Dark Light (25 Lux)	1	Unsuccessful
	2	Unsuccessful
	3	Unsuccessful
Dark Light (25 Lux) + LED (773 Lux)	1	Succeed
	2	Succeed
	3	Succeed
Percentage of Success		75%

Table 2. AUV test results data on winding paths

Condition	Test 1	Test 2	Test 3	Average time
Bright light (493 Lux)	36 second	57 second	58 second	50,33 second
Low Light (107 Lux)	26 second	29 second	30 second	28,33 second
Dark Light (25 Lux)	-	-	-	-
Dark Light (25 Lux) + LED (773 Lux)	32 second	30 second	26 second	29,33 second

Condition	Test	Straight cable tracing process
Bright ligh (493 Lux)	1	Succeed
	2	Succeed
	3	Succeed
Low Ligh (107 Lux)	1	Succeed
	2	Succeed
	3	Succeed
Dark Ligh (25 Lux)	1	Unsuccess ful
	2	Unsuccess ful
	3	Unsuccess ful
Dark Ligh (25 Lux) + LED (773 Lux)	1	Succeed
	2	Succeed
	3	Succeed
Percentage of Success		75%

CONCLUSION

Based on various tests and observations carried out on the AUV with an automatic underwater cable tracking system, the following conclusions were obtained:

1. The object detection system with visual image processing method using the OpenCV library works well in optimal conditions with a light intensity of more than 25 lux
2. The AUV cannot track on a straight line of 2 meters with a light intensity of less than 25 lux.
3. The detection system can work optimally at speeds of 0.27 m/s to 0.42 m/s
4. The vertical motion control system and speed control are less stable due to the absence of PID control.
5. The AUV can track on a straight and winding path of 2 meters with a bright light intensity of ±493 lux, a dim light of ±107 lux and a dark light intensity of ±25 lux with the help of an LED beam with a light intensity of ±773 lux.
6. The percentage of success of the scoping experiment on a straight track and a winding track with three trials is 75%.

REFERENCES

[1] Y. Ito, N. Kato, J. Kojima, S. Takagi, K. Asakawa, and Y. Shirasaki, “Cable tracking for autonomous underwater vehicle,” in *Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology (AUV’94)*, Jul. 1994, pp. 218–224. doi: 10.1109/AUV.1994.518628.

[2] D. L. Msongaleli, F. Dikbiyik, M. Zukerman, and B. Mukherjee, “Disaster-Aware Submarine Fiber-Optic Cable Deployment for Mesh Networks,” *J. Light. Technol.*, vol. 34, no. 18, pp. 4293–4303, Sep. 2016, doi: 10.1109/JLT.2016.2587719.

[3] S. Ren, “Evaluation of Ship Traffic Control Safety Based on Analytic Hierarchy Process,” in *2010 International Conference on Intelligent Computation Technology and Automation*, May 2010, vol. 2, pp. 237–240. doi: 10.1109/ICICTA.2010.216.

[4] W. S. Mary, S. Muthukumar, V. Suganya Pragathi, E. R. Reshma, and R. Viiavaraghavan, “International Maritime Boundary Alert for Fishermen,” in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Mar. 2018, pp. 246–248. doi: 10.1109/ICECA.2018.8474726.

[5] T. J. B. H. K. K. dan I. RI, “Law of The Republic of Indonesia Number 36 of 1999 on Telecommunications.” https://jdih.kominfo.go.id/produk_hukum/view/id/564/t/undangundang+nomor+36+tahun+1999+tanggal+8+september+1999 (accessed Sep. 08, 2021).

[6] L. D. Jatmiko, “Kabel Laut Putus, Biaya Perbaikan Bisa Lebih dari Rp50 Miliar,” *bisnis.com*, Jakarta, p. Telekomunikasi, Aug. 15, 2019.

[7] H. M. Manik *et al.*, “Autonomous Underwater Vehicle untuk Survei dan Pemantauan Laut,” *J. Rekayasa Elektr.*, vol. 13, no. 1, Art. no. 1, Apr. 2017, doi: 10.17529/jre.v13i1.5964.

[8] K. Asakawa, J. Kojima, Y. Kato, S. Matsumoto, and N. Kato, “Autonomous underwater vehicle AQUA EXPLORER 2 for inspection of underwater cables,” in *Proceedings of the 2000 International Symposium on Underwater Technology (Cat. No.00EX418)*, May 2000, pp. 242–247. doi: 10.1109/UT.2000.852551.

[9] J. O. Hallset, “Simple vision tracking of pipelines for an autonomous underwater vehicle,” in *1991 IEEE International Conference on Robotics and Automation Proceedings*, Apr. 1991, pp. 2767–2772 vol.3. doi: 10.1109/ROBOT.1991.132050.

[10] V. I. Koshelev and D. N. Kozlov, “Wire recognition in image within aerial inspection application,” in *2015 4th Mediterranean Conference on Embedded Computing (MECO)*, Jun. 2015, pp. 159–162. doi: 10.1109/MECO.2015.7181891.

[11] J. Zhang, L. Liu, B. Wang, X. Chen, Q. Wang, and T. Zheng, “High Speed Automatic Power Line Detection and Tracking for a UAV-Based Inspection,” in *2012 International Conference on Industrial Control and Electronics Engineering*, Aug. 2012, pp. 266–269. doi: 10.1109/ICICEE.2012.77.

[12] F. Tian, Y. Wang, and L. Zhu, “Power line recognition and tracking method for UAVs inspection,” in *2015 IEEE International Conference on Information and Automation*, Aug. 2015, pp. 2136–2141. doi: 10.1109/ICInfA.2015.7279641.

[13] G. Zhou, J. Yuan, I. Yen, and F. Bastani, “Robust real-time UAV based power line detection and tracking,” in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, pp. 744–748. doi: 10.1109/ICIP.2016.7532456.

[14] S. Yan and L. Jin, “Extra Matters Recognition of Transmission System Based on Hough Transform,” in *2011 Asia-Pacific Power and Energy Engineering Conference*, Mar. 2011, pp. 1–3. doi: 10.1109/APPEEC.2011.5748630.

[15] C. Pan, X. Cao, and D. Wu, “Power line detection via background noise removal,” in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2016, pp. 871–875. doi: 10.1109/GlobalSIP.2016.7905967.

[16] D. Bore, A. Rana, N. Kolhare, and U. Shinde, “Automated Guided Vehicle Using Robot Operating Systems,” in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2019, pp. 819–822. doi: 10.1109/ICOEI.2019.8862716.

[17] “Development of a Long Range Autonomous Underwater Vehicle for Ocean Observation.” https://ieeexplore.ieee.org/document/9389473/ (accessed Sep. 08, 2021).

[18] “OpenCV: Graph API: Converting image from one color space to another.” https://docs.opencv.org/master/dc/d38/group__gapi__colorconvert.html#ga3e8fd8197ab16811caf9b31cb1e08a05 (accessed Sep. 08, 2021).

[19] “OpenCV: Image Thresholding.” https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html (accessed Sep. 08, 2021).

[20] “OpenCV: Contour Features.” https://docs.opencv.org/3.4/dd/d49/tutorial_py_contour_features.html (accessed Sep. 08, 2021).

[21] E. Chebotareva and L. Gavrilova, “Educational Mobile Robotics Project ‘ROS-Controlled Balancing Robot’ Based on Arduino and Raspberry Pi,” in *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, Oct. 2019, pp. 209–214. doi: 10.1109/DeSE.2019.00047..

.