

New lambda tuning approach of single input fuzzy logic using gradient descent algorithm and particle swarm optimization

Fauzal Naim Zohedi, Mohd Shahrieel Mohd Aras, Hyreil Anuar Kasdirin, Nurdiana Binti Nordin

Underwater Technology Research Group (UTeRG), Center for Robotics and Industrial Automation (CERIA),

Fakulti Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

Article Info

Article history:

Received Oct 29, 2021

Revised Jan 11, 2022

Accepted Jan 17, 2022

Keywords:

Gradient descent algorithm
Particle swarm optimization
Proportional integral derivative controller
Remotely operated vehicle
Single input fuzzy logic controller

ABSTRACT

Underwater remotely operated vehicle (ROV) is important in underwater industries as well as for safety purposes. It can dive deeper than humans and can replace humans in a hazardous underwater environment. ROV depth control is difficult due to the hydrodynamic of the ROV itself and the underwater environment. Overshoot in the depth control may cause damage to the ROV and its investigated location. This paper presenting a new tuning approach of single input fuzzy logic controller (SIFLC) with gradient descent algorithm (GDA) and particle swarm optimization (PSO) implementation for ROV depth control. The ROV was modeled using system identification to simulate the depth system. Proportional integral derivative (PID) controller was applied to the model as a basic controller. SIFLC was then implemented in three tuning approaches which are heuristic, GDA, and PSO. The output transient was simulated using Matlab/Simulink and the percent overshoot (OS), time rise (Tr), and settling time (Ts) of the systems without and with controllers were compared and analyzed. The result shows that SIFLC GDA output has the best transient result at 0.1021% (OS), 0.7992 s (Tr), and 0.9790 s (Ts).

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Fauzal Naim Zohedi

Underwater Technology Research Group (UTeRG), Center for Robotics and Industrial Automation (CERIA),

Fakulti Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka

76100 Durian Tunggal, Melaka, Malaysia

Email: fauzal@utem.edu.my

1. INTRODUCTION

In the underwater engineering field, remotely operated vehicle (ROV) plays an important role in underwater observation, investigation, and inspection [1]–[3]. Especially in the oil and gas industry, ROV is used to do underwater pipe inspections as well as repairing jobs. ROV normally suffered from problems that include pose recovery or station keeping, underactuated conditions, coupling issues, and communication techniques [4]. This research paper was focusing on ROV depth control or station keeping.

Station keeping at a certain depth is very important for underwater exploration and inspection [5]–[7]. However, controlling ROV is difficult because of the unexpected and unpredictable underwater environment [4], [8]. This is due to the nonlinear hydrodynamics effect, coupled characters of plant equations, lack of precise models of underwater vehicle hydrodynamics and uncertainty parameters [9], [10], as well as the presence of environmental disturbances [1], [11]–[14].

Controller design, based on simple models of underwater vehicle mass and drag, generally yields unacceptable performances [15]. Linear (conventional) controller is unable to adequately control the unmanned underwater vehicle (UUV) satisfactorily [16]. Even for a one-axis motion such as vertical motion or heave motion, consistent performance for a desirable range is required. Overshoot in the system is

unacceptable as it can harm the ROV or its inspection location [14], [17]–[20]. It is best to have as minimum overshoot as possible in the ROV system.

There are many controllers designed by the researcher to cater to this problem. There are proportional, integral, and derivative (PID) based controllers and artificial based controllers. PID is a simple control technique that has been universally used because of the simplicity of implementation in a real-time system. Even for work class ROV, PID is used as its controller. However, the limitation is that it cannot dynamically compensate for unmodelled vehicle's hydrodynamics forces or unknown disturbances. Furthermore, the contradictory parameter configuration such as between the rise time and overshoot may also exist. PID controllers have been implemented in previous work for tracking purposes in UUV [21]–[25] as well as in ROV [17], [26]. Normally, the PID controller was used as a basic controller to be compared with another complex controller such as with linear quadratic regulator (LQR) [6], Fuzzy-based PID [27], [28]. The PID was hard to be tuned to cope with the non-linear nature of the underwater environment, in which it typically produces high overshoot and high steady-state error. Due to the limitation of PID, artificial intelligent (AI) based controllers such as fuzzy logic controller (FLC) and artificial neural network (ANN) had been introduced to control ROV.

ANN was used to predict the performance of the ROV depth system based on previous input and minimize the cost function [29]. Then, the best input is suggested. The ANN results were superior compared to other controllers that were experimented with. ANN was also used to tune the PID and adapt with the depth changing of ROV [30]. It was also used for the same purpose in [31] by implementing a radial basis function neural network (RBFNN) for trajectory tracking for the autonomous underwater vehicle (AUV). Both showed good results. However, the downside of ANN was the long computational time that may lead to a lagging problem. Another AI-based controller for the ROV system is the FLC which was applied in ROV [21], [23], and AUV [32]. The FLC controller can cope with an unknown mathematical modeling system. Implementation of FLC eases the need for precise and complex hydrodynamic modeling of the vehicle. For example, FLC was successfully used to tune the PID controller for underwater vehicle [33].

However, even with the adaptability advantage, FLC has a challenging level of complexity. Hence, a simplified single input fuzzy logic controller (SIFLC) is proposed to control the depth of ROV. Previous works have shown that SIFLC has excellent performance, and it exactly resembled conventional FLC transient response [34], [35]. SIFLC reduces the input of conventional FLC into single input single output (SISO) system. Normally, a trial-and-error (heuristic) method was used to find the optimum parameter. Consequently, it takes more time execution to find the optimum parameters.

This paper presents a new tuning approach of SIFLC with gradient descent algorithm (GDA) and particle swarm optimization (PSO) implementation for ROV depth control. The ROV was modeled using system identification to simulate the depth system. PID controller was applied to the model as a basic controller. SIFLC was then implemented in three different tuning approaches which were trial-and-error (heuristic), GDA, and PSO. The output transient was simulated using Matlab/Simulink and the percent overshoot (OS), rise time (Tr), and settling time (Ts) of the systems without and with controllers were compared and analyzed. In terms of depth control, the overshoot (%OS) is an important parameter to observe as a high value may damage the ROV or its investigation place [14], [18]–[20], [36]. The rise time (Tr) shows the time taken to get to the desired point while the settling time is the time ROV stabilizes at a steady state.

2. SYSTEM MODELLING

In this paper, the ROV was modeled using the system identification (SI) method in a Matlab computing environment. For system identification, the heave or vertical movement of ROV is being tested experimentally. Real-time input-output experimental data was gathered. 5 steps need to be considered in implementing system identification. Figure 1 shows the 5 steps for the SI approach. The steps are observation and data gathering, model structure selection, model estimation, model validation, and model application [37]. There are two sets of data needed for this experiment: training and validation data. The multi-sine signal was used to get the experimental data for training and validation. The input and output data were recorded, and the transfer function of the system was estimated using the Matlab toolbox. The input given to the ROV system can be a pulse, steps, random binary sequence (RBS), pseudo-random binary (PRBS), m-level pseudo-random (m-PRS), and multi-sine [37].

In the Matlab computing environment, multi-sine input was given to the system and the instrument variable (IV) approach was selected. Then, the selected model structure was implemented for model estimation and model validation to generate an ROV model. Finally, the model generated is used to design the ROV controller. The experiment was conducted in a controlled environment where the disturbances were not considered. In the instrument variable approach, the IV model was combined with 3 poles and 2 zeros for the transfer function. The best-fitting match was 96.43%. The transfer function generated is shown in (1).

$$H(S) = \frac{0.02332s^2+0.04058s+0.01126}{s^3+0.7114s^2+0.1861s+0.01398} \tag{1}$$

The generated output transient response is shown in Figure 2. The output result had no overshoot, 18.18 s of Tr, 33.21 s Ts, and 0.1947 of steady state error (SSE). Although the ROV did not overshoot, these results showed that the ROV model had a 19.47% target error and it took approximately half a minute to stabilize. This generated model was then simulated in Matlab/Simulink as a closed-loop system shown in the block diagram in Figure 3.

This generated model was then simulated in Matlab/Simulink as a closed-loop system shown in the block diagram in Figure 3. From Figure 4, the closed-loop model had a faster Tr (9.07 s) and Ts (14.76 s) compared to the open-loop result but the steady-state error shoot up to 55.55% from the input given to the system. From the output result, the controller needs to be applied to get a better output response.

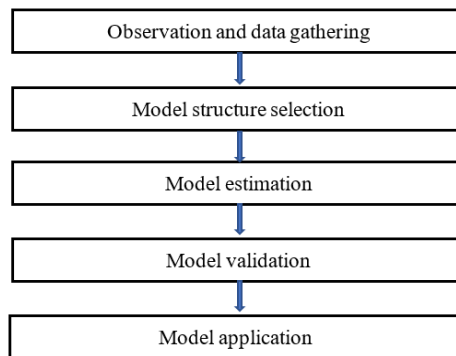


Figure 1. System identification approach for modelling of ROV

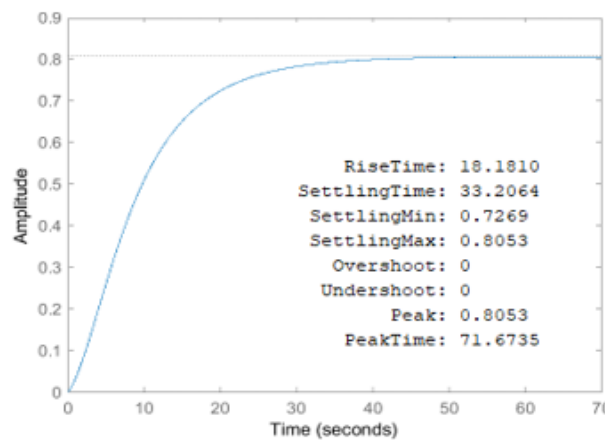


Figure 2. Transient response of the ROV model

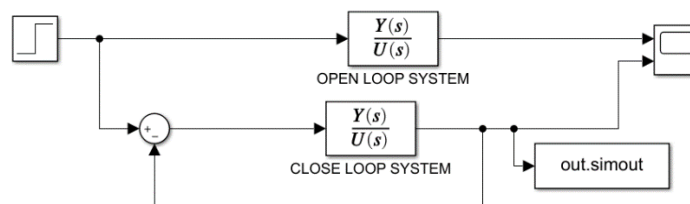


Figure 3. Matlab/Simulink closed-loop block diagram

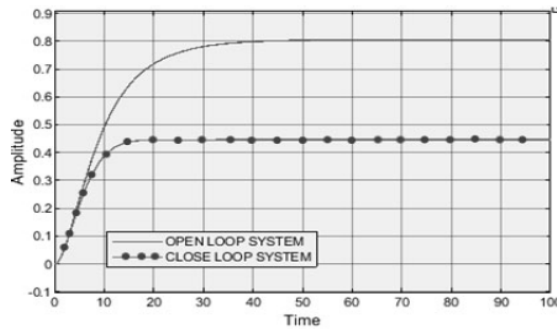


Figure 4. Open-loop and close loop transient output comparison

3. CONTROLLER DESIGN

In this paper, the PID controller was designed using auto-tuning provided by Matlab/Simulink. The SIFLC controller was designed and tuned using a heuristic method, GDA, and PSO. PID was used as a basic controller to be compared with the SIFLC controller designed.

3.1. PID controller

PID controller is the basic controller applied to the ROV system. The P, I, and D blocks were put in parallel in front of the plant to control the system. The P-block counters the direct error, the I-block rectifies the total errors in the system while the D-block minimizes the speed of the errors. The P-controller will make the response faster but intends to produce overshoot. The I-controller eliminates SSE while the D controller decrease overshoot. The PID controller block diagram is shown in Figure 5. The PID was tuned using automatic tuning in Matlab/Simulink [38].

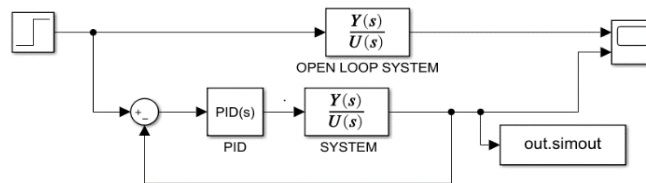


Figure 5. PID controller block diagram

3.2. FLC controller

Fuzzy logic controller (FLC) is a human decision based controller introduced by Lotfi A. Zadeh. It was introduced in 1965. What do be done or reaction of the system is based on human perspective of the thig itself. In FLC algorithm, there are four (4) basic components; fuzzification, knowledge based (rules), inference engine and defuzzification. The fuzzification change the raw data into membership functioned, knowledge based create rules for decision making, inference engine which act as intelligent system and defuzzification change the fuzzy decision data into real output raw data. Figure 6 shows the basic configuration of FLC component in a block diagram [39].

The common 7 X 7 FLC table is shown in Table 1. The Z in the table stand for zero which indicate the center of the decision table. PL stand for positive large, PM stand for positive medium and PS stand for positive small. The NL, NM, and NS are the opposite of PL, PM and PS where they indicate the negative side of the table. The output result for FLC is selected based on the table.

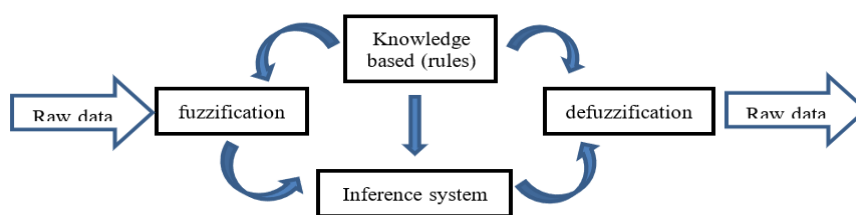


Figure 6. Basic configuration of FLC component

Table 1. 7 X 7 FLC table

Err vs du/dt or 1/s	PL	PM	PS	Z	NS	NM	NL
NL	Z	NS	NM	NL	NL	NL	NL
NM	PS	Z	NS	NM	NL	NL	NL
NS	PM	PS	Z	NS	NM	NL	NL
Z	PL	PM	PS	Z	NS	NM	NL
PS	PL	PL	PM	PS	Z	NS	NM
PM	PL	PL	PL	PM	PS	Z	NS
PL	PL	PL	PL	PL	PM	PS	Z

3.3. SIFLC controller

SIFLC controller is designed based on conventional FLC designed. The conventional FLC table, Table 1 is manipulated using signed distance method (SDM) which reduced the rules table to a one-dimensional array [40], [41]. From Table 1, it can be seen there is a consistent pattern in the decision-making of the FLC output. From Tables 1 and 2 diagonal lines were created which are named LZ and LNS. ‘d’ is the distance between LZ and LNS given by (2). The lambda (λ) equation is shown in (3) and (4).

$$d = \frac{w+Z_e\lambda}{\sqrt{1+\lambda^2}} = \frac{w}{\sqrt{1+\lambda^2}} + \frac{Z_e\lambda}{\sqrt{1+\lambda^2}} \tag{2}$$

$$\dot{e} + \lambda e = 0 \tag{3}$$

$$\therefore \lambda = -\frac{\dot{e}}{e} \tag{4}$$

Figure 7 shows the derivation of d, which is the distance between point, Q, and point, P. The conventional FLC table is now reduced to Table 2 where the diagonal line was represented by LNL, LNM, LNS, LZ, LPS, LPM, and LPL while NL, NM, NS, Z, PS, PM, and PL represent the output of corresponding diagonal lines.

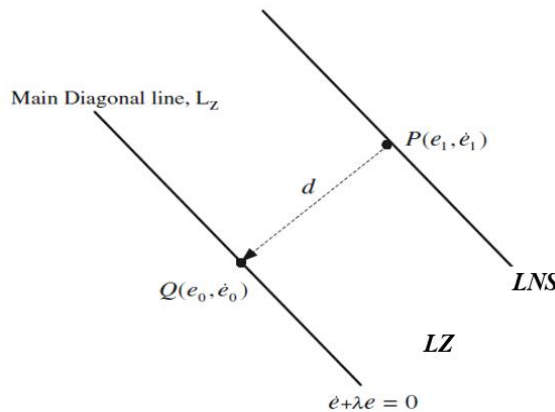


Figure 7. Derivation of d, the distance between point Q and P

This input-output of SIFLC can be replaced by a lookup table. SIFLC was then tuned using the proposed lambda (λ) tuning method. The value of (λ) was heuristically changed to obtain the best output. The (λ) was linked to the FLC by the input of the FLC. The range of error and integral error was plotted in a graph shown in Figure 8.

Table 2. Reduced FLC table using SDM

d	LNL	LNM	LNS	LZ	LPS	LPM	LPL
out	NL	NM	NS	Z	PS	PM	PL

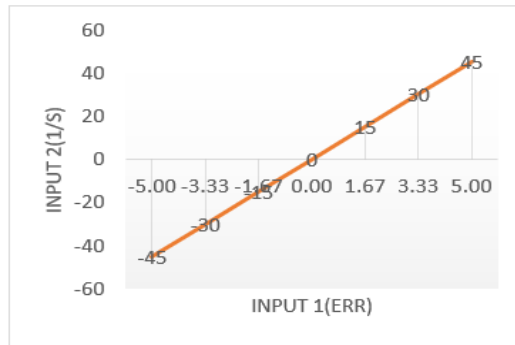


Figure 8. Plotted graph of input 2 versus input 1 FLC

3.4. SIFLC heuristic tuning method

The gradient of the line is lambda (λ). The lambda (λ) was varied up and down heuristically. The variation of (λ)-tuned SIFLC result was then analyzed, and the best result was selected. Figure 9 shows the flow diagram of the heuristic tuning process.

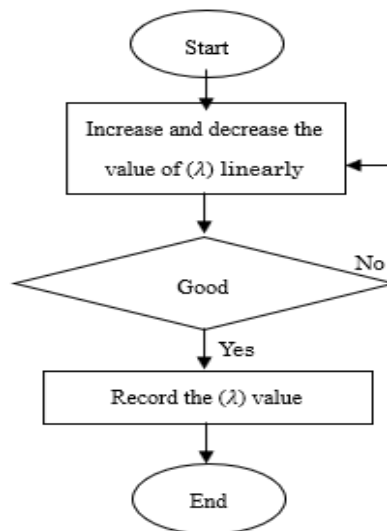


Figure 9. Flow diagram for SIFLC heuristic tuning

3.5. SIFLC GDA tuning method

GDA is an algorithm that iteratively runs until it reaches the minimum value of a function. The GDA is used to replace the heuristic lambda (λ) tuning for SIFLC. The objective function was obtained from the predicted output when compared to the input given. It is a simple mathematical method that is based on the differential equation where the initial point output was moved towards the targeted output by calculating the errors. Derivative of an objective function will determine the weight of the objective function for the next point. Two (2) important parameters are considered which are the direction of movement and the size of the step. The direction of movement is defined by the tangent of the initial point. The steepness of the tangent line also shows how near the point is to the minimum point and influences the choice of the learning rate. Figure 10 shows the flow diagram of the gradient descent algorithm [42]. The basic equation is shown in (5).

$$P^1 = P^0 - (Err * L) \tag{5}$$

Where,

P^0 = current position

P^1 = next position

Err = error ($P^0 - dP^0$)

L = Learning rate

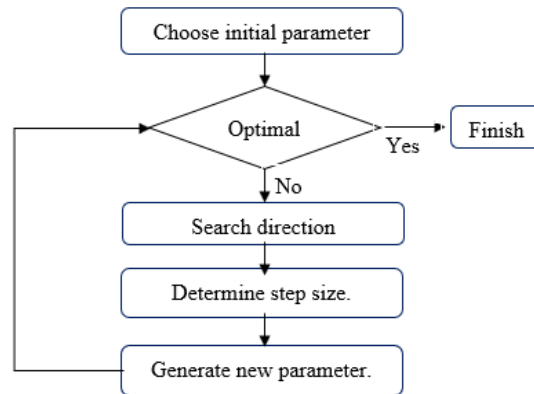


Figure 10. Flow diagram of gradient descent algorithm

3.6. SIFLC PSO tuning method

PSO was proposed by [43] in 1995. It is inspired by the behaviors of fishes schooling and birds flocking to search for foodstuff at a certain speed and position. The likeness is recognized between a particle and a swarm element [34], [44]. The particle movement is categorized by two factors: its current position x and velocity v , respectively. It has been useful effectively to a variety of optimization problems [45]–[47]. The particle swarm optimization algorithm is analyzed by using standard results from the dynamic theory [48]. The PSO algorithm begins by initializing the swarm randomly in the search space. Two consecutive iterations, t and $t + 1$ correspond to the position x of each particle that changes during the iterations by adding a new velocity v . The new velocity is estimated by summing an increment to the previous velocity value. The increment is a function of two components representing cognitive and social knowledge [49]. The cognitive knowledge of each particle is included by evaluating the difference between the current position x and its best position, $pbest$. The social knowledge of each particle is incorporated through the difference between its current position x and the best swarm global position achieved, $gbest$. He cognitive and social knowledge factors are multiplied by randomly uniform generated terms respectively [49]. Equation (6) shows the position vector while (7) shows the velocity vector. P in the equation is $pbest$ while G is $gbest$.

$$\vec{X}_i^{t+1} = \vec{X}_i^t + \vec{V}_i^{t+1} \quad (6)$$

$$\vec{V}_i^{t+1} = w\vec{V}_i^t + c_1r_1(\vec{P}_i^t - \vec{X}_i^t) + c_2r_2(\vec{G}^t - \vec{X}_i^t) \quad (7)$$

where,

\vec{X}_i^{t+1} = next position

\vec{X}_i^t = current position

\vec{V}_i^{t+1} = velocity

$w\vec{V}_i^t$ = inertia (maintain current movement direction)

w = weight (increase and decrease exploitation and exploration)

\vec{P}_i^t = personal best position

\vec{G}^t = group best position

c_1 & c_2 = the impact factors

r_1 & r_2 = the random value (0 – 1)

t = number of iteration

4. RESULTS

The output for all controllers designed was combined into one block diagram to compare the result. There are 6 signals analyzed which are step input, open-loop, closed-loop, PID, SIFLC heuristic, SIFLC GDA, and SIFLC PSO. Figure 11 shows the block diagram for the 6 signals investigated. From the block diagram, Scope 1 shows the 6 signals while scope 2 is used to compare between PSO result when the

equation was used (SIFLC PSO) and PSO result when the lookup table was used (SIFLC PSO1). This scope output shows an identical result (Figure 12) as expected. Figure 12 shows the output result for scope 1. In Figure 13, SIFLC GDA showed the most identical result to the step input given. It was then followed by the SIFLC heuristic. The SIFLC PSO showed improvement in the Tr but with a slight steady-state error. The PID showed some overshoot but no steady-state error. The output result is tabulated in Table 3.

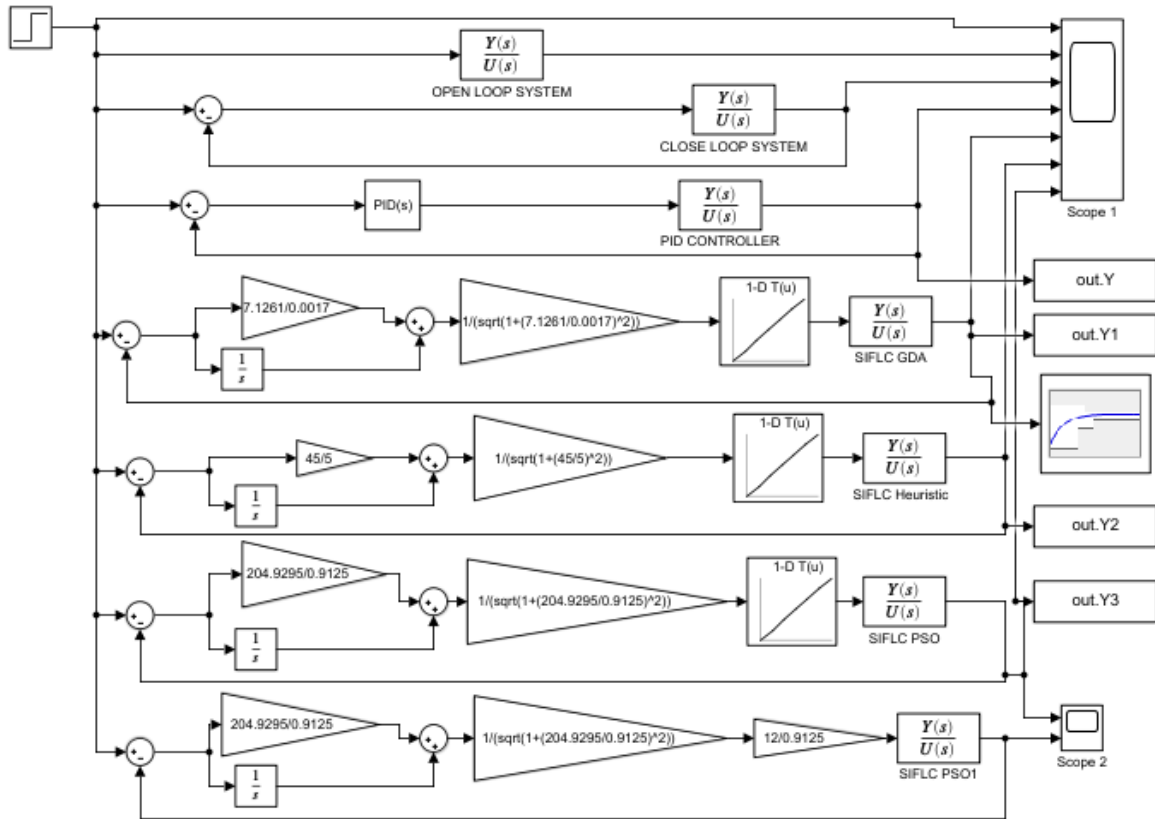


Figure 11. Block diagram for the 6 signals investigated

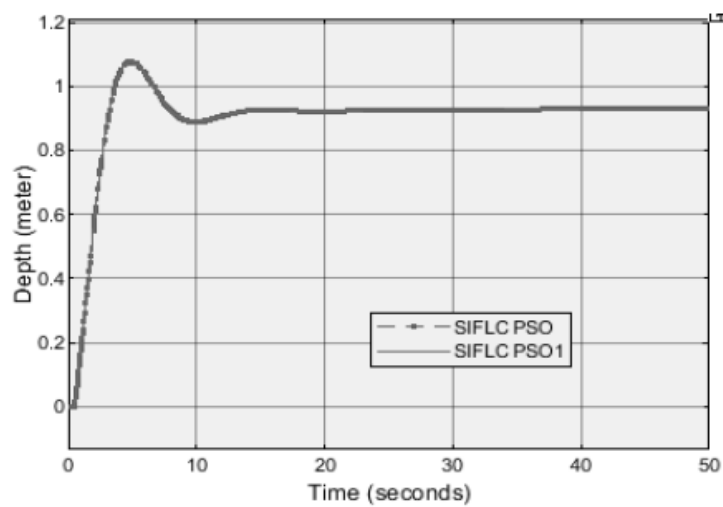


Figure 12. Comparison result between PSO result using command windows (SIFLC PSO) and Simulink (SIFLC PSO1)

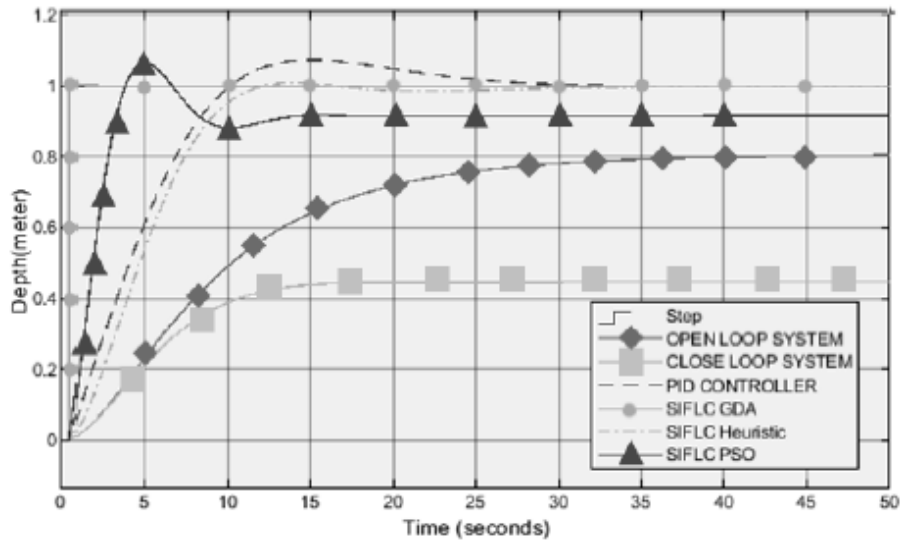


Figure 13. The output result of scope 1

Table 3. Output result of the controllers' implementation to ROV system

	PID	SIFLC Heuristic	SIFLC GDA	SIFLC PSO
Tr (s)	7.0665	7.2529	0.7992	2.3686
Ts (s)	24.6687	10.9736	0.9790	12.2348
%OS	7.3613	0.7988	0.1021	16.2368
SSE	0	0	0	0.1

From the bar chart in Figure 14, it is obvious that the SIFLC GDA control method showed the best result as it managed to get the lowest values for all the performance parameters. In terms of the rise time, Tr, the SIFLC GDA recorded 0.7992 s while the SIFLC heuristic was the worst at 7.2529 s, nearly ten times the SIFLC GDA record. The heuristic approach and PID approach had almost similar values at 7 s. These records showed that SIFLC GDA had the fastest response time to the change of input level. For settling time, Ts, the SIFLC GDA method recorded 0.9790 s and PID recorded the worst at 24.6687 s. Other than SIFLC GDA, other methods to stabilize the ROV took ten to twenty times longer. For the percentage of overshoot, the SIFLC GDA showed the best result which is at 0.1021%. It is then followed by the SIFLC heuristic (0.7988%), PID (7.3613%), and SIFLC PSO (16.2368%). These results showed that when the ROV was controlled using the SIFLC GDA method, the error to attain the target was very small. Figure 15 shows the comparison of SIFLC GDA with step signal. The SIFLC GDA looks nearly identical to the given step input.

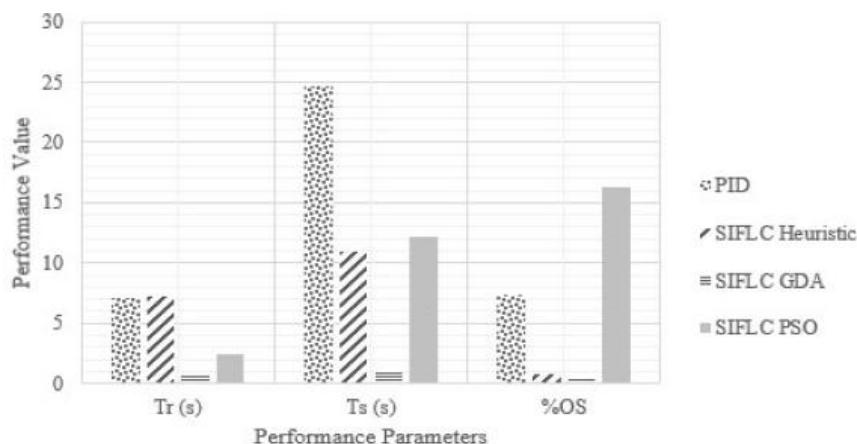


Figure 14. Bar chart of the performance parameters

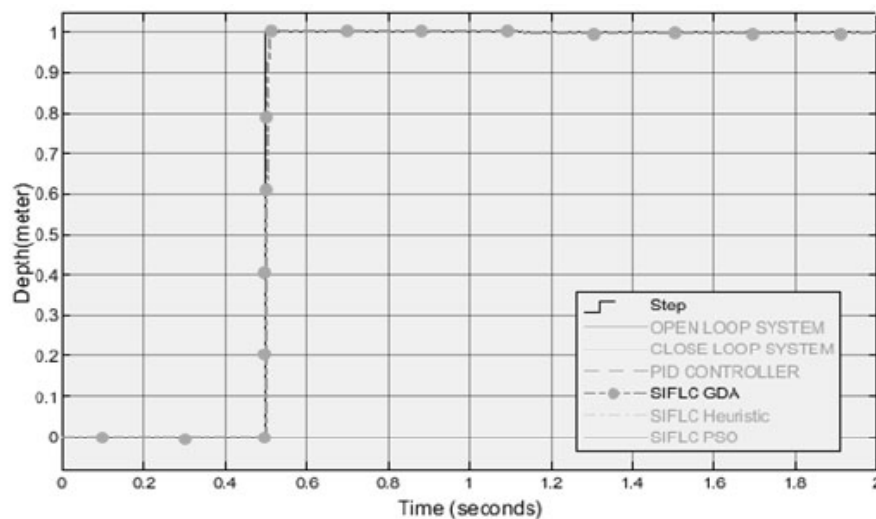


Figure 15. Comparison of SIFLC GDA with step signal

5. CONCLUSION

Two controllers with different tuning had been applied to the ROV depth system. A new tuning approach of SIFLC controller based on lambda (λ) is proposed and was compared with the basic PID controller. The SIFLC GDA showed the best result as it had the lowest values in all performance parameters investigated which were 0.1021% (OS), 0.7992 s (Tr) and 0.9790 s (Ts). The SIFLC PSO suffered from high overshoot and some steady-state error. The basic PID controller also suffered from some overshoot and had a long settling time. The SIFLC Heuristic had a better result compared to the PID controller in Ts and %OS however it was difficult to tune and required experience and time. The SIFLC GDA was able to obtain plausible results because it uses specific tuning of the objective function which is based on all parameters; %OS, Tr and Ts. On the other hand, the SIFLC PSO had a larger error compared to SIFLC GDA because the objective function used was the absolute mean error value. It doesn't specifically tune based on each parameter. From all results, it is proven that SIFLC lambda (λ) tuning approach successfully produced good output result. With the implementation of an optimization approach such as GDA and PSO, a better output result can be obtained. This happen because it optimizes the value of lambda (λ) to suit the input output result. The objective function selected in running the optimization approach also plays an important role in getting a good result. For future implementation, varieties of objective functions implementation can be studied and proposed to the system.

ACKNOWLEDGEMENTS

We wish to express our gratitude to honorable University, Universiti Teknikal Malaysia Melaka (UTeM). Special appreciation and gratitude to especially for Underwater Technology Research Group (UTeRG), Centre of Research and Innovation Management (CRIM), Center for Robotics and Industrial Automation (CeRIA) and Faculty of Electrical Engineering from UTeM for supporting this research.

REFERENCES




- [1] W. Chen, Y. Wei, H. Liu, and H. Zhang, "Bio-inspired sliding mode controller for ROV with disturbance observer," *2016 IEEE International Conference on Mechatronics and Automation, IEEE ICMA 2016*, pp. 599–604, 2016, doi: 10.1109/ICMA.2016.7558631.
- [2] M. Sanap, S. Chaudhari, C. Vartak, and P. Chimurkar, "HYDROBOT: An underwater surveillance swimming robot," *Proceedings - 2018 International Conference on Communication, Information and Computing Technology, ICCICT 2018*, vol. 2018-Janua, pp. 1–7, 2018, doi: 10.1109/ICCICT.2018.8325872.
- [3] F. Hoffmann and A. B. Kesel, "Biologically inspired optimization of underwater vehicles hull geometries and fin propulsion," *OCEANS 2019 - Marseille*, pp. 1–4, 2019, doi: 10.1109/oceanse.2019.8867134.
- [4] R. D. Christ and R. L. Wernli, *The ROV manual: a user's guide to remotely operated vehicles*. 2013.
- [5] H. Yu, C. Guo, and Z. Yan, "Globally finite-time stable three-dimensional trajectory-tracking control of underactuated UUVs," *Ocean Engineering*, vol. 189, no. March, p. 106329, 2019, doi: 10.1016/j.oceaneng.2019.106329.
- [6] C. Mai, S. Pedersen, L. Hansen, K. Jepsen, and Z. Yang, "Modeling and Control of Industrial ROV's for Semi-Autonomous Subsea Maintenance Services," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13686–13691, 2017, doi: 10.1016/j.ifacol.2017.08.2535.
- [7] A. Khadhraoui, L. Beji, S. Otmane, and A. Abichou, "Stabilizing control and human scale simulation of a submarine ROV navigation," *Ocean Engineering*, vol. 114, pp. 66–78, 2016, doi: 10.1016/j.oceaneng.2015.12.054.
- [8] Z. Chu, X. Xiang, D. Zhu, C. Luo, and D. Xie, "Adaptive trajectory tracking control for remotely operated vehicles considering thruster dynamics and saturation constraints," *ISA Transactions*, vol. 100, pp. 28–37, 2020, doi: 10.1016/j.isatra.2019.11.032.

- [9] Y. Tang, J. Wang, and C. Chen, "Development, Sea Trial and Application of Haidou Autonomous and Remotely-Operated Vehicle for Full-Depth Ocean Detection," *IEEE International Conference on Unmanned Systems (ICUS)*, pp. 73–77, 2020, doi: 10.1109/ICUS50048.2020.
- [10] C. S. Chin and W. P. Lin, "Robust Genetic Algorithm and Fuzzy Inference Mechanism Embedded in a Sliding-Mode Controller for an Uncertain Underwater Robot," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 655–666, 2018, doi: 10.1109/TMECH.2018.2806389.
- [11] K. W. Humphreys, D.E. and Watkinson, "Hydrodynamic Stability and Control Analyses of the UNH-EAVE, Autonomous Underwater Vehicle," 1982.
- [12] D. R. Yoerger, J. G. Cooke, and J. J. E. Slotine, "The Influence of Thruster Dynamics on Underwater Vehicle Behavior and Their Incorporation Into Control System Design," *IEEE Journal of Oceanic Engineering*, vol. 15, no. 3, pp. 167–178, 1990, doi: 10.1109/48.107145.
- [13] Y. Wang *et al.*, "Depth control of ROVs using time delay estimation with nonsingular terminal sliding mode," *OCEANS 2015 - MTS/IEEE Washington*, no. 51221004, 2016, doi: 10.23919/oceans.2015.7401804.
- [14] B. Huang and Q. Yang, "Double-loop sliding mode controller with a novel switching term for the trajectory tracking of work-class ROVs," *Ocean Engineering*, vol. 178, no. March, pp. 80–94, 2019, doi: 10.1016/j.oceaneng.2019.02.043.
- [15] N. E. Leonard, "Stability of a bottom-heavy underwater vehicle," *Automatica*, vol. 33, no. 3, pp. 331–346, 1997, doi: 10.1016/s0005-1098(96)00176-8.
- [16] D. R. Yoerger and W. Hole, "Robust Trajectory Control of Underwater Vehicles," pp. 184–197.
- [17] M. W. N. Azmi *et al.*, "Comparison of controllers design performance for underwater remotely operated vehicle (ROV) depth control," *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 3.14 Special Issue 14, pp. 419–423, 2018.
- [18] L. G. García-Valdovinos, F. Fonseca-Navarro, J. Aizpuru-Zinkunegi, T. Salgado-Jiménez, A. Gómez-Espinosa, and J. A. Cruz-Ledesma, "Neuro-sliding control for underwater rov's subject to unknown disturbances," *Sensors (Switzerland)*, vol. 19, no. 13, 2019, doi: 10.3390/s19132943.
- [19] J. Xu and N. Wang, "Optimization of ROV control based on genetic algorithm," *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans, OCEANS - Kobe 2018*, pp. 1–4, 2018, doi: 10.1109/OCEANSKOB.2018.8559384.
- [20] S. M. Zanolì and G. Conte, "Remotely operated vehicle depth control," *Control Engineering Practice*, vol. 11, no. 4, pp. 453–459, 2003, doi: 10.1016/S0967-0661(02)00013-8.
- [21] M. A. Grosebaugh, "Robust Control for Underwater Vehicle Systems with Time Delays," *IEEE Journal of Oceanic Engineering*, vol. 16, no. 1, pp. 146–151, 1991, doi: 10.1109/48.64894.
- [22] A. Faruq, S. S. Abdullah, M. Fauzi, and S. Nor, "Optimization of depth control for Unmanned Underwater Vehicle using surrogate modeling technique," *2011 4th International Conference on Modeling, Simulation and Applied Optimization, ICMSAO 2011*, 2011, doi: 10.1109/ICMSAO.2011.5775543.
- [23] K. Ishaque, "Intelligent Control of Diving System Of An Underwater Vehicle.," Universiti Teknologi Malaysia, 2009.
- [24] M. Santhakumar and T. Asokan, "A self-tuning proportional-integral-derivative controller for an autonomous underwater vehicle, Based on Taguchi method," *Journal of Computer Science*, vol. 6, no. 8, pp. 862–871, 2010, doi: 10.3844/jcssp.2010.862.871.
- [25] D. A. Smallwood and L. L. Whitcomb, "Model-Based Dynamic Positioning of Underwater Robotic Vehicles: Theory and Experiment," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 1, pp. 169–186, 2004, doi: 10.1109/JOE.2003.823312.
- [26] R. Mardiyanto and K. T. Trisuta, "Design of Aquatic Quadcopter with Hold Position Control and Gimbal Controller for capturing Underwater Video," *International Journal of Control and Automation*, vol. 11, no. 3, pp. 105–116, 2018, doi: 10.14257/ijca.2018.11.3.10.
- [27] E. H. Binugroho, Wafiqquochman, M. I. Mas'Udi, B. Setyawan, R. S. Dewanto, and D. Pramadihanto, "EROV: Depth and Balance Control for ROV Motion using Fuzzy PID Method," *IES 2019 - International Electronics Symposium: The Role of Techno-Intelligence in Creating an Open Energy System Towards Energy Democracy, Proceedings*, pp. 637–643, 2019, doi: 10.1109/ELECSYM.2019.8901673.
- [28] J. Guerrero, J. Torres, V. Creuze, A. Chemori, and E. Campos, "Saturation based nonlinear PID control for underwater vehicles: Design, stability analysis and experiments," *Mechatronics*, vol. 61, no. May 2018, pp. 96–105, 2019, doi: 10.1016/j.mechatronics.2019.06.006.
- [29] M. S. M. Aras *et al.*, "Depth control of an underwater remotely operated vehicle using neural network predictive control," *Jurnal Teknologi*, vol. 74, no. 9, pp. 85–93, 2015, doi: 10.11113/jt.v74.4811.
- [30] R. Hernández-Alvarado, L. G. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espinosa, and F. Fonseca-Navarro, "Neural network-based self-tuning PID control for underwater vehicles," *Sensors (Switzerland)*, vol. 16, no. 9, pp. 1–18, 2016, doi: 10.3390/s16091429.
- [31] N. Kumar and M. Rani, "An efficient hybrid approach for trajectory tracking control of autonomous underwater vehicles," *Applied Ocean Research*, vol. 95, no. October 2019, p. 102053, 2020, doi: 10.1016/j.apor.2020.102053.
- [32] M. Nizam Kamarudin, S. Md. Rozali, and A. Rashid Husain, "Observer-based output feedback control with linear quadratic performance," *Procedia Engineering*, vol. 53, pp. 233–240, 2013, doi: 10.1016/j.proeng.2013.02.031.
- [33] C. Yu, X. Xiang, F. Maurelli, Q. Zhang, R. Zhao, and G. Xu, "Onboard system of hybrid underwater robotic vehicles: Integrated software architecture and control algorithm," *Ocean Engineering*, vol. 187, no. April 2018, p. 106121, 2019, doi: 10.1016/j.oceaneng.2019.106121.
- [34] D. R. Parhi, P. B. Kumar, A. Chhotray, and H. Rawat, "Experimental and Simulation Analysis of Hybrid Ant Colony Experimental and Simulation Analysis of Hybrid Ant Colony Genetic Technique Using Ai Methodology for," no. May, 2018.
- [35] M. S. M. Aras and S. S. Abdullah, "Adaptive simplified fuzzy logic controller for depth control of underwater remotely operated vehicle," *Indian Journal of Geo-Marine Sciences*, vol. 44, no. 12, pp. 1995–2007, 2015.
- [36] M. Wahyuddin, N. Azmi, M. Shahrieel, M. Aras, M. Khairi, and M. Zambri, "Comparison of Controllers Design Performance for Underwater Remotely Operated Vehicle (ROV) Depth Control," no. January, 2018.
- [37] M. N. Taib, R. Adnan, and M. H. F. Rahiman, "Practical System Identification," Penerbit UiTM, 2007, p.600.
- [38] 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.
- [39] 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 Computing and Pattern Recognition*, pp. 711–715, 2009, doi: 10.1109/SoCPaR.2009.143.
- [40] M. S. M. Aras, A. M. Kassim, A. Khamis, S. S. Abdullah, and M. A. A. Aziz, "Tuning factor the single input fuzzy logic controller to improve the performances of depth control for underwater remotely operated vehicle," *Proceedings - UKSim-AMSS 7th European Modelling Symposium on Computer Modelling and Simulation, EMS 2013*, pp. 3–7, 2013, doi: 10.1109/EMS.2013.1.




- [41] K. Ishaque, S. S. Abdullah, S. M. Ayob, and Z. Salam, "Single input fuzzy logic controller for unmanned underwater vehicle," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 59, no. 1, pp. 87–100, 2010, doi: 10.1007/s10846-010-9395-x.
- [42] J. Zhu, E. Liu, S. Guo, and C. Xu, "A gradient optimization based PID tuning approach on quadrotor," *Proceedings of the 2015 27th Chinese Control and Decision Conference, CCDC 2015*, pp. 1588–1593, 2015, doi: 10.1109/CCDC.2015.7162172.
- [43] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.
- [44] E. J. Solteiro Pires, J. A. Tenreiro Machado, and P. B. de Moura Oliveira, "Particle Swarm Optimization: Dynamical Analysis through Fractional Calculus," *Particle Swarm Optimization*, 2009, doi: 10.5772/6761.
- [45] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 158–173, 2002, doi: 10.1109/4235.985692.
- [46] T. H. Kim, I. Maruta, and T. Sugie, "Robust PID controller tuning based on the constrained particle swarm optimization," *Automatica*, vol. 44, no. 4, pp. 1104–1110, 2008, doi: 10.1016/j.automatica.2007.08.017.
- [47] M. I. Solihin, Wahyudi, M. A. S. Kamal, and A. Legowo, "Optimal PID controller tuning of automatic gantry crane using PSO algorithm," *Proceeding of the 5th International Symposium on Mechatronics and its Applications, ISMA 2008*, pp. 25–29, 2008, doi: 10.1109/ISMA.2008.4648804.
- [48] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003, doi: 10.1016/S0020-0190(02)00447-7.
- [49] H. I. Jaafar *et al.*, "Optimal performance of a nonlinear gantry crane system via priority-based fitness scheme in binary PSO algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 53, no. 1, 2013, doi: 10.1088/1757-899X/53/1/012011.

BIOGRAPHIES OF AUTHORS






Fauzal Naim Zohedi    was born in Pulau Pinang, Malaysia in 1984 and graduated from department of Electrical & Electronic Engineering, Universiti Malaysia Pahang in 2008. He received his Master of Engineering (Industrial Electronics & Control) from Universiti Malaya in 2012. In 2008 to 2009 he was working as Device Engineer at ON Semiconductor, Senawang, Malaysia. He serves as a lecturer at Universiti Teknikal Malaysia Melaka (UTeM) from 2012 until 2019. Currently he is doing his PhD at Universiti Teknikal Malaysia Melaka (UTeM). His research interest covers control, instrumentation, automation, fuzzy application and underwater robotic. He can be contacted at email: fauzal@utem.edu.my.






Mohd Shahrieel Mohd Aras    owned both his Electrical Engineering Diploma and Degree from Universiti Teknologi MARA (UITM). Next, he pursued both master and PhD at Universiti Teknologi Malaysia (UTM). Currently he serves as Associate Professor at Universiti Teknikal Malaysia Melaka (UTeM). He is also Head of Mechatronics Department and Chair of IEEE Oceanic Engineering Society Malaysia Chapter. His research interest are Underwater Robotics, Fuzzy Logic, Neural Network, and System Identification. He can be contacted at email: shahrieel@utem.edu.my.



Hyreil Anuar Kasdirin    gained his bachelor's degree from Universiti Teknologi MARA (UITM) in Electrical Engineering. Next, he pursued his master and PhD in control system at Sheffield University, United Kingdom. Currently he is senior lecturer and serve as Director at Center of Academic Development, Universiti Teknikal Malaysia Melaka (UTeM). His research interest are Communications, Control Engineering, Artificial Intelligence and Optimization. He can be contacted at email: hyreil@utem.edu.my.



Nurdiana Nordin    gained her bachelor's degree from International Islamic University Malaysia in Mechatronic Engineering. Next, she pursued her master in Mechatronic Engineering at University of Siegen, Germany. She gained her PhD from University of Auckland in Mechatronic Engineering. Currently she is senior lecturer at Fakulti Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka (UTeM). Her research interest are Mechatronics, Machine Vision and Mobile Robotics. She can be contacted at email: nurdiana@utem.edu.my.