International Review on Modelling and Simulations (I.RE.MO.S.), Vol. 6, No.5
October 2013

A Comparison Study Between Two Algorithms Particle Swarm Optimization for Depth Control of Underwater Remotely Operated Vehicle

M.S.M. Aras[1], S.S. Abdullah[2] , H.I Jaafar[3] , Razilah A.R.[4], Arfah Ahmad[5]

Abstract – This paper investigates two algorithms based on particle swarm optimization (PSO) to obtain optimum parameter. In this research, an improved PSO algorithm using a priority-based fitness PSO (PFPSO) and priority-based fitness binary PSO (PFBPSO) approach. This comparison study between two algorithms applied on underwater Remotely Operated Vehicle for depth control. Two parameters in Single Input Fuzzy Logic Controller will tune using two algorithms to obtain optimum parameter. There are two parameters to be tuned namely the break point and slope for the piecewise linear or slope for the linear approximation. The study also covered a comparison for time execution for every time the parameter tuning was done. Based on the results the PFBPSO gives a consistent value of optimum parameter and time execution very fast. The best optimum parameter of SIFLC determined using 2 methods such that average of optimum parameter and intersection of y-axis. The PFBPSO gives comparative results in term of two parameters and time execution very fast compared with improved PSO.

Keywords*:* Priority Fitness PSO, Priority Fitness Binary PSO, Optimum parameter, Single Input FLC, time execution.

Nomenclature

X_i	the particle position
K1	the break point for the piecewise linear of SIFLC
K2	the slope for the piecewise linear of SIFLC
x_{max}	the maximum values in the search space boundary
x_{min}	the minimum values in the search space boundary
r_1	random function values
r_2	random function values
c_1	cognitive component
c_2	social component
ω	to balance between local and global search capabilities
H	hypothesis testing

I. Introduction

Particle Swarm Optimization (PSO) is one of most excellent optimization technique to obtain optimum parameter for a system. PSO is one of the artificial intelligence families that were introduced by [1]. The basic PSO is developed based on behaviors of fish schooling and bird flocking in order to search and move to the food with a certain speed and position. It has been applied successfully to a wide variety of optimization problems by [2- 9]. In this research, an improved PSO algorithm using a priority-based fitness PSO approach or called as (PFPSO) and priority-based fitness binary PSO (PFBPSO) is proposed for tuning of Single Input Fuzzy Logic Controller parameters to depth control of underwater Remotely Operated Underwater Vehicle (ROV). These two techniques will obtain the optimum parameter for SIFLC to give the best performances on system response. For simplicity describes details PFPSO in short PSO while PFBPSO in short BPSO.

The proposed techniques are continuing research based on [10] and [11] where the focus was made on Single Input Fuzzy Logic Controller tuning uses PSO based on Simple Feed Forward and Output Feedback Observer for Underwater Remotely Operated Vehicle (ROV). The design and specification of the ROV can be referred to [12-14]. The objective is to improve system performance in terms of overshoot, rise time, and settling time in system response. In this study the improvement will be one parameter of the Single Input Fuzzy Logic Controller. The parameters for SIFLC will be tuned by PSO and BPSO. Figure 1 shows the proposed technique that is an Output Feedback Observer Tuning by using Single Input Fuzzy Logic Controller (SIFLC) for underwater ROV for depth control.

This paper is organized as follows. In section 1, a brief introduction to Particle Swarm Optimization and the ROV are discussed. Section 2 describes the Priority-based Fitness Particle Swarm Optimization (PFPSO). The Section 3 presents the Priority-based Fitness Binary Particle Swarm Optimization (PFBPSO) while Section 4 describes the results and discussion. Finally, the final remarks are elucidated in Section 5.

Manuscript received and revised July 2013, accepted October 2013

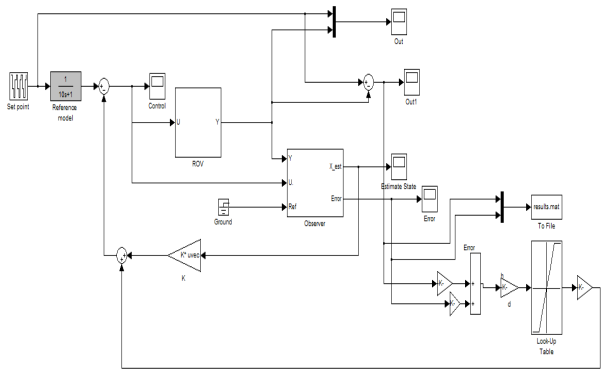


Fig. 1: Output Feedback Observer Tuning using Single Input Fuzzy Logic Controller

II. Priority Based Fitness PSO (PFPSO)

This PFPSO or PSO algorithm is adapted from [6] but the implementation is for the conventional PID controller to control nonlinear gantry. In this research, overshoot, OS is set as the highest priority, followed by settling time, Ts and steady state error, SSE. The objective is to develop a controller that can guarantee the suppression or eliminate overshoot in the system response. For depth control, overshoot in the system response is particularly dangerous. Clearly an overshoot in the ROV vertical trajectory may cause damages to both the ROV and the inspected structure such as operating in cluttered environments. Figure 2 illustrates the priority-based fitness approach where the $\boldsymbol{P_{BEST}}$ and $\boldsymbol{G_{BEST}}$ are updated according to the priority [8]: OS, Ts and SSE.

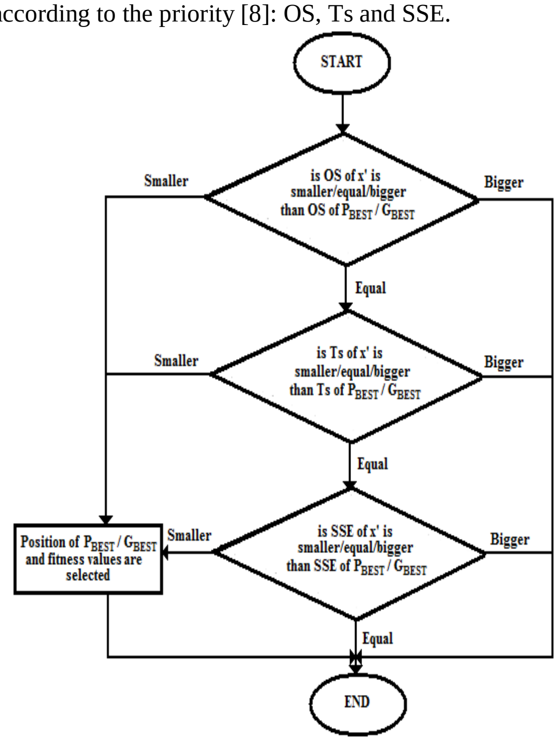


Fig. 2: Updated rules for $\boldsymbol{P_{BEST}}$ and $\boldsymbol{G_{BEST}}$ using a priority-based fitness approach

PSO could be implemented and applied easily to solve various function optimization problems especially for nonlinear models. For such problems, the particle position in PSO can be modelled as Equation 1.

$$X_i = [K1, K2] \tag{1}$$

K1 and K2 parameter values namely the break point and slope for the piecewise linear of SIFLC controller to control the position of the ROV, respectively. It is initialized and started with a number of random particles. Initialization of particles is performed using Equation 2.

$$X_i = x_{min} + \text{rand}\,(x_{max} - x_{min}) \tag{2}$$

Where x_{max} and x_{min} are the maximum and minimum values in the search space boundary. Then, the particles find for the local best, P_{BEST} and subsequently global best, G_{BEST} for every iteration in order to investigate for optimal result. Each particle is assessed by the fitness function. Thus, all particles try to imitate their historical success and in the same time try to follow the success of the best agent. It means that the P_{BEST} and G_{BEST} are updated if the particle has a minimum fitness value compared to the current P_{BEST} and G_{BEST} value. Nevertheless, only particles that are within the range of the system's constraint are accepted. The new velocity and new position can be calculated and tabulated as in Equation 3 and 4.

$$v_{i+1} = \omega v_i + c_1 r_1 (P_{BEST} - x_i) + c_2 r_2 (G_{BEST} - x_i) \tag{3}$$

$$x_{i+1} = x_i + v_{i+1} \tag{4}$$

Where r_1 and r_2 represent random function values [0,1] while c_1 is cognitive component and c_2 is social component, respectively. The function of ω parameter is to balance between local and global search capabilities by [15-16]. The PSO algorithm is used to tune and find two optimal parameters of SIFLC. Figure 3 shows a flowchart of the PSO algorithm for tuning of SIFLC parameters. In this study, 20 particles are considered with 100 iterations. The initial particles are bounded around 0 to 200. As default values, c_1 and c_2 are set as 2. The initial value of ω is 0.9 and linearly decreased to 0.4 at some stage in the iteration.

III. Priority –Based Fitness Binary PSO (PFBPSO)

Priority-based Fitness Binary PSO (PFBPSO) or BPSO has been introduced to solve discrete optimization problem in [17]. Applications of BPSO can be seen in many engineering problems, such as routing in VLSI, computational biology, job scheduling and agriculture [18-19]. In this research, a new method of Priority-based Fitness Binary Particle Swarm Optimization (BPSO) is proposed for tuning of SIFLC parameters. As explained in PSO in the previous section, overshoot, OS is set as the highest priority, followed by settling time, Ts and steady state error, SSE. Figure 2 illustrated the BPSO and PSO process where the P_{BEST} and G_{BEST} are updated according to the priority. The particles find for the local best, P_{BEST} and subsequently global best, G_{BEST} for each iteration in order to search for an optimal solution. Each particle is assessed by the fitness function. Thus, all particles try to replicate their historical success and in the same time try to follow the success of the best agent. It means that the P_{BEST} and G_{BEST} are updated if the particle has a minimum fitness value compared to the current P_{BEST} and G_{BEST} value. Nevertheless, only particles that within the range of the system's constraint are accepted. The new velocity can be calculated and as in equation (3).

Next, new particles are updated using equation (5) based on the sigmoid concept which is the probability of the normal distribution. All the parameters are obtained based on binary numbers (either 0 or 1) and then converted into decimal number that represents K1 and K2.

$$Sigmoid = \begin{cases} 1, rand < \frac{1}{1+e^{-V}} \\ 0, rand \geq \frac{1}{1+e^{-V}} \end{cases} \tag{5}$$

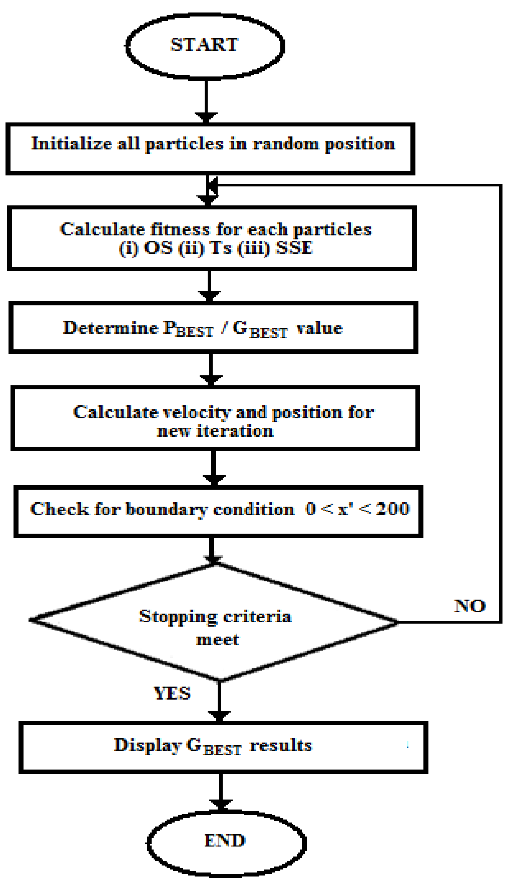


Fig. 3: Implementation of PSO and BPSO to tune SIFLC parameters

IV. Results and Discussion

Table 1 the results between two algorithms for PSO for single Input Fuzzy Logic Controller parameter. The results obtained from process of tuning for 20 times. Based on Table 1, BPSO will more consistent results compared PSO. The range of optimum parameter is reduced in size to minimize range compared with PSO. The data tabulated in graph as shown in Figure 4 and 5. It looks BPSO more consistent results obtained parameter in tune SIFLC. The range of parameter obtained from 20 to 62 for BPSO while PSO from 2 to 200 for K1. For K2 the range for BPSO from 0 to 55 while for PSO from 16 to 200. It seems the PSO algorithms totally random and almost the same weight range of setting parameter. BPSO obtained parameter more convenience to a certain range.

TABLE 1
COMPARISON BETWEEN BPSO AND PSO FOR K1

Test	BPSO K1	PSO K1	BPSO K2	PSO K2
1	61.796875	177.4653	55.375	24.1349
2	36.90625	199.9355	0.25	64.5831
3	30.375	87.8562	7.84375	29.8352
4	41.15625	147.3219	37.453125	199.7984
5	33.171875	2.6375	31.5	79.5406
6	34.53125	118.7462	48.453125	111.9401
7	41.328125	33.3279	22.453125	36.0662
8	22.921875	197.6138	17.75	148.1167
9	38.828125	117.1429	23.984375	91.4263
10	23.5	58.1121	25.359375	16.9369
11	48.125	5.3590	15.109375	145.2503
12	23.8125	105.7983	56	77.6201
13	46.734375	180.2101	34.359375	54.9053
14	21.4375	2.0845	28.375	182.5259
15	51.46875	65.6253	42.140625	86.4944
16	51.96875	26.3654	4.125	75.1863
17	24.03125	106.0368	27.109375	187.6596
18	54.375	157.6878	34.5	18.3357
19	21.609375	133.3886	21.203125	95.0725
20	49.125	117.3356	25.90625	16.5637

Figure 6 and Figure 7 shows the linear equation plotted for every graph for K1 and K2 using the optimum value of BSPO and PSO, respectively. Based on linear equation obtained, only intersection in y-axis considered as an optimum parameter of K1 and K2. It looks like the average value of tabulated data for K1 and K2. Table 2 shows the optimum parameter using a linear equation and the average value. Then each value for K1 and K2 for BPSO and PSO will be tested in simulation for Output Feedback Observer Tuning using Single Input Fuzzy Logic Controller as shown in Figure 1.

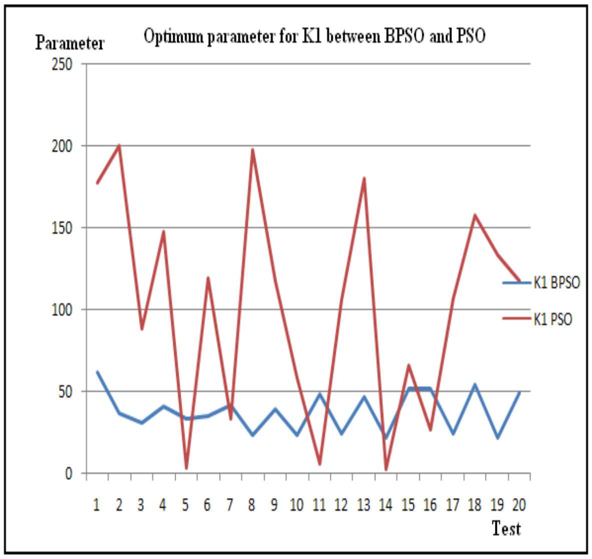


Fig. 4: Optimum parameter for K1 between BPSO and PSO

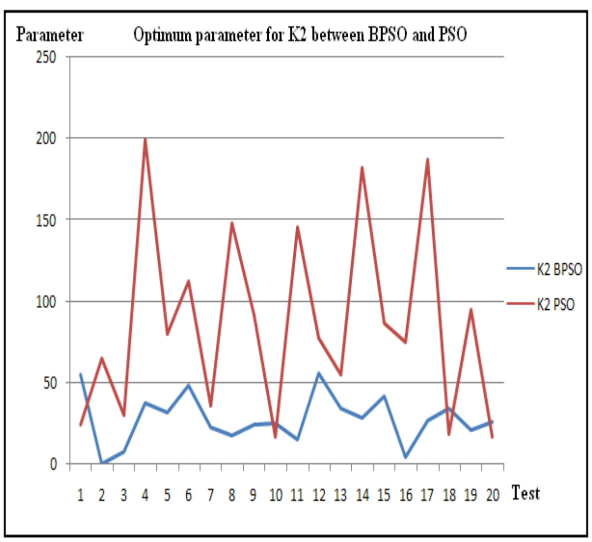
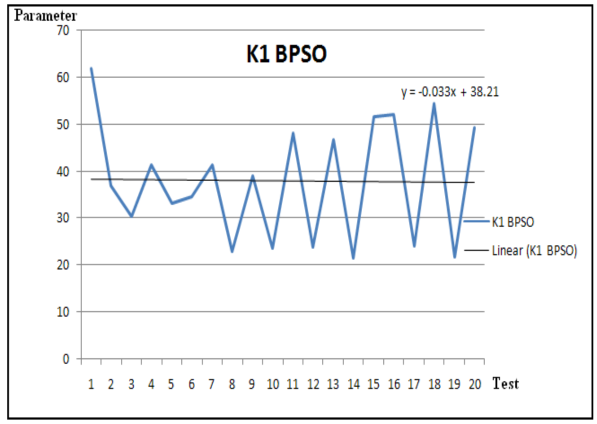
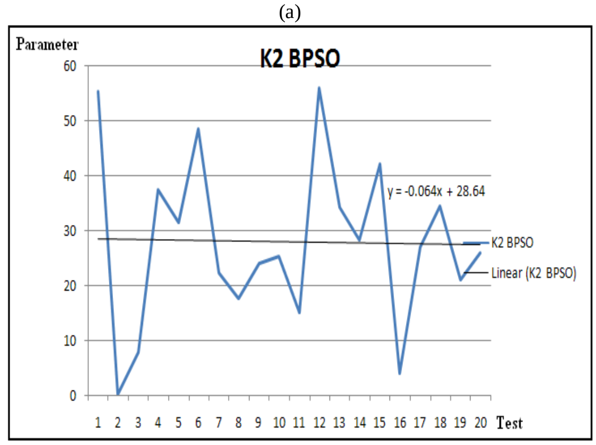


Fig. 5: Optimum parameter for K2 between BPSO and PSO

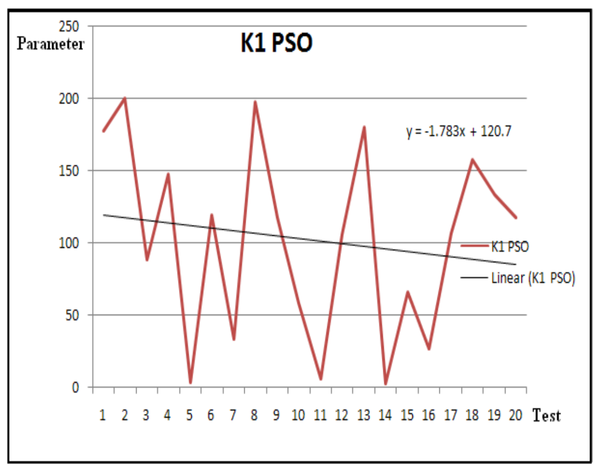


(a)

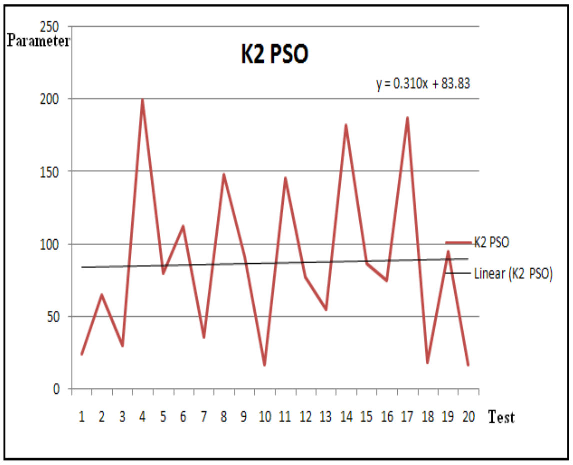


(b)

Fig. 6: (a) K1 for BSPO (b) K2 BPSO



(a)



(b)
Fig. 7: (a) K1 for PSO (b) K2 for PSO

TABLE 2
OPTIMUM PARAMETER USING A LINEAR EQUATION AND AVERAGE.

	K1 PSO	K1 BPSO	K2 PSO	K2 BPSO
Optimum Parameter	120.7	38.21	83.83	28.64
Average	120	37.86	87.1	27.96

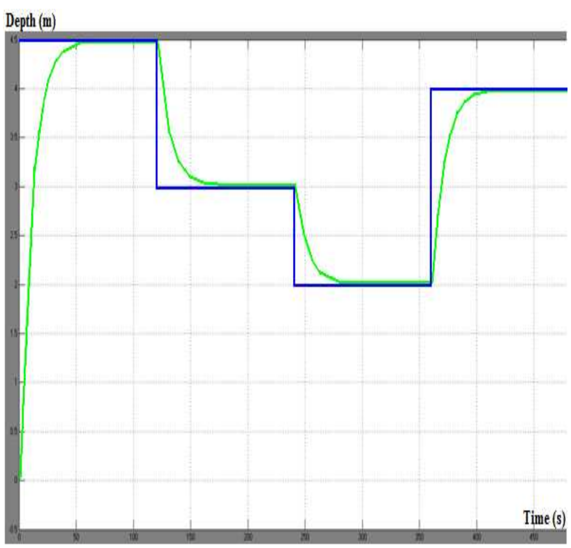


Fig. 8: System response for the average value of optimum value tuning by PSO

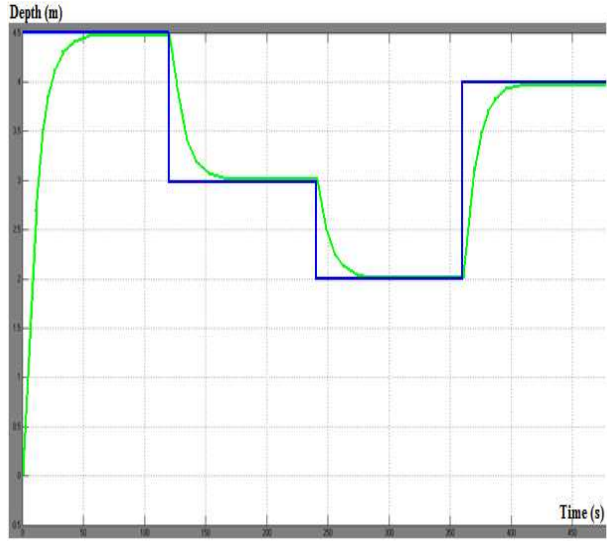


Fig. 9: System response for intersection value of optimum value tuning by PSO

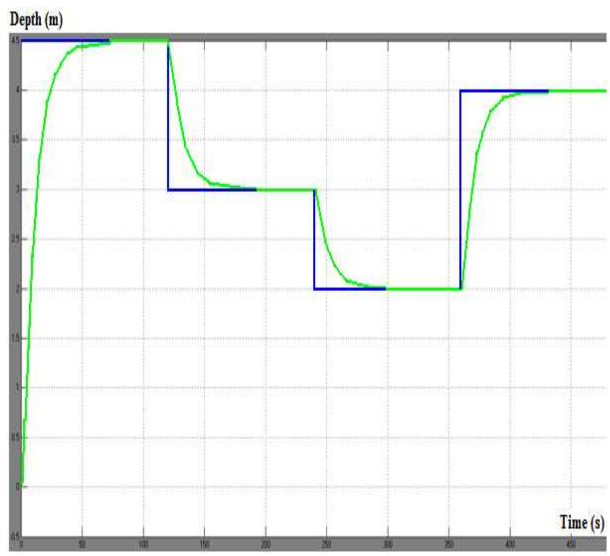


Fig. 10: System response for the average value of optimum value tuning by BPSO

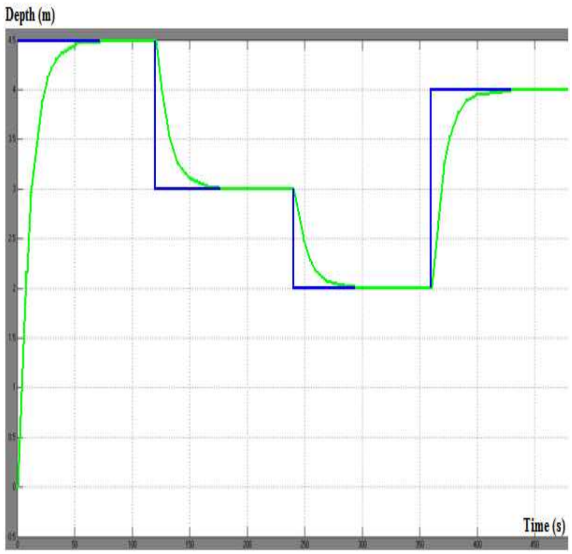
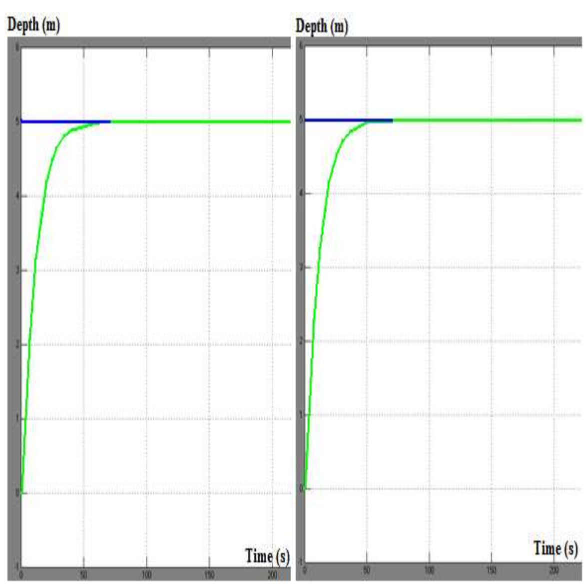


Fig. 11: System response for intersection value of optimum value tuning by BPSO

Figure 8 and Figure 9 shows the system response for intersection value and average value tuning using PSO. Figure 10 and Figure 11 shows the system response for intersection value and average value tuning using BPSO. The BPSO gives the best response in term of overshoot, rise time and steady state error. The system response of optimum value tuning by BPSO for average value intersection value as shown in Figure 12 almost same. Conclude that BPSO gives the best results of system response performances and optimum value can obtained from both techniques either using a linear equation or average value.



(a) (b)

Fig. 12: System response of optimum value tuning by BPSO for (a) average value (b) intersection value.

Hypothesis testing to see any significant difference between BPSO and PSO for K1 and K2 parameter. All data used in hypothesis testing can be seen in Appendix 1.

For KI:

The test: $H_0 : \mu_{BPSO} = \mu_{PSO}$
$H_1 : \mu_{BPSO} \neq \mu_{PSO}$

Two-sample T for BPSO K1 vs PSO K1

	N	Mean	StDev	SE Mean
BPSO K1	20	37.9	12.6	2.8
PSO K1	20	102.0	65.1	15

Difference = mu (BPSO K1) - mu (PSO K1)
Estimate for difference: -64.1424
95% CI for difference: (-95.0728, -33.2120)
T-Test of difference = 0 (vs not =): T-Value = -4.33 P-Value = 0.000 DF = 20

The value of the test statistics: $t = -4.33$
p-value for the test: $p = 0.000$

Therefore, there exists a significant difference between BPSO and PSO technique for K1.

For K2:

The test: $H_0 : \mu_{BPSO} = \mu_{PSO}$
$H_1 : \mu_{BPSO} \neq \mu_{PSO}$

Two-sample T for BPSO K2 vs PSO K2

	N	Mean	StDev	SE Mean
BPSO K2	20	28.0	15.3	3.4
PSO K2	20	87.1	59.0	13

Difference = mu (BPSO K2) - mu (PSO K2)
Estimate for difference: -59.1371
95% CI for difference: (-87.4611, -30.8131)
T-Test of difference = 0 (vs not =): T-Value = -4.34 P-Value = 0.000 DF = 21

The value of the test statistics: $t = -4.34$
p-value for the test: $p = 0.000$

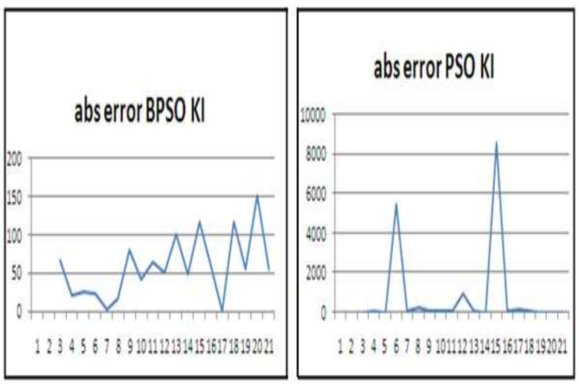
Therefore, there exists a significant difference between BPSO and PSO technique for K2.

Since there exists a significant difference between PSO and BPSO, so, we can decide which one is a better technique for obtaining optimum value for SIFLC.

i. Based from the value of variance
For KI, the value of standard deviation for BPSO is smaller than the variance for PSO.
For K2, the value of standard deviation for BPSO is smaller than the variance for PSO.
→ BPSO technique gives better optimization value than PSO for both K1 and K2.

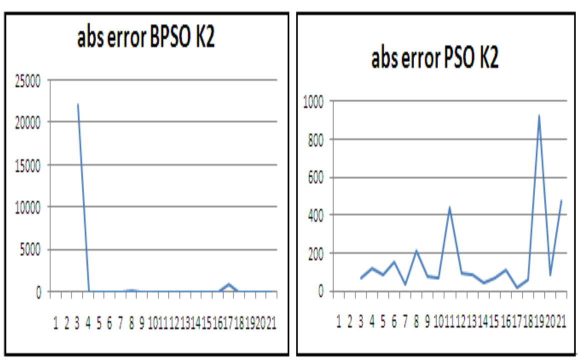
ii. Based from the value of error calculated.
For KI, the absolute value of the average error and the standard deviation are smaller for BPSO technique compare to PSO. For K2, the absolute value of the average error and the standard deviation are smaller for PSO technique compare to BPSO.

But referring to graphs of absolute error as shown in Figure 13 and Figure 14, the graphs exhibit a random pattern for BPSO K1, PSO K1, and PSO K2. Only the absolute error of BPSO K2 reduce with the number of iterations. From these graphs, it is better to increase the number iterations for both PSO and BPSO for each K1 and K2 until the error is as small as possible and the optimum value converges.



(a) (b)

Fig. 13: Graph exhibit random pattern for absolute error (a) K1 BPSO (b) K1 PSO



(a) (b)

Fig.14: Graph exhibit random pattern for absolute error (a) K2 BPSO (b) K2 PSO

TABLE 3
TIME EXECUTION FOR EVERY TESTING FOR PSO AND BPSO.

Test	Time Execution BPSO (s)	Time Execution PSO (s)
1	209.133031	971.929451
2	212.898391	954.122003
3	210.532833	952.974722
4	211.300865	953.310943
5	211.454390	952.983798
6	213.168615	949.453573
7	211.354633	952.731717
8	211.326776	959.364054
9	211.789711	952.090653
10	210.985970	950.623896
11	211.457587	948.845327
12	212.217930	949.341673
13	211.708684	947.177484
14	210.062159	951.637467
15	211.155030	959.819544
16	211.630295	956.206627
17	213.780470	955.06636
18	212.512246	958.722276
19	211.749119	958.027874
20	213.493897	964.185327

Others advantage of BPSO is time execution as shown in Table 3. It shows the BPSO more faster in time execution compared with PSO. It takes 3 minutes 30 seconds while PSO need 16 minutes for tuning the system to obtain optimum parameter. If we need to do a cycle of 20 iterations like tabulated in Table 3 it take 6 hours to complete the data compared BPSO need only 1 hour.

V. Conclusion

Two algorithms based on improving particle swarm optimization (PSO) algorithm using a priority-based fitness PSO (PFPSO) and binary priority-based fitness PSO (BPFPSO) approach to obtain optimum parameter in Single Input Fuzzy Logic Controller for underwater Remotely Operated Vehicle for depth control are successful. The study also covered a comparison for time execution for every time the parameter tuning was done. The BPFPSO gives comparative results in term of two parameters and time execution very fast compared with improved PSO. Also BPSO gives the best results of system response performances and optimum value can obtain from both techniques either using a linear equation or average value. Based from the value of variance for KI and K2, the value of standard deviation for BPSO is smaller than the variance for PSO. BPSO technique gives better optimization value than PSO for both K1 and K2.

Acknowledgement

Special appreciation and gratitude to the University (Universiti Teknikal Malaysia Melaka, UTeM and Universiti Teknologi Malaysia, UTM) especially to the both Faculties of Electrical Engineering for providing the financial as well as moral support to complete this project successfully.

References

[1] Kennedy, J. and Eberhart, R. “Particle Swarm Optimization”, Proceedings of the 1995 IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.

[2] J. Kennedy and R. Eberhart, A discrete binary version of the particle swarm algorithm, Proc. Of IEEE International Conference on Systems, Man, and Cybernetics, pp.4104-4108, 1997.

[3] 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 (ISMA08), Amman, Jordan, May 27-29, pp. 1-5, 2008.

[4] Clerc, M.; Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 6, 1, (2002) 58-73.

[5] Oko Pitono, Adi Soeprijanto, Mauridhi Hery Purnomo, Indar Chaerah Gunadin Power Generation Optimization Based on Steady State Stability Limit Using Particle Swarm Optimization (PSO),International Review on Modelling and Simulations (IREMOS), Vol. 6 N. 4, 2013.

[6] Tae-Hyoung Kim, Ichiro Maruta and Toshiharu Sugie, 2007 “Robust PID Controller Tuning Based on Constrained Particle Swarm Optimization”, Automatica, 44(4), pp. 1104-1110, 2008

[7] Hazriq Izzuan Jaafar, Z. Mohamed, Amar Faiz Zainal Abidin, Z. Ab Ghani, PSO-Tuned PID Controller for a Nonlinear Gantry, Crane System, IEEE International Conference on Control System, Computing and Engineering, 23 - 25 Nov. 2012, pp 1-5.

[8] Hazriq Izzuan Jaafar, Nursabillilah Mohd Ali, Z. Mohamed, Nur Asmiza Selamat, Anuar Mohamed Kassim, Amar Faiz Zainal Abidin, J.J. Jamian, Optimal Performance of a Nonlinear Gantry Crane System via Priority-based Fitness Scheme in Binary PSO Algorithm,pp 1 -6, 2013.

[9] Sonia Alimi, Mohamed Chtourou, Stability Analysis of Fuzzy Dynamic Model Identification, International Review on Modelling and Simulations (IREMOS), Vol. 5. n. 1, pp. 506-516, 2011.

[10] M.S.M Aras, S.S. Abdullah, H.I. Jaafar, A. A Rahman, M.A.A Aziz, Single Input Fuzzy Logic Controller tuning using PSO based on Simple Feed Forward and Output Feedback Observer for Underwater Remotely Operated Vehicle, Submitted to related journal (under review), 2013.

[11] F.A. Azis, M.S.M. Aras, S.S. Abdullah, Rashid, M.Z.A, M.N. Othman, Problem Identification for Underwater Remotely Operated Vehicle (ROV): A Case Study, Procedia Engineering; Volume 41, pp: 554-560, 2012.

[12] Mohd Shahrieel Mohd Aras, Shahrum Shah Abdullah, Azhan Ab Rahman, Muhammad Azhar Abd Aziz, Thruster Modelling for Underwater Vehicle Using System Identification Method, International Journal of Advanced Robotic Systems, Vol. 10, pp 1 – 12, 2013.

[13] M. S. M. Aras, F.A.Azis, M.N.Othman, S.S.Abdullah. A Low Cost 4 DOF Remotely Operated Underwater Vehicle Integrated With IMU and Pressure Sensor. In: 4th International Conference on Underwater System Technology: Theory and Applications 2012 (USYS'12), 2012 Malaysia, pp 18-23.

[14] Aras, M.S.M, S.S. Abdullah , Rashid, M.Z.A, Rahman, A. Ab, Aziz, M.A.A, Development and Modeling of underwater Remotely Operated Vehicle using System Identification for depth control,Jatit, 2013.

[15] E. J. Solteiro Pires1, J. A. Tenreiro Machado2 and P. B. de Moura Oliveira1, Particle Swarm Optimization: Dynamical Analysis through Fractional Calculus, Chapter 24, InTech Publisher, 2009.

[16] J. Kennedy and R. Eberhart, A discrete binary version of the particle swarm algorithm, Proc. of IEEE International Conference on Systems, Man, and Cybernetics, pp.4104-4108, 1997.

[17] A.M Kassim, T. Yasuno, N. Abas, M.S.M Aras, M. Z. A. Rashid, “Performance study of Reference Height Control Algorithm for Tripod Hopping Robot”, International Review of Mechanical Engineering, Vol. 7 N. 4, pp. 347 – 355, 2013.

[18] Sonia Alimi, Mohamed Chtourou, Stability Analysis of Fuzzy Dynamic Model Identification, International Review on Modelling and Simulations (IREMOS), Vol. 5. n. 1, pp. 506-516, 2011.

[19] Leonimer Flávio de Melo, Silas Franco dos Reis Alves, João Maurício Rosário, Mobile Robot Navigation Modelling, Control and Applications, International Review on Modelling and Simulations (IREMOS) Vol. 5. n. 2, pp. 1059-1068, 2011.

Authors’ Information

Mohd Shahrieel b Mohd Aras is a lecturer at Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka UTeM. He currently pursues his PhD in Control and Automation, Faculty of Electrical Engineering, Universiti Teknology Malaysia. His current research is focusing on control system design of underwater technology. His primary interests related to underwater robotics and Artificial Intelligence.

Dr. Shahrum Shah b Abdullah is a senior lecturer at Faculty of Electrical Engineering, Universiti Teknologi Malaysia.14 Years of experience in this university .He got Ph.D in Artificial Neural Networks, Department of Electrical and Electronic Engineering, Faculty of Engineering, Imperial College of Science, Technology and Medicine, University of London. Thesis: Experiment Design for Deterministic Model Reduction and Neural Network Training. Currently he is Deputy Dean, Malaysia – Japan International Institute of Technology (MJIIT). website http://ac.utm.my/web / shahrum

Hazriq Izzuan Jaafar received the B.Eng degree in Electrical Engineering from Universiti Teknologi Malaysia (UTM), in 2008. He was an Assistant Manager in Telekom Malaysia Berhad, from 2008 to 2010. He received the M.Eng degree in Mechatronics and Automatic Control engineering also from UTM, in 2013. Currently, he is a Lecturer at Universiti Teknikal Malaysia

Melaka (UTeM) and his interests are in control system and optimization techniques.


Razilah Abdul Rahim is a lecturer at Universiti Teknikal Malaysia Melaka. She obtained her B.A. in English Language Teaching from the University of Surrey, UK and her M.A. in English Language Studies from Universiti Kebangsaan Malaysia. She has more than 15 years of experience teaching proficiency and specialized courses for general and English for Specific Purposes (ESP). Her research interests include technology-enhanced language learning, ESP and cross-cultural communication.

.


Arfah Ahmad is a lecturer at Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka UTeM. She received the Master Degree (Statistics), and Bachelor Degree (Statistics),from UKM. Her interest in Statistical Analysis, Modelling, Survey & Sampling, Mathematics.

DESCRIPTIVE STATISTICS AND ERROR CALCULATION
STATISTIC

Appendix

TABLE 4

Test	BPSO K1	error BPSO KI	PSO K1	error PSO KI	BPSO K2	error BPSO K2	PSO K2	error PSO K2
1	61.797		177.4653		55.375		24.135	
2	36.906	-67.443	199.9355	11.239	0.250	-22050.000	64.583	62.630
3	30.375	-21.502	87.8562	-127.571	7.844	96.813	29.835	-116.466
4	41.156	26.196	147.3219	40.364	37.453	79.057	199.798	85.067
5	33.172	-24.070	2.6375	-5485.664	31.500	-18.899	79.541	-151.190
6	34.531	3.937	118.7462	97.779	48.453	34.989	111.940	28.944
7	41.328	16.446	33.3279	-256.297	22.453	-115.797	36.066	-210.374
8	22.922	-80.300	197.6138	83.135	17.750	-26.496	148.117	75.650
9	38.828	40.966	117.1429	-68.695	23.984	25.993	91.426	-62.007
10	23.500	-65.226	58.1121	-101.581	25.359	5.422	16.937	-439.805
11	48.125	51.169	5.359	-984.383	15.109	-67.839	145.250	88.340
12	23.813	-102.100	105.7983	94.935	56.000	73.019	77.620	-87.130
13	46.734	49.047	180.2101	41.292	34.359	-62.983	54.905	-41.371
14	21.438	-118.003	2.0845	-8545.243	28.375	-21.090	182.526	69.919
15	51.469	58.349	65.6253	96.824	42.141	32.666	86.494	-111.026
16	51.969	0.962	26.3654	-148.907	4.125	-921.591	75.186	-15.040
17	24.031	-116.255	106.0368	75.136	27.109	84.784	187.660	59.935
18	54.375	55.805	157.6878	32.755	34.500	21.422	18.336	-923.466
19	21.609	-151.627	133.3886	-18.217	21.203	-62.712	95.073	80.714
20	49.125	56.011	117.3356	-13.681	25.906	18.154	16.564	-473.981
mean	37.860	-20.402	102.003	-798.778	27.963	-1203.952	87.100	-109.508
variance	158.238	4809.461	4239.081	5118031.22	234.680	25531703.62	3475.314	66107.031
std dev	12.579	69.350	65.108	2262.307	15.319	5052.891	58.952	257.113
median	37.867	0.962	111.590	-13.681	26.508	5.422	78.580	-41.371
max	61.797	58.349	199.936	97.779	56.000	96.813	199.798	88.340
min	21.438	-151.627	2.085	-8545.243	0.250	-22050.000	16.564	-923.466