https://doi.org/10.31436/iiumej.v24i2.2759

SYSTEM IDENTIFICATION (SI) MODELLING, CONTROLLER DESIGN AND HARDWARE TESTING FOR VERTICAL TRAJECTORY OF UNDERWATER REMOTELY OPERATED VEHICLE (ROV)

FAUZAL NAIM ZOHEDI[1*], MOHD SHAHRIEEL MOHD ARAS[1], HYREIL ANUAR KASDIRIN[1], MOHD BAZLI BAHAR[1] AND LOKMAN ABDULLAH[3]

[1]Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka,
[2]Center for Robotics and Industrial Automation (CeRIA),
[3]Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka,
76100 Durian Tunggal, Melaka, Malaysia

[*]Corresponding author: fauzal@utem.edu.my

(Received: 20 February 2023; Accepted: 20 May 2023; Published on-line: 4 July 2023)

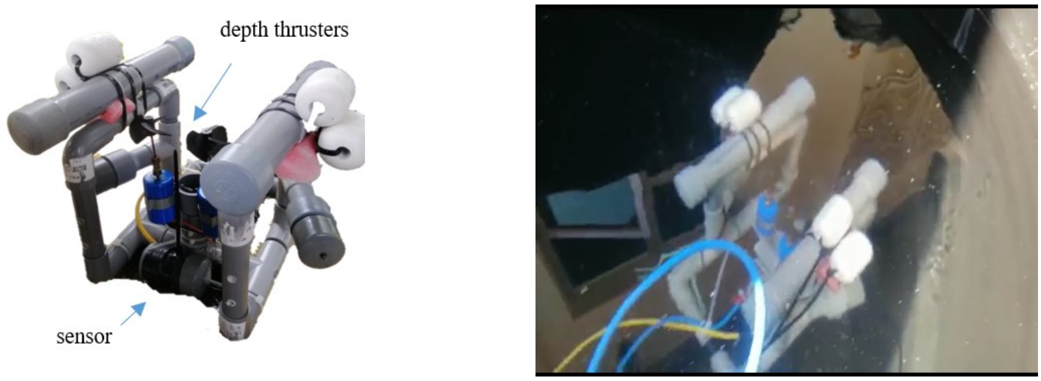
ABSTRACT: Underwater remotely operated vehicles (ROV) are important in marine industries to accomplish underwater exploration and surveying. The underwater environment makes it hard for ROV operators to control the manipulator while holding position simultaneously. This led to modelling and controller design for the vertical trajectory of ROV. In this paper, the System Identification (SI) modeling technique was used to model the vertical trajectory of the ROV. Then, the Proportional, Integral, and Derivative (PID) controller was implemented to control the trajectory. The SI modelling technique was used as it estimates the model based on the input and output relationship. MATLAB SI toolbox was used as the analytical software. Step and multiple step inputs were given to the system and the responses were recorded. The model with the best fit of 84.7% was selected and verified by comparing with actual output. The model response was then analyzed and the PID controller was implemented. The actual model had high percent overshoot (%OS) and steady state error (SSE). The PID implementation successfully reduced the %OS and eliminated the SSE.

ABSTRAK: Kenderaan bawah air kendalian jauh (ROV) adalah penting dalam industri marin bagi melaksanakan penerokaan dan pemerhatian bawah air. Persekitaran dalam air menyukarkan pengendali ROV bagi memanipulasi manipulator sambil memastikan kedudukan ROV secara serentak. Ini membawa kepada pemodelan dan mereka bentuk kawalan pergerakan menegak bagi ROV. Kajian ini menggunakan teknik pemodelan Sistem Pengenalan (SI) bagi memodelkan pergerakan menegak ROV. Kemudian, kawalan seimbang, menyeluruh dan terbitan (PID) dilaksanakan bagi mengawal trajektori. Teknik pemodelan SI digunakan kerana ia menganggarkan model berdasarkan kemasukan dan keluaran. Aplikasi MATLAB SI digunakan sebagai perisian analisis. Masukan satu langkah dan berbilang kali masukan telah dijalankan dan respons sistem direkodkan. Model yang paling sesuai mencapai 84.7% dipilih dan disahkan dengan perbandingan nilai keluaran sebenar. Respons model kemudiannya dianalisis dan kawalan PID dilaksanakan. Model sebenar mempunyai peratusan tinggi melampaui (%OS) dan ralat keadaan stabil (SSE). Pelaksanaan PID telah berjaya mengurangkan %OS dan menghapuskan SSE.

KEYWORDS: *system identification (SI) modelling; remotely operated vehicle (ROV); depth control; PID controller*

1. INTRODUCTION

Underwater remotely operated vehicles (ROV) are robots that substitute humans for underwater investigation and exploration. Underwater surveillance is important for sustainability of resources and underwater research [1]. Underwater exploration and surveillance, oil and gas pipeline monitoring, ship hull cleaning and other [2] complicated tasks require a highly skilled ROV operator to operate the ROV. The ability to manipulate the manipulator and maneuver the ROV at the same time is a must [3,4] for the operator. To ease the operator, automatic trajectory and position holding is necessary to ensure accurate control of the manipulator is achieved [4]. Other than that, there are also disturbances made by currents and waves [5,6] in the underwater environment. It increases difficulties for ROV operators to maintain the ROV at a certain depth.



(a) (b)

Fig 1: (a) ROV used in the research project; (b) ROV tested in controlled environment.

Underwater remotely operated vehicles (ROV) have six (6) degrees of freedom (DOF) which are surge (forward-reverse), sway, heave (vertical), roll, yaw (left and right) and pitch [7,8]. All these six degrees of freedom are coupled together. This coupling makes it difficult to model the ROV using mathematical modelling as many assumptions need to be made. Another alternative for modelling the ROV is using the System Identification (SI) technique. The SI technique is based on experimental data of the actual system. Input data is given to the system and the output response is recorded. These data are then run through the SI MATLAB toolbox to estimate the model based on the input-output relationship.

In this paper, a small ROV prototype was used. It was driven by 4 thrusters where 2 thrusters for vertical trajectory while another 2 thrusters were for forward-reverse and left-right turns. The vertical thrusters were put at the center of the ROV and placed side-by-side to provide stability to the ROV. The forward and reverse thrusters were placed at both sides of the ROV with 45 degrees angle that eased the turning process of the ROV. The ROV did not need to move fast as its solely used for investigation. The ROV had slightly positive buoyancy to ensure its ability to float if anything happened to the system. This position was measured as its initial position. The small ROV prototype used in the research project is shown in Fig. 1.

The ROV was tethered using a local area network cable (LAN). This cable provided 12V direct current (DC) to all 4 thrusters. The tethered cable also supplied 5V direct current (DC) for the sensor and was also used for data transfer between sensor and controller. The depth of the ROV was sensed using an MPC4250 pressure sensor. This sensor produced an output of 1.825 V at 1 atmosphere (ATM) and 1.8624 V at 22 cm in the water. The 22 cm is the location of sensor at the initial ROV condition floating in the water. This paper focuses on SI modelling and the Proportional, Integral, and Derivative (PID) controller was designed for the vertical trajectory of a newly developed small ROV. PID was selected as the controller due to its simple and easy implementation.

There are several basic controllers that have been implemented to perform ROV depth control. The controllers are the PID, a fuzzy logic controller (FLC), a neural network controller (ANN), and a sliding mode control (SMC) [4,9-13]. The easiest controller to implement is the PID controller as it can be tuned using auto-tuning in MATLAB Simulink[14]. The PID is good at controlling linear systems. The FLC controller is good at adapting with disturbance [15,16] but difficult to tune. ANN tunes its system by using historical data or online data [17,18]. Due to this, the ANN controller has a slow reaction time. The SMC controller works based on limitations set for the system to follow a slide path [19]. Its disadvantage is energy waste as it bounces through the path. Table 1 shows a summary of advantages and disadvantages of the basic controllers discussed [20].

Table 1: Comparison of the controller design

Type of controller	Advantages	Disadvantages
PID	Easy to execute and maintain	Only for linear systems
FLC	No mathematical modelling required and precise order	Complicated in the tuning process
ANN	Convergence to a precise model	Slower response
SMC	Non-linear system	Energy waste occurs

2. SYSTEM IDENTIFICATION MODELLING

The System Identification (SI) modelling approach starts with data gathering of the inputs and outputs of the system by experimentation [21-24]. In this project, step input and multiple stairs input were given to the system. The step input was used to model the system while the multiple stairs input was used to verify the model with the actual system. SI will estimate the relationship between the input and the output and generate a model with percentage best fit value. The best fit value will indicate how close the model is to the actual system. The SI approach consists of five (5) steps [20], [25] which are observation and data gathering, model structure selection, model estimation, model validation, and application. Fig. 2 shows the flow chart of the SI approach.

From Fig. 2, the first step is observation and data gathering. In this process, the system is observed, and input-output data are gathered. The next 3 flows, model structure selection, model estimation, and model validation are selected automatically by the MATLAB SI toolbox. Model structure selection is where the estimation method is selected. Model estimation is where the system will create a model, and model validation is where the generated model is validated. The last step is model application where the generated model is applied with any controller to control the system. The generated model must at least obtain an 80% best fit value to be considered as an acceptable model. The generated model can be in transfer function or state space before implementing to any system, the model must also be checked for controllability. The state space of the model is substituted into Eq. 1. If the Mc rank is equal to matrix A in the state space, the system is controllable.

$$M_C = [B\ AB\ A^2B\ A^{n-1}B] \qquad (1)$$

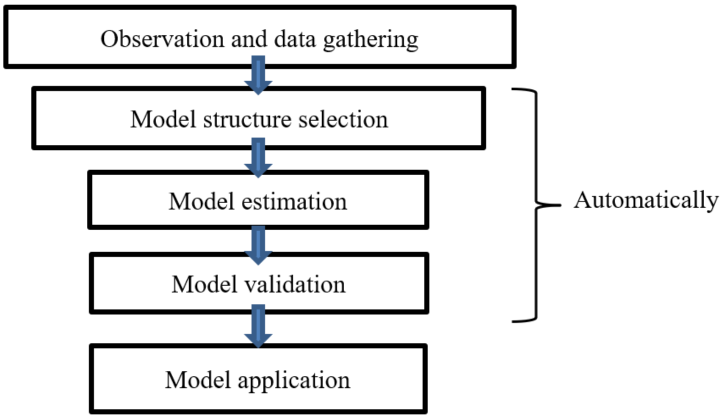


Fig. 2: Flow chart of SI approach.

2.1 Experimental Setup

In the experimental setup, there are 5 components as shown in Fig. 3. The components are a computer for user interface, a microcontroller Arduino Mega 2560 as interface between computer and ROV, a motor driver L298N to drive the thruster motor, a pressure sensor MPX425 to measure water pressure for depth, and the ROV.

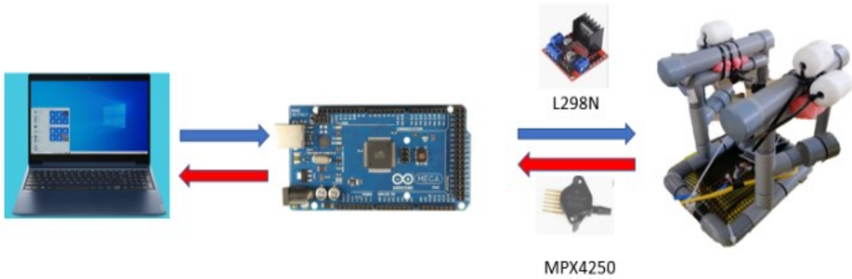


Fig. 3: Experimental setup.

The vertical trajectory of the ROV was observed by giving a step input and multiple stairs input to the system. The setups are listed below:

- Input step signal = 57 cm depth (limitation of tank)
- Repetitive stairs input = 57 cm and 40 cm
- Sampling time = 0.001 s
- Initial depth = 22 cm
- Initial volt = 1.825 V
- Triggered = 0.5 s for step and 0 s multiple input
- Experimental time = 10 s (step input) and 50 s (multiple input)

As the step given was 57 cm, the ROV submerged until 57 cm and maintained that position for 3 s. For multiple stairs input, the ROV submerged between 57 cm and 40 cm. From the sampling time, the step input produced 30000 data points while the multiple stairs input produced 40000 data points. The Simulink block diagram for data gathering is shown Fig. 4. The highlighted block in Fig. 4 shows the conversion block from volt to depth. The measurement from the sensor was divided with 1023 and multiplied with 5 V. This is due to the analog to digital conversion (ADC) of the Arduino that has a 10-bit configuration. The measured voltage was then converted to depth in centimeters.

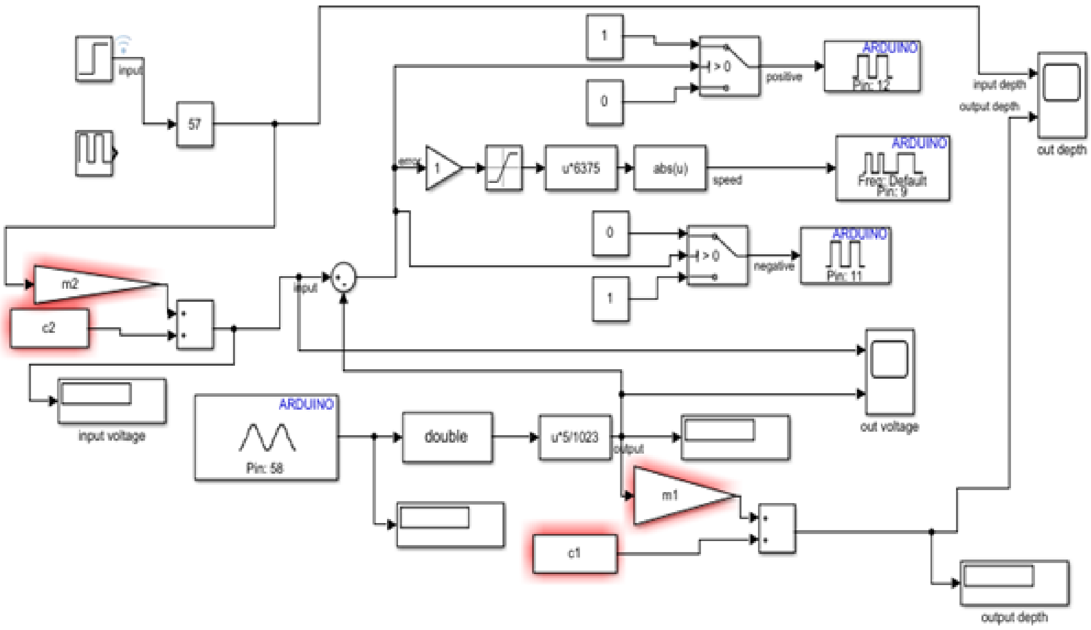


Fig. 4: Simulink block diagram for data gathering.

3. CONTROLLER DESIGN

For this research project, the PID controller was implemented by simulation and experimentation. It was first developed by Elmer Sperry in 1911. Proportional (P) is used to reduce the time rise (Tr) but affects the percentage of Overshoot (%OS). Derivative (D) is used to decrease the %OS and increase the stability, while Integral (I) is used to eliminate the steady state error (SSE) but may affect the performance of the system [26]. Equation 2 shows the equation of the PID controller where Kp is the gain of P, Kd is the gain of D, and Ki is the gain for I.

$$\frac{\boldsymbol{output\ u(t)}}{\boldsymbol{err\ (e)}} = \boldsymbol{Kp} + \frac{\boldsymbol{Ki}}{\boldsymbol{s}} + \boldsymbol{Kds} \qquad (2)$$

Figure 5 shows the basic block diagram used in this research project for PID controller implementation. The P, I, and D were connected in parallel with each other as highlighted in Fig. 5. Table 2 shows the effect of varying Kp, Ki, and Kd (Fig. 5) to the transient parameters; [26] time rise (Tr), percentage overshoot (%OS), Settling Time (Ts), and Steady State Error (SSE). For this research project, the PID was tuned manually to get the result. A step input of 57 cm depth was given to the system and the result was discussed and analyzed.

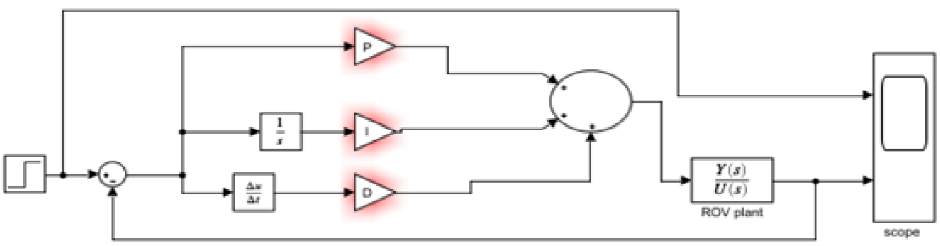


Fig. 5: Basic PID block diagram.

Table 2: The effect of varying Kp, Ki, and Kd (Fig.5) to the transient parameters

https://doi.org/10.31436/iiumej.v24i2.2759

Parameters	Time Rise (Tr)	Overshoot (%)	Settling Time (Ts)	Steady State Error (SSE)
Kp	Decrease	Increase	Small change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small change	Decrease	Decrease	Small change

4. RESULTS AND DISCUSSION

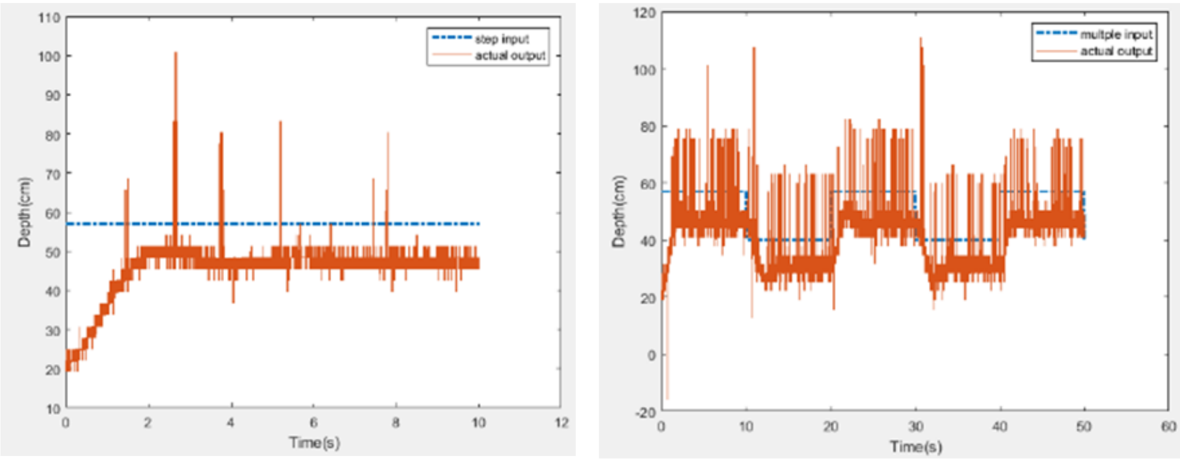
This section displays results and discussion for SI modelling and controller implementation.

4.1 SI Modelling

The experimental data for SI modelling is shown Fig. 6(a) and Fig. 6(b). The dash in the figure shows input given while the line given is the output response. The curves and spikes in the result occur because of the vibrations of the thruster motor and tolerance of the sensor.

The SI approach was implemented using the step data in MATLAB SI toolbox. The result shows 84.7% best fit and is considered acceptable. The nonlinear squares method was automatically selected by the SI system. The transfer function generated is shown in equation 3 below.

$$\mathrm{H}(s) = \frac{0.3808s^2+0.5898s+1.87}{s^2+1.554s+2.32} \tag{3}$$



(a) (a)

Fig. 6: Step input (a) and multiple stairs input (b) result.

The transfer function was then converted to state space and checked for controllability using Eq. 1. The result showed that the transfer function is controllable as the ranks of Mc and matrix A are similar. The transfer function was then verified by comparing it with the actual step output and multiple stairs output shown in Fig. 7 and Fig. 8.

The dashes in the figures were the model output while the wavy and spiked lines were the actual output. From the comparison in Fig. 7 and Fig 8, the model output was almost identical to the actual output. A bit overshoot and steady state error can be seen clearly in the model output and the actual output. The overshoot is 8.18%, the steady state error is about 11.4cm, and the settling time is 5.23 s. The overshoot and the error need were reduced by using the designed controllers.

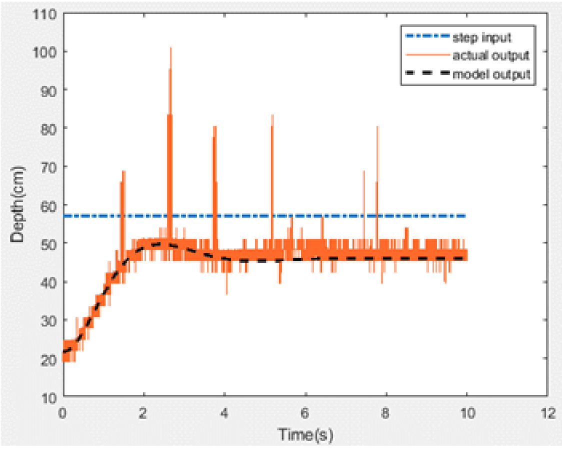


Fig. 7: step model output vs actual step output.

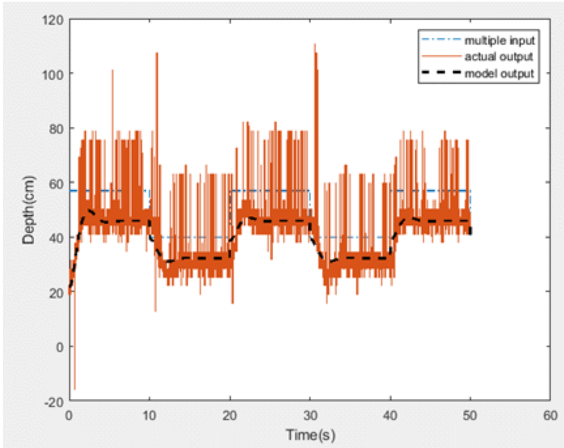


Fig. 8: Multiple stairs model output vs actual multiple stairs output.

4.2 Controller Implementation

The controller applied to the ROV model was PID. Figure 9 shows the result of the controller implementation.

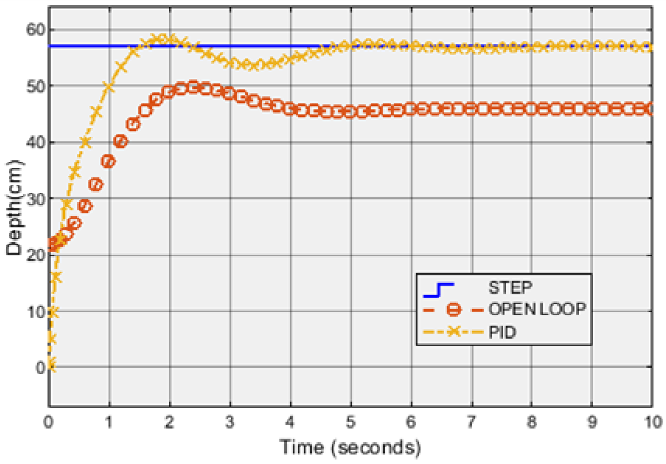


Fig. 9: PID controller implementation result.

The graph result in Fig. 9 was analyzed in terms of percentage overshoot (%OS), settling time (Ts), time rise (Tr) and steady state error (SSE). The result was tabulated in Table 3.

Table 3: Original system versus PID controller result

	Original System	PID
OS (%)	8.18	0.8
Ts (s)	5.23	4.37
Tr (s)	1.09	2.32
SSE (cm)	11.4	0

From the table, the PID shows remarkable reduction in the percentage overshoot (%OS) and steady state error (SSE). %OS went from 8.18% to 0.8% while SSE went from 11.4 to 0c

m. The settling time (Ts) decreased about 0.86 s from 5.23 s to 4.37 s. The drawback of the PID was the time rise (Tr) became slower by about 1s from 1.09 to 2.32s.

From this simulation result, the PID controller was then implemented experimentally to the ROV vertical trajectory. The result is shown in Fig. 10. From this figure, the experimentation for the PID controller looked almost identical to the simulation result. It shows that the PID controller can be implemented to the real ROV trajectory system.

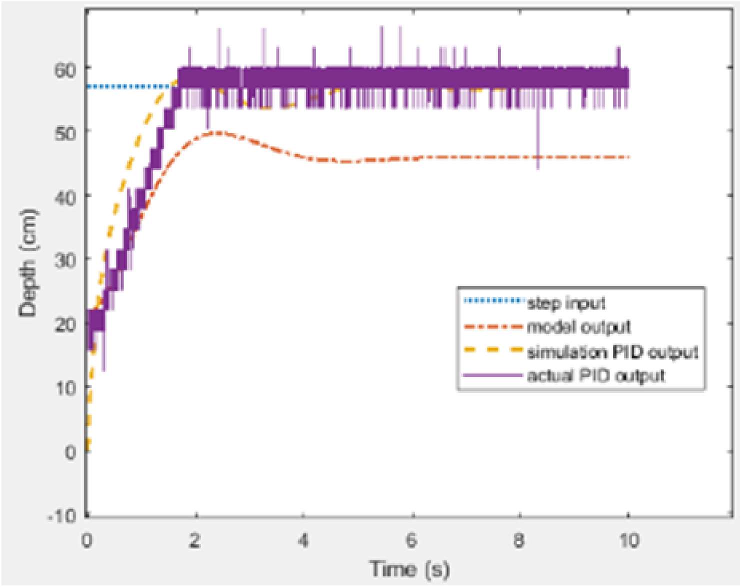


Fig. 10: Actual output result with PID controller implementation.

5. CONCLUSION

The modelling of vertical trajectory of an ROV is important to design a depth controller for the ROV. For the depth controller to maintain the ROV at certain position may offer more flexibility to ROV operators to manipulate the ROV manipulator or focus on any surveillance task. Modelling using the SI approach has proven to be successful as it manages to replicate the trajectory movement of the ROV at 84.7% best fit. The comparison between actual output and model output also shows near-identical results. In the controller design for the vertical trajectory, PID shows acceptable results in simulation and experimentally. Identical simulation and experimental results indicate that the PID controller can definitely be implemented in real life ROV trajectory control. The promising result shown by PID in all performance parameters used improve the original system. This paper is hopefully beneficial to engineers or controller designers to implement the SI model approach and design a PID controller for any real-life system.

ACKNOWLEDGMENT

We wish to express our gratitude and appreciation the support granted by Universiti Teknikal Malaysia Melaka in pursuing this research specially to Underwater Technology Research Group, Faculty of Electrical Engineering, Centre of Research, and Innovation Management for supporting this research.

REFERENCES

[1] Li G, Wu J, Tang T, Chen Z, Chen J, Liu H. (2019) Underwater acoustic time delay estimation based on envelope differences of correlation functions. Sensors, 19(5): 1-11. doi: 10.3390/s19051259.

[2] Binugroho EH, Wafiqqurochman, Mas'Udi MI, Setyawan B, Dewanto RS, and D. Pramadihanto D. (2019) EROV: Depth and Balance Control for ROV Motion using Fuzzy PID Method. IES 2019 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc., pp. 637-643. doi: 10.1109/ELECSYM.2019.8901673.

[3] Goheen KR, Jefferys ER. (1990) Multivariable self-tuning autopilots for autonomous and remotely operated underwater vehicles. IEEE J. Ocean. Eng., 15(3): 144-151. doi: 10.1109/48.107142.

[4] Aras MSM, Abdullah SS. (2015) Adaptive simplified fuzzy logic controller for depth control of underwater remotely operated vehicle. Indian J. Geo-Marine Sci., 44(12): 1995-2007.

[5] B. Huang and Q. Yang, "Double-loop sliding mode controller with a novel switching term for the trajectory tracking of work-class ROVs," Ocean Eng., vol. 178, no. March, pp. 80–94, 2019, doi: 10.1016/j.oceaneng.2019.02.043.

[6] N. Kumar and M. Rani, "An efficient hybrid approach for trajectory tracking control of autonomous underwater vehicles," Appl. Ocean Res., vol. 95, no. October 2019, p. 102053, 2020, doi: 10.1016/j.apor.2020.102053.

[7] G. Antonelli, S. Chiaverini, N. Sarkar, and M. West, "Adaptive control of an autonomous underwater vehicle: Experimental results on ODIN," IEEE Trans. Control Syst. Technol., vol. 9, no. 5, pp. 756–765, 2001, doi: 10.1109/87.944470.

[8] M. C. Nielsen, M. Blanke, and I. Schjølberg, "Efficient Modelling Methodology for Reconfigurable Underwater Robots," IFAC-PapersOnLine, vol. 49, no. 23, pp. 74–80, 2016, doi: 10.1016/j.ifacol.2016.10.324.

[9] M. S. M. Aras, S. N. B. S. Salim, E. C. S. Hoo, I. A. B. W. A. Razak, and M. H. Bin Hairi, "Comparison of fuzzy control rules using MATLAB toolbox and simulink for DC induction motor-speed control," SoCPaR 2009 - Soft Comput. Pattern Recognitpp. 711–715, 2009, doi: 10.1109/SoCPaR.2009.143.

[10] Z. Tang, Luojun, and Q. He, "A fuzzy-PID depth control method with ., overshoot suppression for underwater vehicle," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6329 LNCS, no. PART 2, pp. 218–224, 2010, doi: 10.1007/978-3-642-15597-0_24.

[11] M. S. Bin Mohd Aras, S. M. S. B. Syed Abdul Hamid, F. B. A. Azis, F. A. B. Ali, and S. Shah B Abdullah, "Study of the effect in the output membership function when tuning a Fuzzy logic controller," Proc. - 2011 IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2011, pp. 1–6, 2011, doi: 10.1109/ICCSCE.2011.6190485.

[12] M. L. Corradini and G. Orlando, "Brief Papers," Brain Cogn., vol. 32, no. 2, pp. 273–344, 1996, doi: 10.1006/brcg.1996.0066.

[13] M. . Aras, M.S.M., Abdullah, S.S. , Baharin, K.A. Nor, A.S.B.M. ,Mohd Zambri, "Model identification of an underwater remotely operated vehicle using system identification approach based on NNPC," Int. Rev. Autom. Control, vol. 8, no. 2, pp. 149–154, 2015.

[14] J. Ko, N. Takata, K. Thu, and T. Miyazaki, "Dynamic modeling and validation of a carbon dioxide heat pump system," Evergreen, vol. 7, no. 2, pp. 172–194, 2020, doi: 10.5109/4055215.

[15] K. Ishaque, S. S. Abdullah, S. M. Ayob, and Z. Salam, "Single input fuzzy logic controller for unmanned underwater vehicle," J. Intell. Robot. Syst. Theory Appl., vol. 59, no. 1, pp. 87–100, 2010, doi: 10.1007/s10846-010-9395-x.

[16] M. W. N. Azmi et al., "Comparison of controllers design performance for underwater remotely operated vehicle (ROV) depth control," Int. J. Eng. Technol., vol. 7, no. 3.14 Special Issue 14, pp. 419–423, 2018.

[17] N. Weake, M. Pant, A. Sheoran, A. Haleem, and H. Kumar, "Optimising parameters of fused filament fabrication process to achieve optimum tensile strength using artificial neural network," Evergreen, vol. 7, no. 3, pp. 373–381, 2020.

[18] H. Han, M. Hatta, and H. Rahman, “Smart ventilation for energy conservation in buildings,” Evergreen, vol. 6, no. 1, pp. 44–51, 2019, doi: 10.5109/2321005.
[19] H. Yu, C. Guo, and Z. Yan, “Globally finite-time stable three-dimensional trajectory-tracking control of underactuated UUVs,” Ocean Eng., vol. 189, no. March, p. 106329, 2019, doi: 10.1016/j.oceaneng.2019.106329.
[20] F. N. Zohedi, M. S. Mohd Aras, H. A. Kasdirin, and M. B. Bahar, “A new tuning approach of Single Input Fuzzy Logic Controller (SIFLC) for Remotely Operated Vehicle (ROV) depth control,” Evergreen, vol. 8, no. 3, pp. 651–657, 2021, doi: 10.5109/4491657.
[21] T. N. Dief and S. Yoshida, “System identification for Quad-rotor parameters using neural network,” Evergreen, vol. 3, no. 1, pp. 6–11, 2016, doi: 10.5109/1657380.
[22] T. N. Dief and S. Yoshida, “System identification and adaptive control of mass-varying quad-rotor,” Evergreen, vol. 4, no. 1, pp. 58–66, 2017, doi: 10.5109/1808454.
[23] M. Sanap, S. Chaudhari, C. Vartak, and P. Chimurkar, “HYDROBOT: An underwater surveillance swimming robot,” Proc. - 2018 Int. Conf. Commun. Inf. Comput. Technol. ICCICT 2018, vol. 2018-Janua, pp. 1–7, 2018, doi: 10.1109/ICCICT.2018.8325872.
[24] M. S. M. Aras, S. S. Abdullah, M. Z. A. Rashid, A. A. Rahman, and M. A. A. Aziz, “Development and modeling of unmanned underwater remotely operated vehicle using system identification for depth control,” J. Theor. Appl. Inf. Technol., vol. 56, no. 1, pp. 136–145, 2013.
[25] M. H. F. Taib, M. N., Adnan, R. and Rahiman, Practical System Identification. Penerbit UiTM, 2007.
[26] M. A. Salim, A. Noordin, and A. N. Jahari, “A robust of fuzzy logic and proportional derivative control system for monitoring underwater vehicles,” 2nd Int. Conf. Comput. Res. Dev. ICCRD 2010, pp. 849–853, 2010, doi: 10.1109/ICCRD.2010.187.