

Design And Implementation of a Depth Stability Control System Using A Water Pressure Sensor on An Underwater Remotely Operated Vehicle (ROV)

M. Afid Afandi¹, Puput Wanarti Rusimanto²

¹Electrical Engineering Departement, State University of Surabaya

¹A5 Building Ketintang Campus, Surabaya 60231, Indonesia

¹m.afid.21054@mhs.unesa.ac.id

²puputwanarti@unesa.ac.id

Abstract – This research aims to design and implement a depth control system for an Underwater ROV using the MS5837-30BA water pressure sensor, controlled by ESP8266 and STM32 microcontrollers. Data is transmitted in real time to a Ground Control Station (GCS), which also allows PID parameter configuration and joystick control. Initial testing of the sensor showed an average absolute error of 0.73 cm after adjustment. The PID control system was implemented using two approaches: MATLAB simulation and the trial-and-error method. The results show that the ROV depth control system can effectively maintain a setpoint of 50 cm below the water surface. The best performance was achieved using the trial-and-error method with PID parameters of $K_p=31$, $K_i=0.5$, and $K_d=9$. The system response demonstrated 0% overshoot, 0.02 cm steady-state error, 1.3 seconds rise time, and 1.6 seconds settling time.

Keywords: Underwater ROV, MS5837-30BA, PID Controller, Depth Control, System Identification.

I. INTRODUCTION

The development of global technology, especially in Robotics has many solutions, one of which is remotely operated vehicles, also known as ROVs [1]. Remotely Operated Vehicles (ROV) is one type of unmanned underwater vehicle whose main purpose is to perform underwater operations and observe underwater conditions. ROVs are very commonly used in places that cannot be reached by sailors and scientists whose purpose is to investigate and explore underwater science [2]. ROV underwater remote operation explorers have been widely used as human replacements for time-sensitive tasks and dangerous underwater work, and are usually equipped with various sensors and devices, such as cameras, propulsion systems, sensors, and sonar devices to analyze the collected data [3].

Autonomous Underwater Vehicles (AUVs) and ROVs, which are remotely operated, are essential in deep-sea exploration. They are used in many marine environments ranging from salvage, oil and gas exploration, ocean observation, etc. ROV control faces the major problem of unpredictable parameters (rise, hydrodynamics, coefficients, etc.). This concern is further increased by the current ability of ROVs to be modified. The characteristics of the underwater environment are highly variable, disturbing the ROV in the form of flowing water and interacting with waves on the device in shallow seas, and so on. ROVs are an increasing development over the past decade [4]. These highly variable characteristics of the underwater environment can interfere with ROV performance, making it difficult to maintain position and stability during operation.

High level accuracy and precision are required for ROV posture control system in complex and long-time working environment. ROV requires intelligent control algorithms to

make it a stable control system. In a broad sense, motion control includes depth control and arc control [5]. The implementation of the use of accurate and precise sensor types and adaptive control systems is necessary so that the underwater ROV can hold the depth position and maneuver stably in the water.

II. METHODS

The complete research design stages can be seen in the research design flowchart in Figure 1. below.

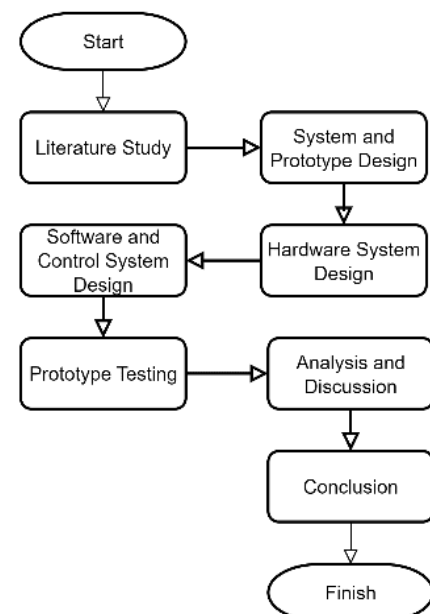


Figure 1. Flowchart of Research Design Underwater

ROV System Design

In this study, there are three subsystems that have their own functions, namely the propulsion system, the sensor and navigation system, and the ground station system. The overall system block diagram can be seen in Figure 2.

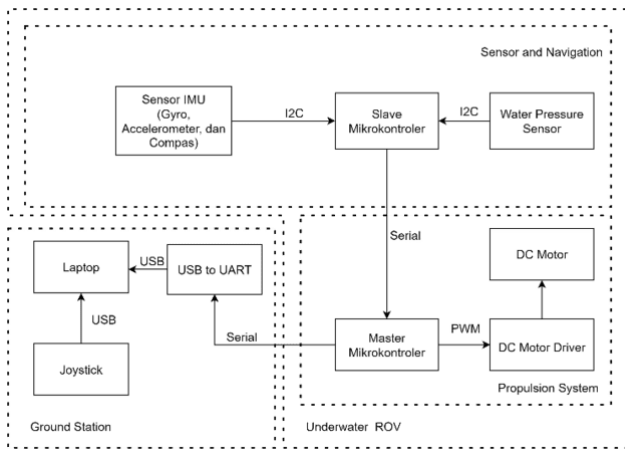


Figure 2. Block Diagram of the Underwater ROV

Prototype Design

The prototype design includes the 3D appearance and mechanical system of the ROV, as shown in Figure 3. It is square in shape with dimensions of 50 cm x 25 cm x 25 cm (length x width x height). The box containing the electrical components is placed in the center of the ROV to ensure good balance when in the water. This ROV is designed with four DC motors, two mounted vertically and two horizontally, as shown in Figure 4. With this motor configuration, the ROV has four degrees of freedom (DOF), as shown in Figure 5.

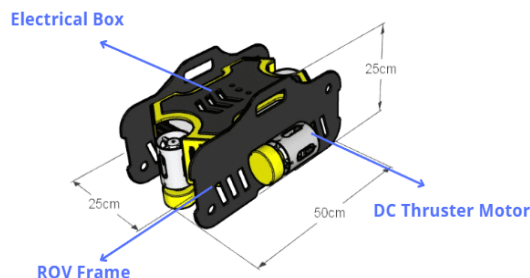


Figure 3. 3D Underwater ROV

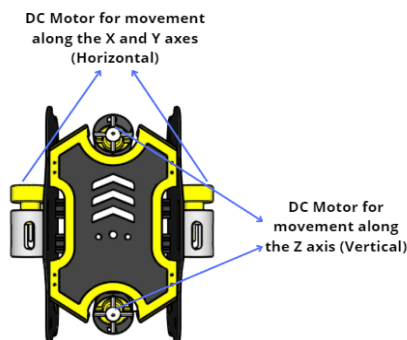


Figure 4. Positioning of the Thruster Motors on the Underwater ROV

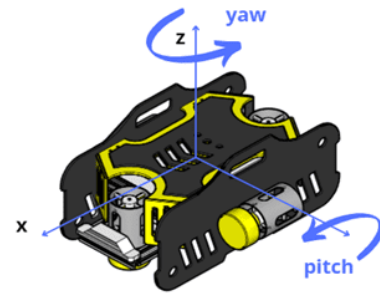


Figure 5. Degrees of Freedom (DOF) of the Underwater ROV

Hardware System Design

The electrical system on the ROV uses various components, including the STM32F103C8T6 as a microcontroller that runs the PID algorithm for depth control and sensor and motor data settings. The MS5837-30BA The MS5837-30BA sensor uses a DC supply voltage of 1.5~3.6V, pressure range of 0~30bar, I2C output [6]. The sensor functions to detect pressure which is converted to depth, while the MPU6050 and HMC5883L provide orientation and direction data for the ROV. The ESP8266 serves as the sensor data processor. Four BTS7960 motor drivers control the DC motors for horizontal, vertical, and rotational movement of the ROV. The primary power source is a 12V battery, distributed through the MINI 560 Step Down and AMS1117 Voltage Regulator to adjust component voltages. The system includes an ON/OFF switch for the main power and an emergency button to cut power to the motors in case of malfunction in the Underwater ROV. The electrical system block diagram of the underwater ROV is shown in Figure 6.

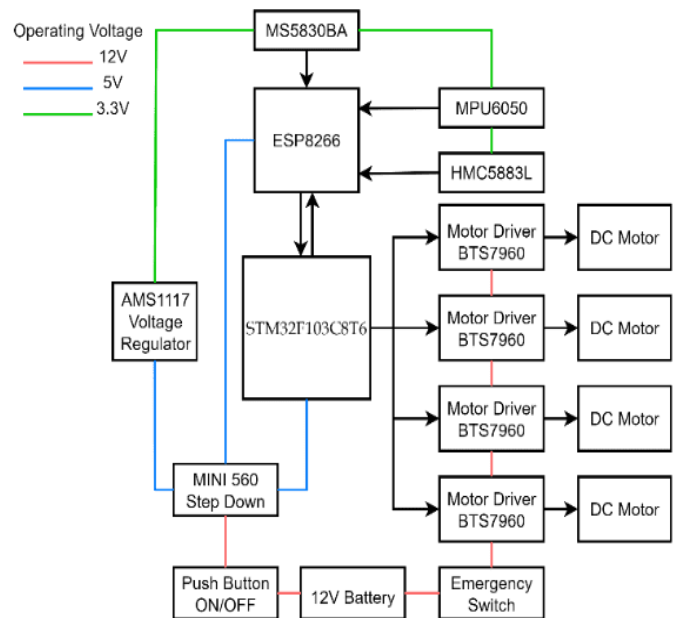


Figure 6. Underwater ROV Electrical System Block Diagram

Software Design

STM32F103C8T6 also processes the PID system to provide output to the thruster motor so that the underwater ROV can move optimally. In designing a PID control system, it is essential to adjust the K_p , K_i , and K_d parameters to ensure the system

responds appropriately to the desired setpoint [7]. ESP8266 as a sensor processing unit and sends data to the main controller of the system. Both microcontrollers are programmed using the C++ language in the Arduino IDE software, Flowchart System Underwater ROV can be seen in Figure 7.

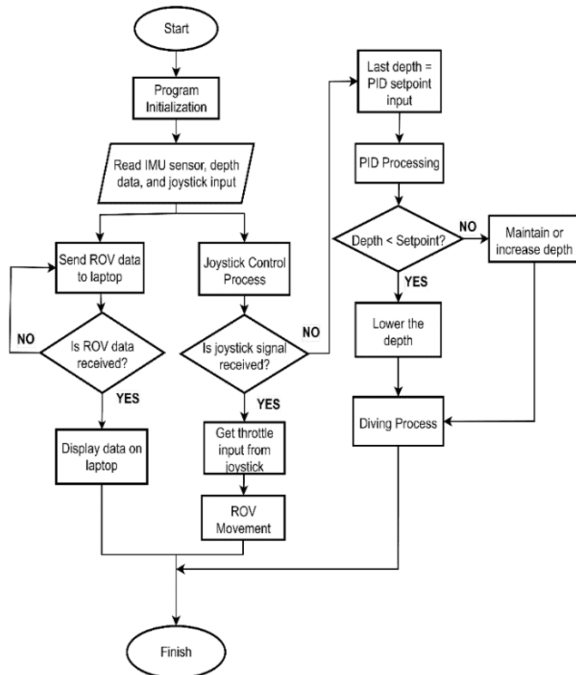


Figure 7. Underwater ROV System Flowchart



Figure 8. Ground Control Station

ROVs have control systems that allow the operator to control the movement and operation of the vehicle. This system includes the use of sensors and remote controls to collect data and provide feedback to the operator [8]. The Ground Control Station (GCS) system is designed using UART communication to connect with the Underwater ROV system. The Ground Control Station (GCS) system is made with features including displaying data from Underwater ROV, inputting PID parameters (K_p , K_i , K_d), displaying diving system response graphs in realtime, and equipped with a serial monitor. This can make it easier to do this research.

Depth Control System Design

The block diagram in Figure 9. represents the design of the control system used in the research. The STM32F103C8T6 microcontroller serves as the PID algorithm processor for each control variable to produce an optimal control signal in regulating the thruster motor speed. The MS5837 30BA water

pressure sensor is used as feedback to read the depth of the ROV, which is then compared to the setpoint value. Based on the error between the actual depth and the setpoint, the PID algorithm generates a control signal that is forwarded to the motor driver, so that the ROV can reach and maintain depth automatically and precisely. This system is capable of automatically adapting and is more accurate compared to an open-loop system [9].

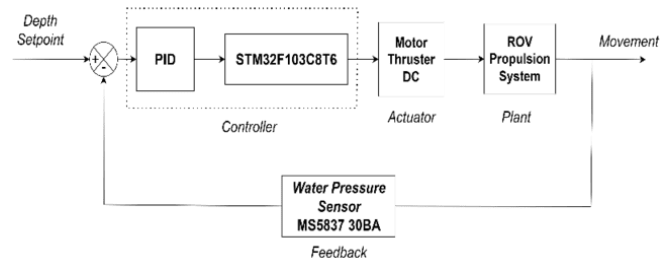


Figure 9. Underwater ROV Control System Block Diagram

Transfer Function Design of Underwater ROV Diving Motion

Determination of the transfer function of Underwater ROV diving motion is done through a system identification approach based on live diving experimental data, which represents the system model of the object. The System Identification Toolbox in MATLAB is used to obtain a mathematical model as the basis for determining the transfer function based on collecting input/output data from a system [10]. The identification stages include: data collection, data import into MATLAB, model structure selection, estimation, validation, and transfer function extraction for simulation. The imported data should be in the format of an id data object or two input-output arrays of appropriate dimensions, It is important to note that the output data matrix must have the same number of entries as the input data matrix [11]. Figure 11. shows the process of inputting data through the command window, which will display all the variables available in the workspace (Figure 10.).

Name	Value
data_id	6x1x1 iddata
input	[0,1,2,3,2,1]
output	[0,0.5000,1,1,...]
time	[0,0.1000,0,2...

Figure 10. Workspace MATLAB

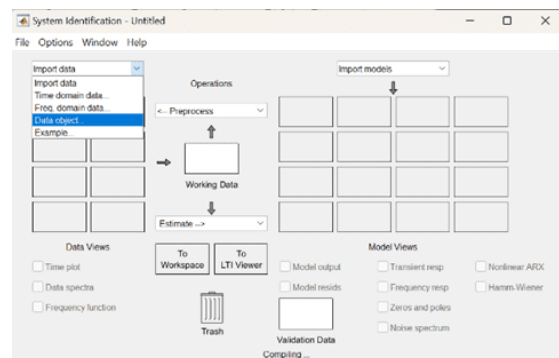


Figure 11. GUI System Identification Toolbox MATLAB

II. RESULT AND DISCUSSION

This research produces an underwater ROV that is designed using a water pressure sensor MS5837-30BA as input to maintain the depth level of the diving underwater ROV. In this test, a 3-cell 2200mAh Lipo battery was used. They are lightweight, rechargeable, and provide higher specific energy than many other battery types [12]. The results of the tool design can be seen in Figure 12. outside and Figure 13. inside.



Figure 12. Underwater ROV Outside



Figure 13. Underwater ROV Inside

Testing the MS5837BA Water Pressure Sensor on ROV

This test aims to evaluate the performance of the MS5837BA sensor in measuring water pressure and converting it into a depth value. The sensor is connected to ESP8266 for data reading and processing, then sent to STM32 and displayed via Arduino IDE serial monitor. The density of water used in the calculation is 997 kg/m^3 , according to the characteristics of fresh water. The following is the hydrostatic pressure formula.

$$P = \rho \times g \times h \quad (1)$$

$$h = P / (\rho \times g) \quad (2)$$

Description:

P = Hydrostatic pressure (Pascal/Pa)

ρ = Fluid density (kg/m^3)

g = Earth's gravity (m/s^2)

h = Depth (m)

Since the sensor measures absolute pressure (including air/atmospheric pressure), the pressure value needs to be reduced by atmospheric pressure (P_0) to obtain pure water pressure. Therefore, the conversion formula is used:

$$h = (P - P_0) / (\rho \times g) \quad (3)$$

$$h = (P - 101325) / (997 \times 9.81) \quad (4)$$

Based on Table 1., the MS5837-30BA sensor reading data is obtained along with the error value for each measurement. The average absolute error of all data is 4.53 cm, indicating that the sensor readings still have a fairly high difference compared to the actual measurements. It was also found that the accuracy of the sensor tends to increase as the depth increases, with the largest error of 6 cm at a depth of 10 cm and the smallest of 3 cm at 140 cm. To correct this, an adjustment was made by adding an average correction value of 4.53 cm to the program. Since the reading data is in the form of an integer (cm), the correction value is rounded to 5 cm.



Figure 14. Sensor Data Capture

In Figure 14. sensor data collection is done by inserting the Underwater ROV directly into the water and then measured using a meter from the sensor point to the water surface.

Table 1. Initial Sensor Reading Results

Actual depth (cm)	Sensor Testing Results	
	Sensor reading results (cm)	Absolute error / Difference (cm)
10	4	6
20	15	5
30	24	6
40	35	5
50	45	5
60	56	4
70	65	5
80	75	5
90	85	5
100	97	3
110	106	4
120	115	5
130	127	3
140	137	3
150	146	4
Rata - Rata		4.53

Table 2. shows the results of the second test after the program adjustments were made. The largest difference was recorded as 4 cm at a depth of 10 cm, and the smallest was 0 cm at several depth points such as 20 cm, 90 cm, 120 cm, and 130 cm. The average difference in sensor readings decreased significantly to 0.73 cm, much better than the initial test of 4.53 cm. The sensor is also able to read data stably every 100 ms. These results show that the sensor is accurate and responsive enough to be used as depth input in further ROV system tests.

Table 2. Sensor Reading Results After Adjustment

Actual depth (cm)	Sensor Testing Results	
	Sensor reading results (cm)	Absolute error / Difference (cm)
10	14	4
20	20	0
30	29	1
40	41	1
50	51	1
60	59	1
70	71	1
80	79	1
90	90	0
100	103	3
110	111	1
120	120	0
130	130	0
140	142	2
150	151	1
Rata - Rata		0.73

Ground Control System (GCS) Testing

Ground Control System (GCS) testing is conducted to ensure the successful reception and visualization of data from the Underwater ROV to the laptop device. The data displayed includes set point, depth, heading, and PWM value of each motor. In addition, GCS also has a feature to configure PID parameters and present system response graphs in real-time.

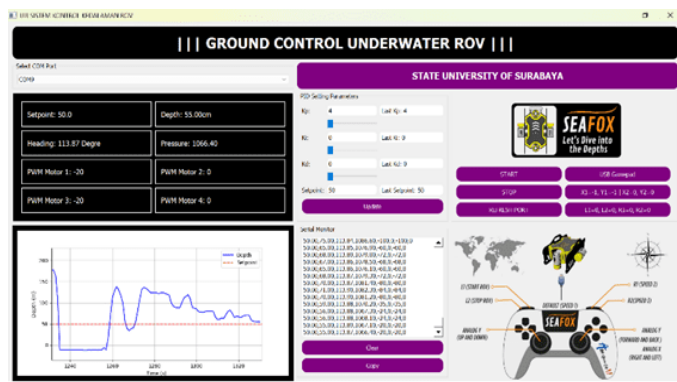


Figure 15. Ground Control Station Interface

The test results show that the GCS is able to accurately display all data sent by the Underwater ROV (Figure 15). The display of response graphs and CSV data on the serial monitor also strengthens the monitoring capabilities of the system. The parameters of the joystick can also be observed directly through the GCS. This system serves as an interface as well as ROV controller from the surface, with communication between GCS and ROV using serial protocol.

ROV diving testing using ON-OFF control

Testing of the Underwater ROV depth control system was conducted using the ON-OFF control method to assess performance without involving complex controllers. This system activates the vertical motor when the depth is below the setpoint and turns it off when it exceeds the setpoint. Test results with a

setpoint of 50 cm show that ON-OFF control produces a fluctuating response around the setpoint, as shown in Figure 16. The absence of stepped settings on the actuator causes the system response to be less smooth and stability is difficult to maintain. Although simple and easy to implement, this method has limitations in maintaining system stability.

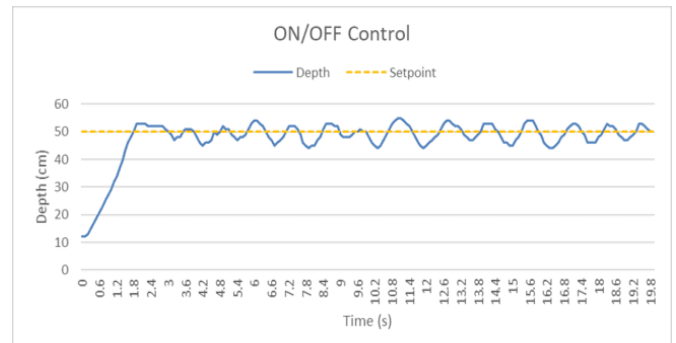


Figure 16. ON-OFF Control response graph

Transfer function design of ROV diving motion

In the transfer function design is done by the identification method through experimental data to obtain a mathematical model of the system. This system identification is done by experimenting and observing the relationship between the PWM value input data and the depth (position) output in the water on the Underwater ROV that has been made before. Data is collected by running the propulsion system by giving a step input signal that gives a sudden change in the input in the form of a PWM signal from 0 to 255. Input data in the form of PWM values and output ROV depth values, data collected in 10 second intervals with a sampling time of 100 ms to capture the dynamics of the system.

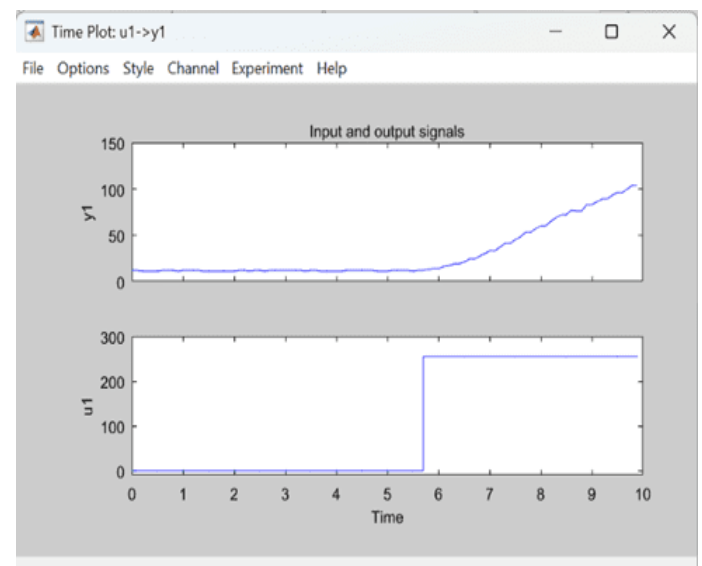


Figure 17. Input and Output data graph

Data entered into the System Identification Toolbox and model estimation with an order of 2 (poles) and 0 zeros resulted in a system model with a 96.04% fit to the estimation data. This value, as shown in Figure 18, is generally considered good or accurate for predicting the system because it exceeds 90%. The Final Prediction Error (FPE) is also low, at 1.42933.

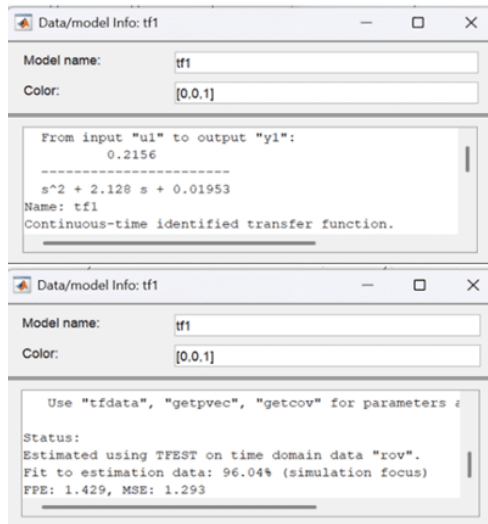


Figure 18. Transfer Function Result

After validation, the transfer function is obtained as follows:

$$G(s) = \frac{0.2156}{s^2 + 2.128s + 0.01953} \quad (5)$$

This model has a Mean Squared Error (MSE) value of 1.293, indicating that the model is very representative of the original data. The comparison between the model output and the actual data is shown in Figure 19.

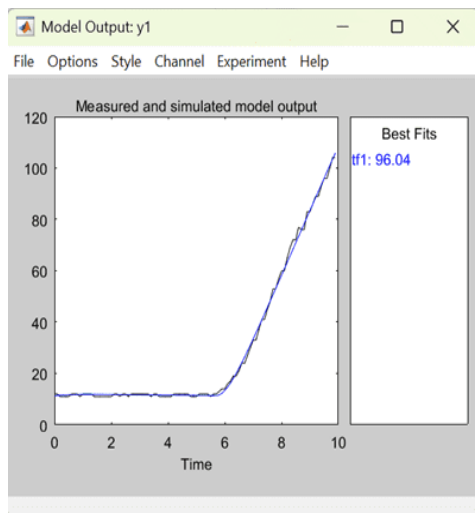


Figure 19. Compare Model Output

System Simulation Testing

Simulation testing is conducted to evaluate the performance of the control system using the Underwater ROV transfer function model that has been obtained through previous system identification. The purpose of this simulation is to determine the optimal PID control parameters to produce a stable system before being implemented on real devices. Simulation is performed using SIMULINK in MATLAB by entering the system model into the control block scheme, as shown in Figure 20.

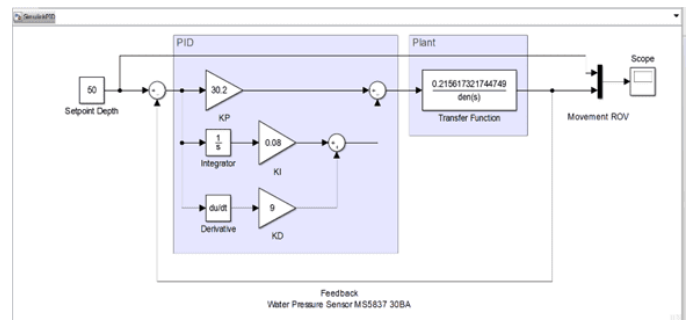


Figure 20. Depth Control System Simulation

After several iterations of simulation, the best response of the system to the 50 cm setpoint input was obtained. This response is used as the basis for selecting the most optimal PID parameters. The following is a graphic image of the response obtained from the simulation shown in Figure 21.

In the system response graph, the system shows good stability without excessive oscillation. The PID parameter values used are $K_p = 30.02$, $K_i = 0.08$, $K_d = 9$ respectively. These values are then applied to direct testing of the Underwater ROV to compare simulation performance with real conditions. The following are the results of simulation implementation in real conditions Figure 22.

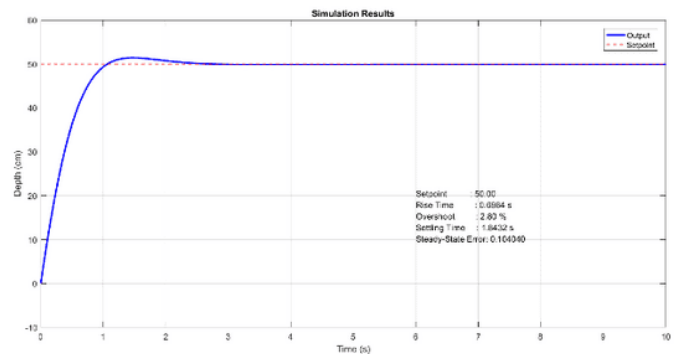


Figure 21. Simulation Result Response

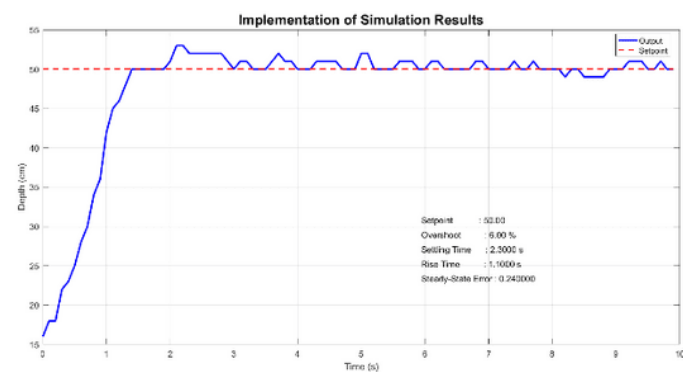


Figure 22. Real Condition Implementation Response

This difference shows that simulation provides more stable and faster results, but does not take into account various physical disturbance factors such as friction, shock, or actuator limitations that occur in real systems.

Table 3. SIMULINK Simulation Tuning Results

System Response	Response Results			
	Rise Time	Overshoot	Settling Time	Error Steady State
Simulation	0.69 s	2.80%	1.84 s	0.10 cm
Real test	1.1 s	6.00%	2.3 s	0.24 cm

ROV diving test using a PID control system with the trial-and-error tuning method.

This test uses the trial-and-error method to determine the PID parameters by trying various combinations of K_p , K_i , and K_d values manually. The goal is to find the combination that produces the best system response, according to the real characteristics of the Underwater ROV system. A total of 9 experiments were conducted with variations in K_p , K_i , and K_d values.

1. Variation of K_p value

Testing is done by trying 3 variations of the K_p value used, namely 25, 31, and 61. For K_i and K_d values, both are set at 0.

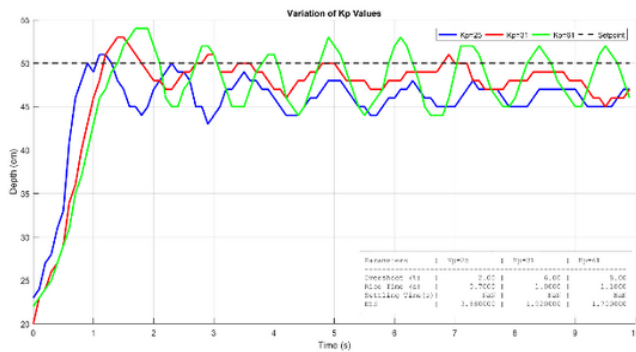


Figure 23. System response with varying K_p values

Table 4. System Response Results with K_p

PID Parameter			System Response			
K_p	K_i	K_d	Overshoot	Error steady state	Rise Time	Settling Time
25	0	0	2%	3.88 cm	0.7 s	-
31	0	0	6%	1.92 cm	1 s	-
61	0	0	8%	1.7 cm	1.1 s	-

Based on Figures 23. and Table 4. it can be seen that of the three proportional values tested, only $K_p = 31$ and $K_p = 61$ are able to reach the setpoint. Among them, $K_p = 31$ provides the best response with an overshoot of 6%, steady state error of 1.92 cm, and rise time of 1 second. In addition, the oscillation on the graph of $K_p = 31$ is smaller than that of $K_p = 61$.

2. Variation of K_i value

The best K_p value that has been obtained in the previous test is used in this test, namely the $K_p = 31$ value and in this test is done by trying 3 variations of the K_i value used, namely 0.08, 0.2, and 0.5. The K_d value is set at 0.

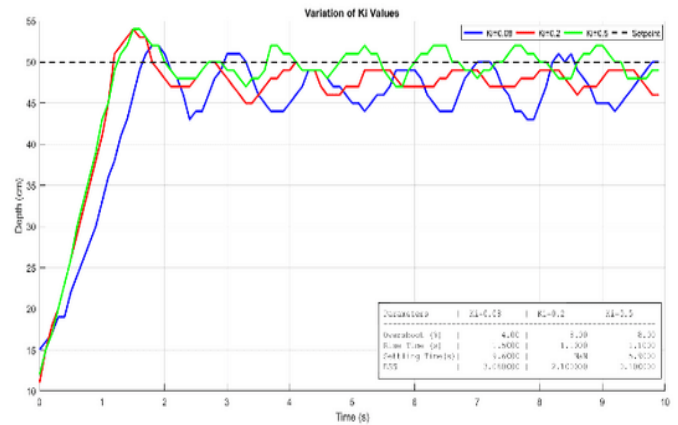


Figure 24. System response with varying K_i values

Table 5. System Response Results with K_p and K_i

PID Parameter			System Response			
K_p	K_i	K_d	Overshoot	Error steady state	Rise Time	Settling Time
31	0.08	0	4%	3.08 cm	1.5 s	9.6 s
31	0.2	0	8%	2.1 cm	1.1 s	-
31	0.5	0	8%	0.1 cm	1.1 s	5.9 s

Based on Figures 24. and Table 5. it can be concluded that of the three K_i value variations tested, $K_i = 0.5$ is the most optimal. This value produces a steady state error of 0.1 cm, rise time of 1.1 seconds, and settling time of 5.9 seconds, although it has an overshoot of 8% which is greater than $K_i = 0.08$ which is only 4%.

3. Variation of K_d value

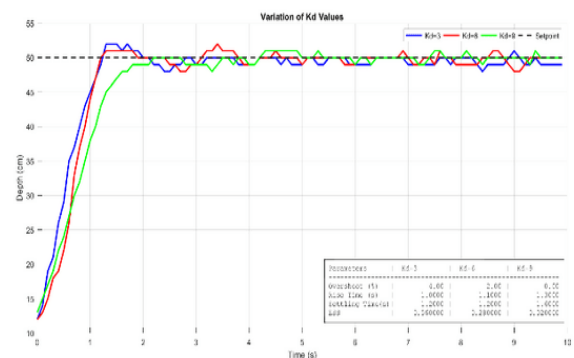


Figure 25. System response with varying K_d values

Table 6. System Response Results with K_p , K_i , and K_d

Parameter PID			System Response			
K_p	K_i	K_d	Overshoot	Error steady state	Rise Time	Settling Time
31	0.5	3	4%	0.56 cm	1 s	1.2 s
31	0.5	6	2%	0.28 cm	1.1 s	1.2 s
31	0.5	9	0%	0.02 cm	1.3 s	1.6 s

Based on Table 6. the results after adding the K_p value, the system with a value of $K_p = 9$ is considered to produce the most optimal and stable system response with overshoot parameters of 0%, steady state error of

0 cm, rise time of 1.3 seconds, and settling time of 1.6 seconds. Although $K_p = 9$ has a rise time value and settling time a fraction of a second slower than other parameters, when viewed from the visualization of the graph obtained, it appears that $K_p = 9$ has less fluctuation or noise.

In the testing phase using the trial-and-error tuning method, the optimal PID parameters were determined to be $K_p = 31$, $K_i = 0.5$, and $K_d = 9$. This method demonstrated excellent performance in maintaining dive stability. Underwater ROV can reach the setpoint with smoother response, and minimal system vibration. In this condition, the performance of the system with trial and error tuning is superior to other methods, because tuning is done directly based on observation of real system behavior. However, this method requires more time and experience to adjust the PID parameters.

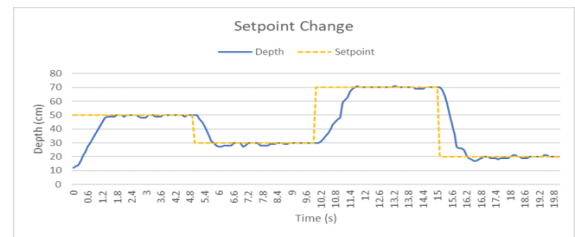


Figure 27. Setpoint Change Graph

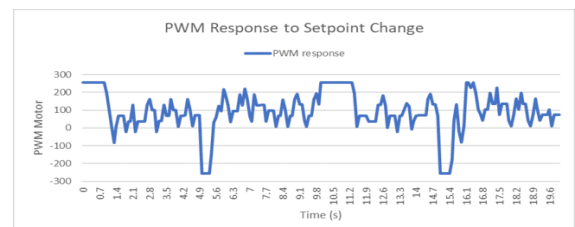


Figure 28. PWM Response of Setpoint Change

Testing the PID depth control system

This test is carried out to test the Underwater ROV depth control system that has been obtained in the previous test. Testing the ROV depth control system is done with setpoint variations and disturbances. Tuning uses MATLAB simulation and trial and error. The results of the two tuning methods can be seen in Table 7. Simulation is suitable for complex systems, while trial and error is more practical although it takes a lot of experiments. The best results were obtained from trial and error with PID $K_p = 31$, $K_i = 0.5$, and $K_d = 9$. These parameters will be used in testing the PID system.

Table 7. PID Implementation Results

Tuning Method	Response Results			
	Rise Time	Overshoot	Settling Time	Error steady state
Matlab Simulation	1.1 s	6.00%	2.3 s	0.24 cm
Trial and error	1.3 s	0%	1.6 s	0.02 cm

1. Testing setpoint value change

This test was conducted to see the system's response to sequential changes in depth (50 cm, 30 cm, 70 cm, and 20 cm) seen in Figure 26. As a result, the system is able to respond to each setpoint change well as seen in Figure 27. and Figure 28. shows the PWM value of the vertical thruster motor, where positive PWM indicates diving motion and negative PWM indicates surface motion.



Figure 26. Testing setpoint value changes

2. Test with interference

The response graph to the disturbance can be seen in Figure 30. which shows the system is disturbed with 6 times the push both from above and below. Underwater ROV is able to return to its original position (setpoint). The response graph of the PWM can also be seen in Figure 31.

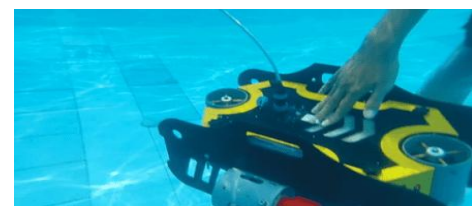


Figure 29. Impulse Disturbance Testing

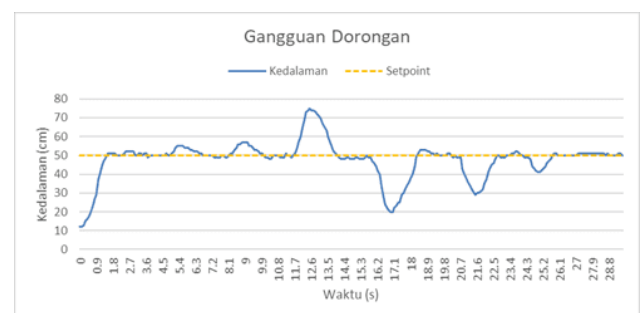


Figure 30. Impulse Disturbance Response Graph

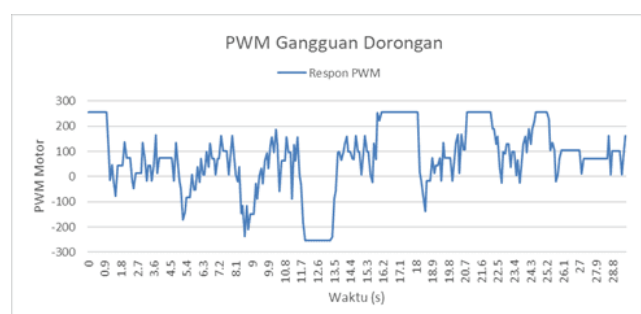


Figure 31. PWM Response Impulse Disturbance

III. CONCLUSION

Based on the results of the research conducted, it can be concluded that:

1. The underwater ROV was successfully created using a water pressure sensor as the main component and input from the system to maintain stability in maintaining diving depth. This sensor works by converting water pressure values into depth data using a hydrostatic formula, which is then processed by the PID control system.
2. The PID control system was successfully applied to the Underwater ROV system and proved to be effective in maintaining stability during vertical movement underwater. Two tuning methods were implemented: tuning using MATLAB software, tuning with Ziegler-Nichol's method and trial-and-error tuning. Among the three methods, the most optimal results were obtained using the trial-and-error tuning method with parameters $K_p = 31$, $K_i = 0.5$, and $K_d = 9$, resulting in system response characteristics with an overshoot value of 0%, steady state error of 0.02 cm, rise time of 1.3 seconds, and settling time of 1.6 seconds. Tests on setpoint changes and disturbance inputs to the Underwater ROV show that the system is able to maintain the depth position and return to its original position stably.
3. The MS5837-30BA sensor demonstrated good performance in reading pressure changes. Test results show that the sensor is accurate and responsive, with an average absolute error of 4.35 cm before calibration and 0.73 cm after calibration. This sensor is capable of providing accurate real-time input every 100ms, as evidenced by the Underwater ROV's ability to maintain depth position using PID control with good system response.

ACKNOWLEDGMENT

Thank you to Dr. Puput Wanarti Rusimamto, S.T., M.T., for her valuable guidance and insightful input throughout the course of this research. Her support and encouragement have been instrumental in the completion of this work.

REFERENCES

- [1] E. H. Binugroho, "eROV : Depth and Balance Control for ROV Motion using Fuzzy PID Method," pp. 637–643, 2019.
- [2] H. B. Amundsen, W. Caharija, S. Member, and K. Y. Pettersen, "Autonomous ROV inspections of aquaculture net pens using DVL," pp. 1–20.
- [3] I. Nterfacing, "Jurnal Teknologi CONTROLLER OF THE ROV USING MICRO-B OX," vol. 9, pp. 119–128, 2015.
- [4] X. Fang, H. Li, S. Zhang, J. Zhang, C. Wang, and X. Wang, "Engineering Applications of Artificial Intelligence Integration of ROV and vision-based underwater inspection for *Limnoperna fortunei* in water conveyance structure," Eng. Appl. Artif. Intell., vol. 124, no. May, p. 106575, 2023.
- [5] F. S. Ren, "Development Research on Vertical Depth-Fixing Control of Small-Size ROV," pp. 113–119, 2020.
- [6] S. Guo, X. Yang, J. Guo, and C. Li, "Design of wireless mobile environment monitoring system based on spherical amphibious robots," Proc. 2018 IEEE Int. Conf. Mechatronics Autom. ICMA 2018, pp. 960–965, 2018, doi: 10.1109/ICMA.2018.8484449.
- [7] S. W. Tika, S. Baqaruzi, and A. Muhtar, "Perancangan Sistem Pemantauan dan Pengendalian Debit Air Menggunakan Kontrol PID," ELECTRON J. Ilm. Tek. Elektro, vol. 2, no. 1, pp. 41–47, 2021, doi: 10.33019/electron.v2i1.2372.
- [8] F. Irawan and A. Yulianto, "Perancangan Prototype Robot Observasi

Bawah Air Dan Kontrol Hovering Menggunakan Metode Pid Control," J. Sains dan Teknol., vol. 1, no. 1, pp. 2460–173, 2015.

- [9] M. Ragno, "Understanding Open-Loop and Closed-Loop Control Systems: Features, Examples, and Applications," rteng.com. Accessed: Nov. 10, 2024.
- [10] Muhammad Thowil Afif and Ilham Ayu Putri Pratiwi, "Analisis Perbandingan Baterai Lithium-Ion, Lithium-Polymer, Lead Acid Dan Nickel-Metal Hydride Pada Penggunaan Mobil Listrik-Review," J. Rekayasa Mesin, vol. 6, no. 2, pp. 95–99, 2015.
- [11] E. Ramdani, "Parameter Identifikasi Transfer Fungsi Menggunakan MATLAB," Setrum Sist. Kendali-Tenaga-elektronika-telekomunikasi-komputer, vol. 4, no. 1, p. 30, 2016.
- [12] F. R. Utami, M. A. Riyadi, and Y. Christyono, "Sebagai Penggerak Robot Lengan Artikulasi," Transient, vol. 9, no. 3, 2020.