Websites:
http://www.sciencepub.net
http://www.sciencepub.net/report

Emails:
editor@sciencepub.net
reportopinion@gmail.com

Report and Opinion

MARSLAND PRESS
Multidisciplinary Academic Journal Publisher

Remotely Operated Underwater Vehicle Depth Control with New Lambda (λ) Tuning Approach of Single Input Fuzzy Logic using Gradient Descent Algorithm and Particle Swarm Optimization

Mustefa Jibril, Mesay Tadese, Nuriye Hassen

Msc, School of Electrical & Computer Engineering, Dire Dawa Institute of Technology, Dire Dawa, Ethiopia
mustefa.jibril@ddu.edu.et

Abstract: Underwater ROV is an important in underwater industries as well as safety purpose. It can dive deeper than human and can replace human in hazard underwater environment. ROV depth control is difficult due to hydrodynamic of the ROV itself and underwater environment. Overshoot in the depth control may cause damage to the ROV and its investigation location. This paper presenting a new tuning approach of SIFLC with GDA and PSO implementation for ROV depth control. The ROV was modelled 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 tuning approach 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 analysed. The result shows that SIFLC GDA output has the best transient result at 0.1021% (OS), 0.7992s (Tr) and 0.9790s (Ts).

[Mustefa Jibril, Mesay Tadese, Nuriye Hassen**. COVID-19, SARS-CoV-2 and variant.Remotely Operated Underwater Vehicle Depth Control with New Lambda (λ) Tuning Approach of Single Input Fuzzy Logic using Gradient Descent Algorithm and Particle Swarm Optimization.** *Rep Opinion* 2021;13(12):72-84]. ISSN 1553-9873(print);ISSN 2375-7205 (online). http://www.sciencepub.net/report. 6. doi:10.7537/marsroj131221.06

Keywords: Remotely Operated Vehicle, Fuzzy Logic Controller, Single Input Fuzzy Logic, Particle Swarm Optimization, Gradient Descent Algorithm

Introduction

In underwater engineering field, ROV plays important role for underwater observation, investigation and inspection [1]–[3]. Especially in oil and gas industry, ROV is used to do underwater pipe inspection as well as repairing job. ROV normally suffered from problems include pose recovery or station keeping, under actuated condition, coupling issues and communication technique [4]. This research paper was focusing on the ROV depth control or station keeping. Station keeping at certain depth is very important for underwater exploration and inspection mention in paper [5]–[7].Controlling ROV is difficult because of unexpected and unpredictable [4], [8] underwater environment. 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 UUV satisfactorily [16]. Even for a one axis motion for example vertical motion or heave motion, consistent performance for a desirable range is required. Overshoot in the system cannot be considered as it can harm the ROV or its inspection location [14], [17]–[20]. It is best to have as least as possible overshoot in the ROV system.

There many controllers designed by researcher to cater this problem. There are Proportional, Integral and Derivative (PID) based controller and artificial based controller. PID is a simple control technique that has been universally used because of the simplicity of implementation in 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. There are also existence of parameter configuration contradictory between different control performance such as between rise time and overshoot. Paper [21]–[25] have implement successfully implement PID controller for tracking purpose for Unmanned Underwater Vehicle (UUV) while paper [17], [26] successfully implement to ROV. Normally, PID controller was used as basic controller to be compared with other complex controller such as paper [27], [28] and [6]. The PID was hard to be tuned to cope with non-linear nature of underwater environment. The PID produce high overshoot and high steady state error. PID controller was not able to cope with underwater wavy environment.

Due to limitation of PID, artificial intelligent based controller such as Fuzzy Logic Controller (FLC) and Artificial Neural Network

(ANN) that had been introduced to control ROV. ANN was used by paper [29] to control the depth of ROV. ANN was used to predict the performance of the ROV depth system based on previous input and minimize the cost function. Then, the best input is suggested. The ANN result shows superior result compare to other controllers that were experimented. Paper [30] and paper [31] also implement ANN based for ROV system. Paper [30] used ANN to tuned PID and adapt with the depth changing of ROV. Difference from Paper [30], paper [31] implemented radial basis function neural network (RBFNN) for trajectory tracking for autonomous underwater vehicle (AUV). Both shows good result. The downside of ANN was long computational time that may lead to lagging problem.

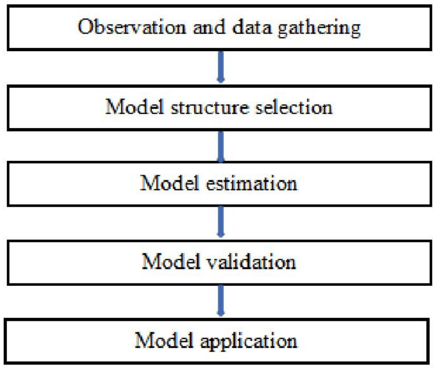
Another artificial intelligent based controller for ROV system is the fuzzy logic controller (FLC). In [21] and [23], the authors successfully applied FLC to ROV while in [32] FLC was successfully applied to AUV. The FLC controller can cope with not well-known mathematical model system. Implementation of FLC ease the need of precise and complex hydrodynamic modelling of the vehicle. In paper [33], FLC was successfully used to tuned PID controller for underwater vehicle. Even with the adaptability advantage, FLC poses its own level of complexity.

Simplified single input fuzzy logic controller (SIFLC) is proposed to control the depth of ROV. Paper [34], [35] revealed that SIFLC has excellent performance, and it exactly resembles conventional FLC transient response. SIFLC reduce the input of conventional FLC into single input single output (SISO) system. Normally, trial an error (heuristic) method was used to find the optimum parameter. Consequently, it takes more time execution to find the optimum parameters.

This paper presenting a new tuning approach of SIFLC with gradient descent algorithm (GDA) and particle swarm optimization (PSO) implementation for ROV depth control. The ROV was modelled 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 tuning approach which are try and error (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 analysed. In terms of depth control, the overshoot (%OS) may damage the ROV or its investigation place [14], [18]–[20], [36]. The time rise (Tr) shows the time taken to get to desired point while the settling time is the time ROV stabilize at steady state.

System Modelling

In this paper, the ROV was modelled using system identification (SI) method. 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 SI approach. The steps are observation and data gathering, model structure selection, model estimation, model validation and model application.[37]


Observation and data gathering

Model structure selection

Model estimation

Model validation

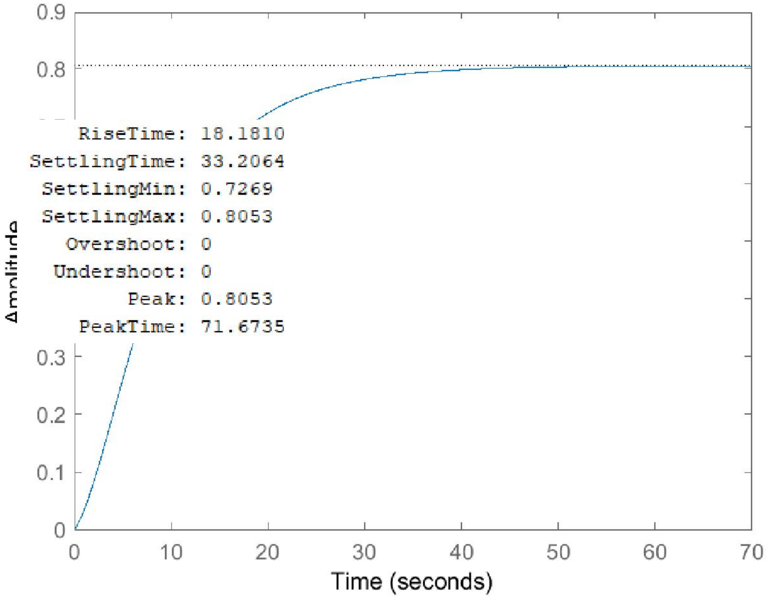
Model application

Figure 1: System identification approach for modelling of ROV

Start with system observation and data gathering, ROV system and the data is gathered. Two sets of data are needed: training and validation data. In this research project, multi-sine signal was used to get the experimental data for training and validation. The input and output data were recorded, and MATLAB is used to get the transfer function of the system. two set of data needed where one (1) set for training and another for validation. The input given to ROV system can be pulse, steps, Random Binary Sequence (RBS), Pseudo Random Binary (PRBS), m-level Pseudo Random (m-PRS) and multi-sine [37]. In this project, multi-sine input was given to the system. In the MATLAB system, the Instrument Variable (IV) approach was selected. Next, the selected model structure is implemented for model estimation and model validation to generate a ROV model. Lastly, the model generated is used to design ROV controller. To gain an ideal result, the experiment was conducted in a controlled environment. Disturbance was not considered. Instrument variable approach: IV combined with 3 poles and 2 zeros transfer function was selected. The best fitting match was 96.43% is acceptable because within 80% to 99% best fits. The transfer function generated shown as equation 1 below.

$$H(S) = \frac{0.02332s^2+0.04058s+0.01126}{s^3+0.7\ 114s^2+0.1861s+0.01398} \qquad (1)$$

The generated output transient response was shown in Figure 2. The output result has no overshoot, 18.18s of Tr, 33.21s Ts and 0.1947 steady state error (sse). The result is not good as it has great steady state error which is 19.47% of the input given. Even though it does not have any overshoot, the system takes a bit long time to rise and st Tr and Ts value.



```
   RiseTime: 18.1810
SettlingTime: 33.2064
 SettlingMin: 0.7269
 SettlingMax: 0.8053
   Overshoot: 0
  Undershoot: 0
        Peak: 0.8053
    PeakTime: 71.6735
```

Figure 2: Transient response of the ROV model

This generated modelled is then simulated in MATLAB Simulink as closed loop system shown in the block diagram in figure 3 below.

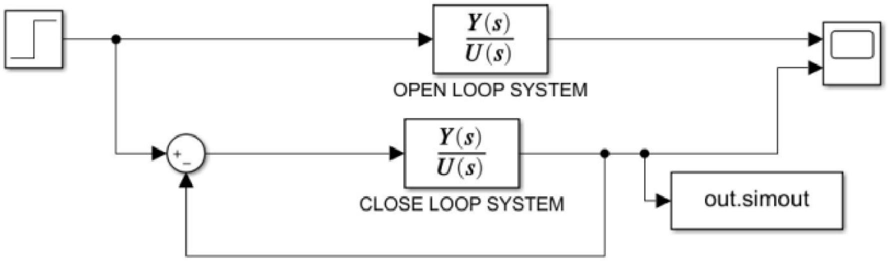


Figure 3: MATLAB Simulink close loop block diagram

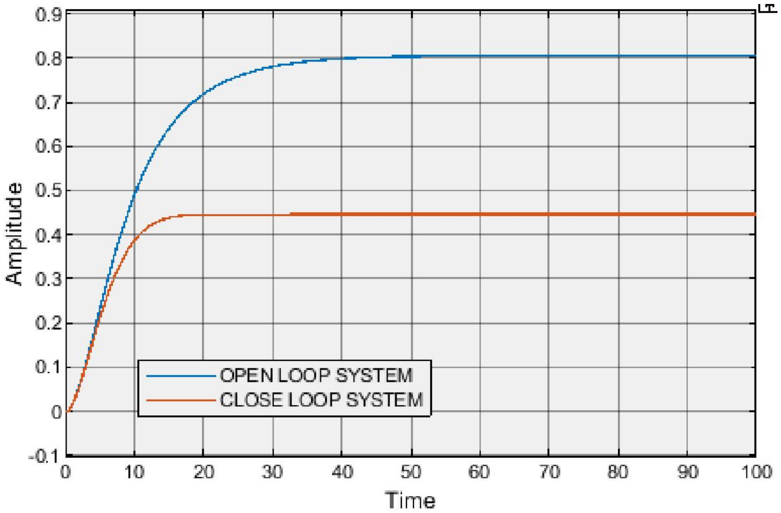


Figure 4: Open loop and close loop transient output comparison

From figure 4, the close loop has faster Tr (9.07s) and Ts (14.76s) compare to open loop result but high steady state error up to 55.55% from the input given to the system. From the output result, controller need to be applied to get better output response.

Proposed Controller Design

In this paper PID controller was designed using auto tuning provided by MATLAB Simulink. The SIFLC controller was designed and tuned using heuristic, GDA and PSO. PID was used as a basic controller to be compare with SIFLC controller designed.

PID Controller

As mentioned previously, PID controller is the basic controller applied to ROV system. The P, I, and D blocks were put in parallel in front of the plan to control the system. The P counter the direct error; the I indicate the total errors in the system while D shows how fast to the errors happen. The P controller will make the response faster but intend to produce overshoot. The I controller tend to eliminate SSE while the D controller decrease overshoot. The PID controller block diagram is shown as in Figure 5. The PID was tuned using automatic tuning in MATLAB Simulink [38].

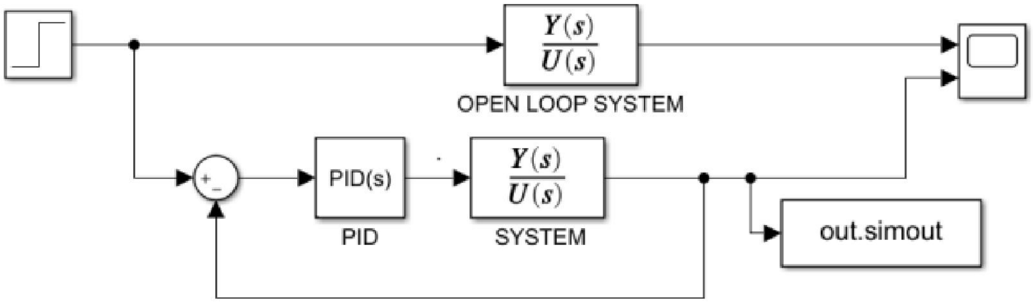
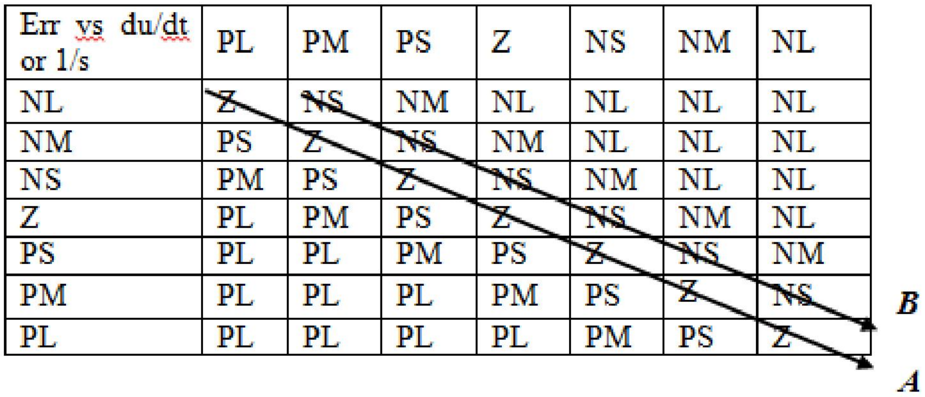


Figure 5: PID controller block diagram

SIFLC controller

SIFLC controller is designed based on conventional FLC designed. The normal FLC table; Table 1, is manipulated using sign distance method (SDM) which reduced the rules table to a one-dimensional array [39], [40]. From table, it can be seen there is consistent pattern in the decision making of the FLC output.

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~~

From table 1, 2 diagonal lines were created which named as A and B. ‘d’ is distance between A and B given by equation 2. Figure 5 shows the derivation of d, which is distance between point, Q and point, P.

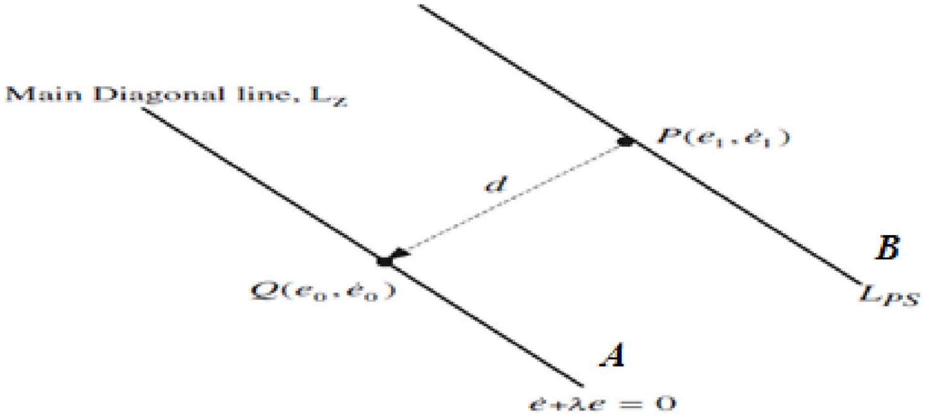


Figure 6: Derivation of d, distance between point Q and P[40]

$$d = \frac{w+Z_e\lambda}{\sqrt{1+\lambda^2}} = \frac{w}{\sqrt{1+\lambda^2}} + \frac{Z_e\lambda}{\sqrt{1+\lambda^2}} \qquad (2)$$

$$\dot{e} + \lambda e = 0 \qquad (3)$$

$$\therefore \lambda = -\frac{\dot{e}}{e} \qquad (4)$$

The conventional FLC table is now reduced to table 2 where diagonal line was represent 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.

Table 2: Reduced FLC table using SDM

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

This input output of SIFLC can be replaced by lookup table. SIFLC was then tuned using proposed lambda (λ) tuning method. The value of (λ) is varies up and down to get the best output result. The (λ) 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 7.

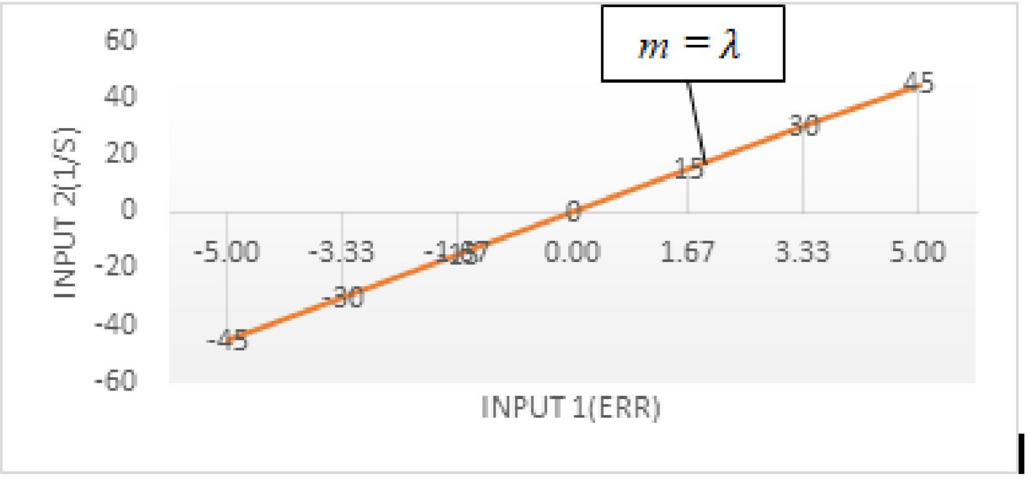


Figure 7: Plotted graph of input 2 versus input 1 FLC.

SIFLC Heuristic Tuning Method

The gradient of the line is lambda (λ). The varying of (λ) SIFLC result was then analysed, and the best result was selected. The varying of lambda (λ) up and down experimentally is called heuristic method. Figure 8 shows the flow diagram of the heuristic tuning process.

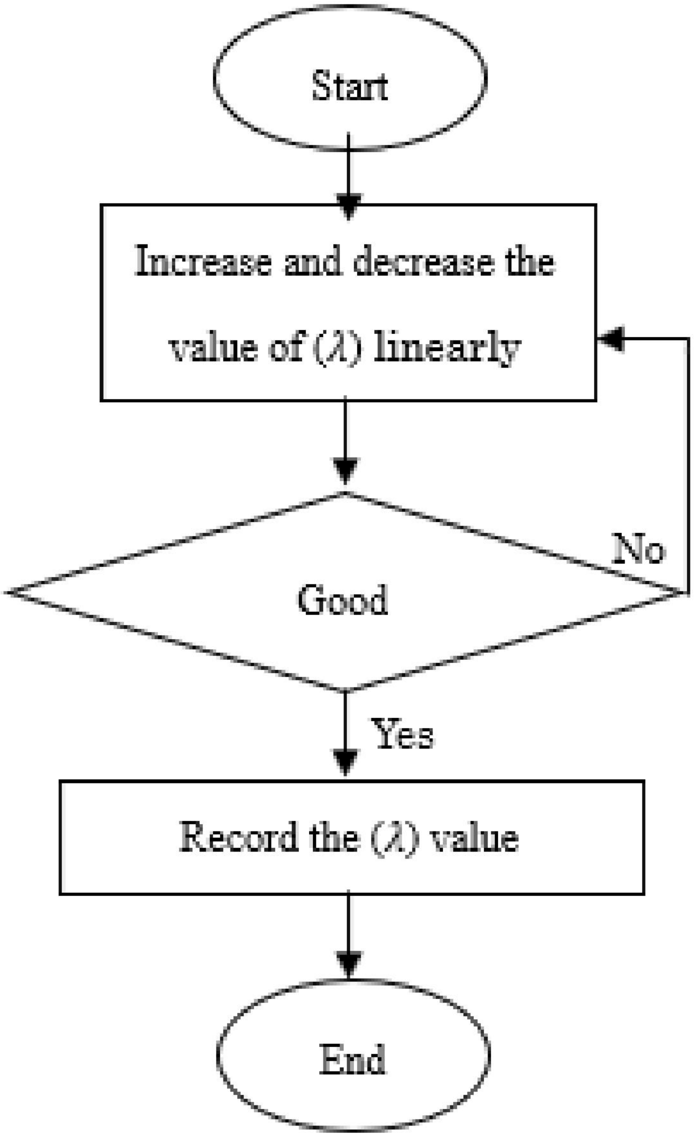

Start
Increase and decrease the value of (λ) linearly
Good
No
Yes
Record the (λ) value
End

Fig. 8: Flow diagram for SIFLC heuristic tuning

As shown in figure 8, the varying of (λ) value or the gradient was done until the best result generated. It takes much time and experience of controller designer is tested.

SIFLC GDA Tuning Method

GDA is an algorithm that iteratively run until it manages to get the minimum of a function. The GDA is used to replace the heuristic lambda (λ) tuning for SIFLC. The objective function was from the predicted output compared to input given. It is a simple mathematical method that is based on differentiation equation where the initial point output was move towards the targeted output by calculating the errors. Two (2) important parameters need to be considered which are direction of movement and the size of step need to be used. The direction of movement defines by the tangential of the initial point. The sharpness of the tangent line also shows how near the point to the minimum point and how to decide the learning rate that should be selected. Figure 9 shows the flow diagram of gradient descent algorithm [41]. From figure 9, the GDA will keep on running until the optimum condition is generated or the iteration reach.

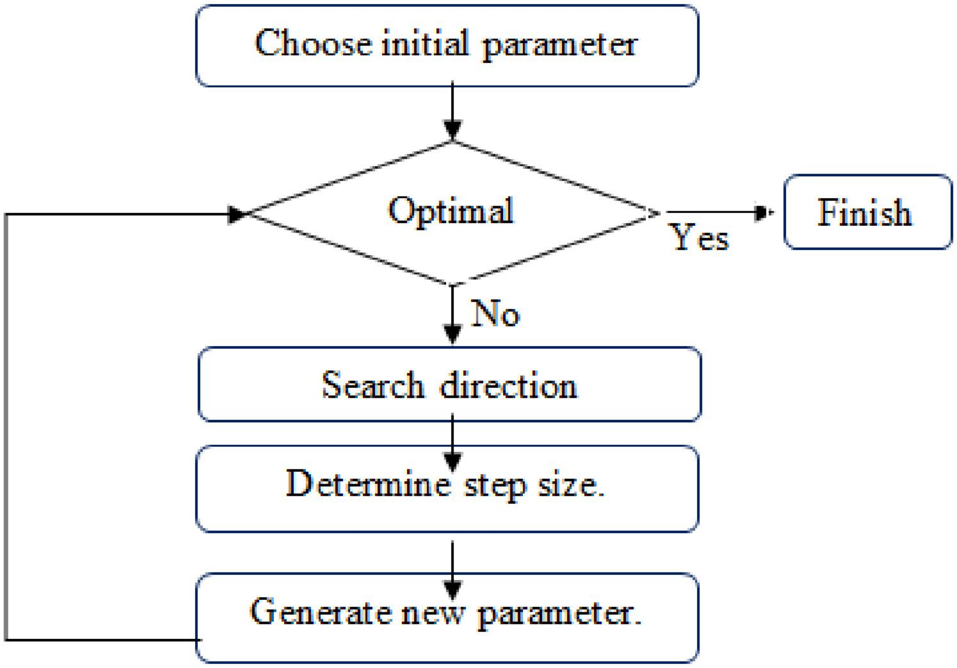


Figure 9: Flow diagram of gradient descent algorithm

SIFLC PSO Tuning Method

PSO was proposed by [42] in 1995. It is inspired by behaviours of fish schooling and bird flocking to search for foodstuff at a certain speed and position. The likeness is recognized between a particle and a swarm element [34][43]. 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 [44]–[46]. The particle swarm optimization algorithm is analysed by using standard results from the dynamic theory [47]. 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 changed 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 the cognitive and the social knowledge [48]. 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. The cognitive and social knowledge factors are multiplied by randomly uniform generated terms ϕ1 and ϕ2, respectively [48]. Equation (5) shows the position vector while equation 6 shows the velocity vector. P in equation is PBEST while the G is GBEST.

$$\overrightarrow{X_i^{t+1}} = \overrightarrow{X_i^t} + \overrightarrow{V_i^{t+1}} \qquad (5)$$

$$\overrightarrow{V_i^{t+1}} = w\overrightarrow{V_i^t} + c_1 r_1 \left(\overrightarrow{P_i^t} - \overrightarrow{X_i^t}\right) + c_2 r_2 \left(\overrightarrow{G^t} - \overrightarrow{X_i^t}\right) \qquad (6)$$

Result and Discussion

All controllers designed was combined into one block diagram to compare the result. There are 6 signals analysed which are step input, open loop, close loop, PID, SIFLC heuristic, SIFLC GDA and SIFLC PSO. Figure 10 shows the block diagram for the 6 signals investigated.

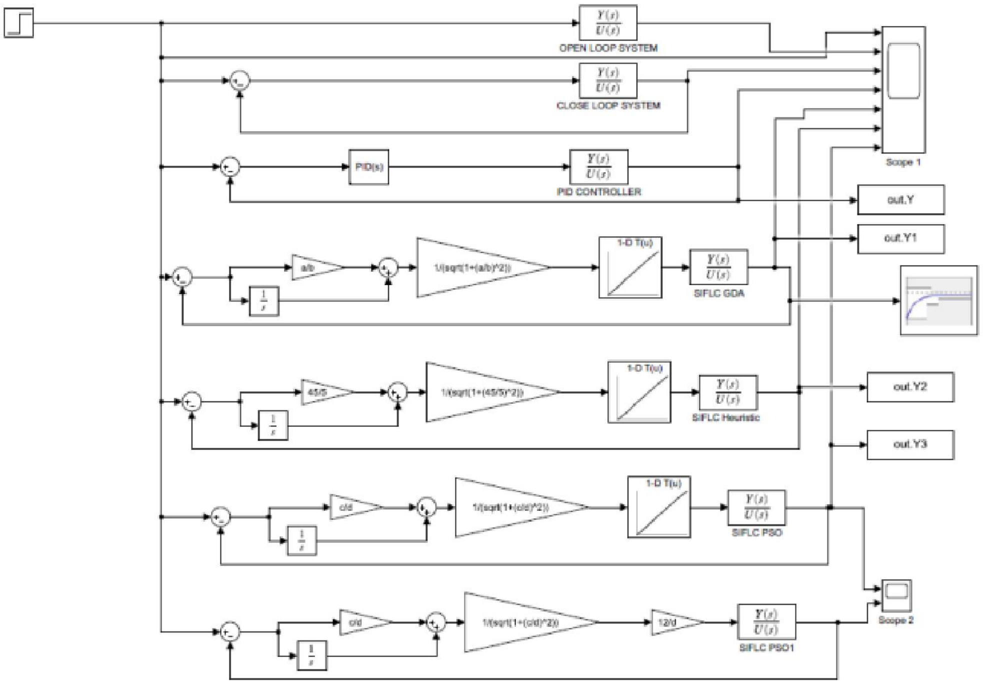


Figure 10: Block diagram for the 6 signals investigated.

From block diagram, Scope 1 shows the 6 signals while Scope 2 used to compare between PSO result equation based (SIFLC PSO) and lookup table Simulink (SIFLC PSO1). This scope output shows identical result (figure 11). Figure 12 shows the output result for Scope 1.

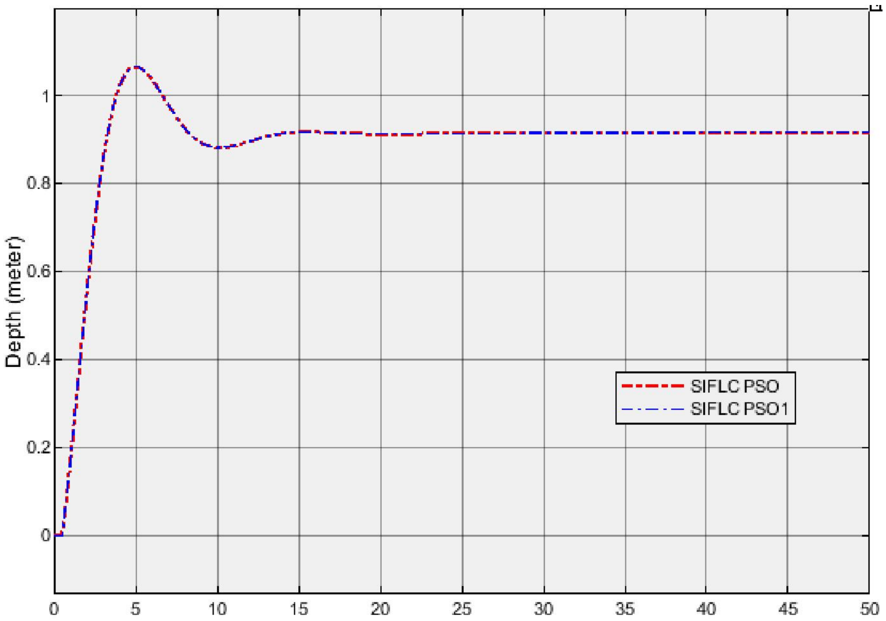


Figure 11: Comparison result between PSO result using command windows (SIFLC PSO) and Simulink (SIFLC PSO1).

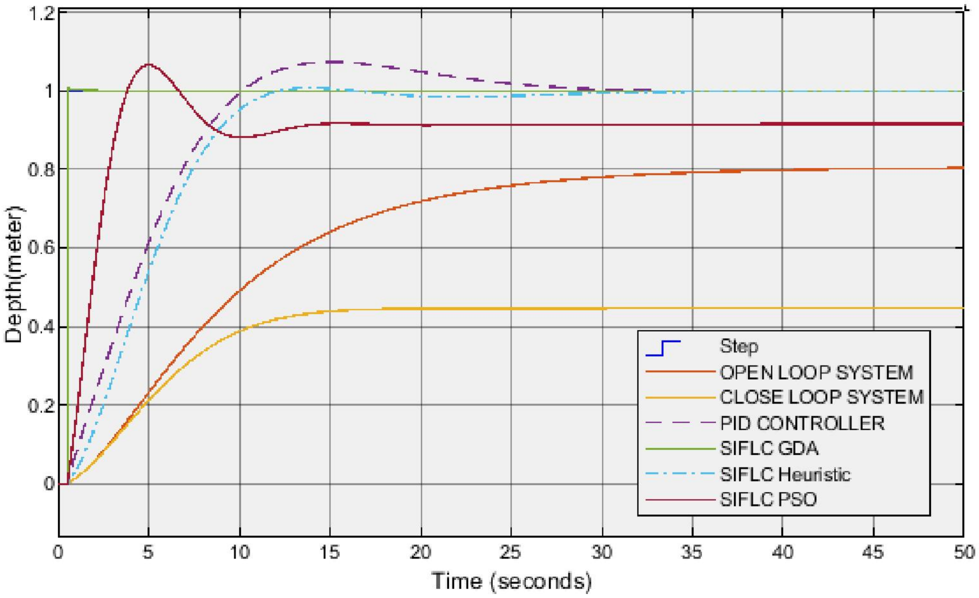


Figure 12: The output result of Scope 1

In figure 12, SIFLC GDA shows the most identical result to the step input given. It is then followed by SIFLC heuristic. The SIFLC PSO shows improvement in the Tr but a bit steady state error. The PID shows a bit overshoot but no steady state error. The output result is tabulated in Table 3 below.

Table 3: Output result of the controller's 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

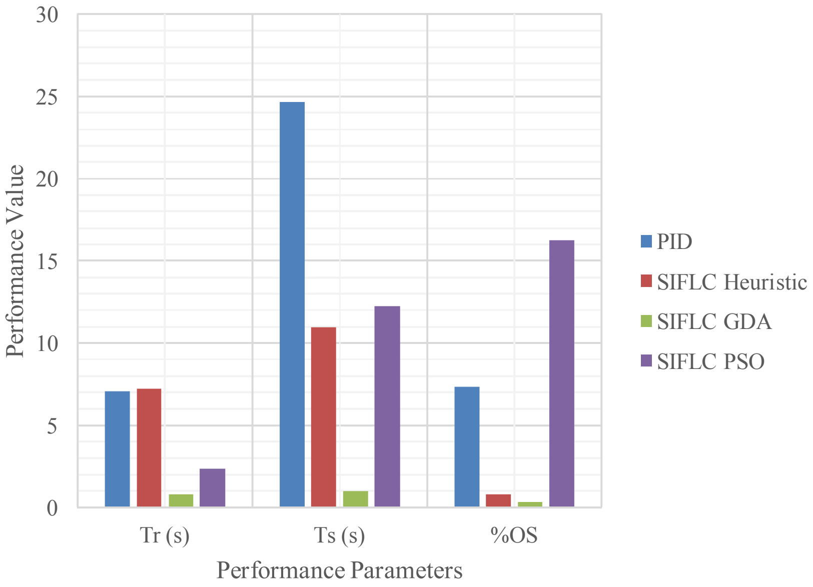


Figure 13: Bar chart of the performance parameters

From the bar chart in figure 13, it is obvious that SIFLC GDA shows the best and balance result as it manages to get the lowest errors at all parameters. For Tr (s), SIFLC GDA shows 0.7992s result. Next to it are SIFLC PSO (2.366s), PID (7.066s) and SIFLC Heuristic (7.2592s). Heuristic approach and PID approach have almost similar value for Tr (s) at 7s. For Ts(s), next to SIFLC GDA (0.9790s) are SIFLC Heuristic (10.973s), SIFLC PSO (12.2348s) and PID (24.6687s). For the last parameter (overshoot), SIFLC GDA shows the best result which is 0.1021%. It is then followed by SIFLC heuristic (0.7988%), PID (7.3613%) and SIFLC PSO (16.2368%). Figure 14 shows the comparison of SIFLC GDA with step signal. The SIFLC GDA looks nearly identical to the given step input.

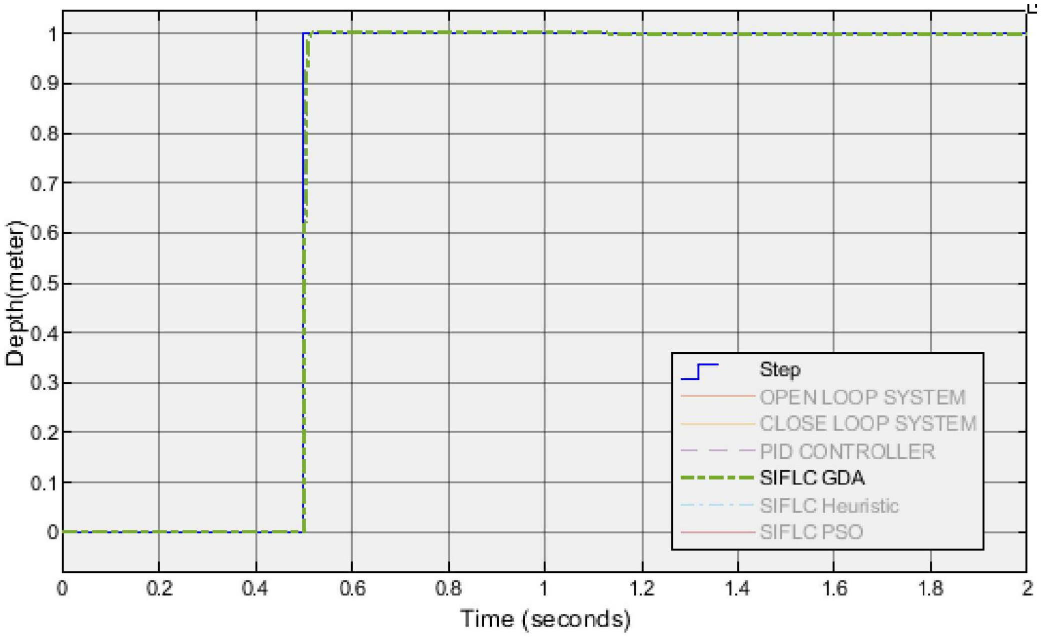


Figure 14: Comparison of SIFLC GDA with step signal.

Conclusion

Two controllers with difference tuning had been applied to ROV depth system. New tuning approach of SIFLC controller based on lambda (λ) is proposed and compare with basic PID controller. The SIFLC GDA shows the best and balance result as it has the lowest errors in all parameters investigated. The SIFLC PSO suffer from high overshoot and a bit steady state error. The basic PID controller can be excepted but suffered from a bit overshoot and long settling time, Ts (s). For SIFLC Heuristic, the result can also be excepted as it has better result compare to PID controller in Ts and %OS. The problem with SIFLC heuristic is experience controller designer is needed and it takes much time to tune it. The SIFLC GDA get good result because it used specific tuning of objective function based on all parameters. SIFLC PSO get higher error compare to SIFLC GDA because the objective functioned used was absolute mean error value. From all results, it is proven that SIFLC lambda (λ) tuning approach successfully produced good output result. With the implementation of optimization approach such as GDA and PSO, the better output result can be obtained. The objective function selected in running the optimization approach also plays important roles in getting a good result. For future implementation, varieties of objective functions implementation can be studied and proposed to the system.

Reference

[1] W. Chen, Y. Wei, H. Liu, and H. Zhang, "Bio-inspired sliding mode controller for ROV with disturbance observer," *2016 IEEE Int. Conf. Mechatronics Autom. 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," *Proc. - 2018 Int. Conf. Commun. Inf. Comput. Technol. 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," *Ocean. 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 Eng.*, 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 Eng.*, 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 Trans.*, 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 Int. Conf. Unmanned Syst.*, 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 Trans. 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 J. Ocean. Eng.*, 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," *Ocean. 2015 - MTS/IEEE Washingt.*, 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 Eng.*, 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," *Int. J. Eng. Technol.*, 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 Ocean. - MTS/IEEE Kobe Techno-Oceans, Ocean. - Kobe 2018*, pp. 1–4, 2018, doi: 10.1109/OCEANSKOBE.2018.8559384.

[20] S. M. Zanoli and G. Conte, "Remotely operated vehicle depth control," *Control Eng. Pract.*, vol. 11, no. 4, pp. 453–459, 2003, doi: 10.1016/S0967-0661(02)00013-8.

[21] M. A. Grosenbaugh, "Robust Control for Underwater Vehicle Systems with Time Delays," *IEEE J. Ocean. Eng.*, 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 Int. Conf. Model. Simul. Appl. Optim. 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," *J. Comput. Sci.*, 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 J. Ocean. Eng.*, 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," *Int. J. Control Autom.*, vol. 11, no. 3, pp. 105–116, 2018, doi: 10.14257/ijca.2018.11.3.10.

[27] E. H. Binugroho, Wafiqqurochman, 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 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc.*, 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," *J. Teknol.*, 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," *Appl. Ocean Res.*, 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 Eng.*, 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 Eng.*, 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 J. Geo-Marine Sci.*, 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. H. F. Taib, M. N., Adnan, R. and 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, 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," *Proc. - UKSim-AMSS 7th Eur. Model. Symp. Comput. Model. Simulation, EMS 2013*, pp. 3–7, 2013, doi: 10.1109/EMS.2013.1.

[40] 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.

[41] J. Zhu, E. Liu, S. Guo, and C. Xu, "A gradient optimization based PID tuning approach on quadrotor," *Proc. 2015 27th Chinese Control Decis. Conf. CCDC 2015*, pp. 1588–1593, 2015, doi: 10.1109/CCDC.2015.7162172.

[42] R. Kennedy, J. and Eberhart, "Particle Swarm Optimization," in *Proceedings of the IEEE International Conference on Neural Networks.*, 1995, pp. 1942–1948.

[43] E. J. Solteiro Pires, J. A. Tenreiro Machado, and P. B. de Moura Oliveir, "Particle Swarm Optimization: Dynamical Analysis through Fractional Calculus," *Part. Swarm Optim.*, 2009, doi: 10.5772/6761.

[44] J. Clerc, M. and Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 158–173, 2002.

[45] 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.

[46] M. I. Solihin, Wahyudi, M. A. S. Kamal,

and A. Legowo, “Optimal PID controller tuning of automatic gantry crane using PSO algorithm,” *Proceeding 5th Int. Symp. Mechatronics its Appl. ISMA 2008*, pp. 25–29, 2008, doi: 10.1109/ISMA.2008.4648804.

[47] I. C. Trelea, “The particle swarm optimization algorithm: Convergence analysis and parameter selection,” *Inf. Process. Lett.*, vol. 85, no. 6, pp. 317–325, 2003, doi: 10.1016/S0020-0190(02)00447-7.

[48] H. I. Jaafar *et al.*, “Optimal performance of a nonlinear gantry crane system via priority-based fitness scheme in binary PSO algorithm,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 53, no. 1, 2013, doi: 10.1088/1757-899X/53/1/012011.

11/21/2021