Intelligent Active Force Control of an Underwater Remotely Operated Vehicle Using Evolutionary Computation Technique

See Zu Yuan*, Musa Mailah and Tang Howe Hing

School of Mechanical Engineering, Faculty of Engineering
Universiti Teknologi Malaysia
81310 UTM Johor Bahru
Johor, Malaysia

Received: 15 June 2020; Revised: 11 September 2020; Accepted: 3 November 2020; Published: 15 December 2020

ABSTRACT

The purpose of this study is to develop a robust control system on an underwater remotely operated vehicles (ROV). An intelligent active force control (AFC) with evolutionary computation (EC) algorithms was implemented to a selected dynamic model of the UROV. System performance employing different hybrid control schemes that applied AFC and EC was observed and studied to evaluate review their performance and capability. The system performance of the six degree of freedom (6-DOF) UROV related to the surge, sway, heave, roll, pitch and yaw motions, respectively were observed to obtain the responses related to the steady-state error, overshoot and settling time as the criteria for evaluation. Conventional PID controller was first implemented on the system and tuned using a heuristic method before applying any disturbances and other control techniques. A comparative study between PID, PID-AFC and PID-AFC-EC was then conducted to determine the best control technique amongst them. The results showed that PID-AFC-EC is robust, has low steady state error and fast settling time even when disturbances were presence in and applied to the system.

Keywords: *Underwater remotely operated vehicle (UROV), proportional-integral-derivative (PID), active force control (AFC), evolutionary computation (EC), robustness*

1.0 INTRODUCTION

Underwater remotely operated vehicles (ROVs) play an important role in a number of shallow and deep-water missions for marine science, oil and gas extraction, exploration and salvage. In these applications, the motions of the ROV are guided either by a human pilot on a surface support vessel through an umbilical cord providing power and telemetry, or by an automatic pilot. These systems are crucial because of the varying operating range and depth posed by underwater areas. Due to unknown non-linear hydrodynamic effects, parameter uncertainties and lack of a precise model of the ROV dynamics and parameters, developing a control system of the ROV can be challenging given the complexity [1]. Therefore, to have a good performance, a robust control technique is one of the important elements in ROV [2]. The proportional-integral-derivative (PID) controller was first proposed to the ROV. PID controller provides preliminary satisfaction by eliminating the errors, increased the rise time and reduced the settling time with the low-performance requirement. However, the PID compensation is unsuccessful to reject the disturbances when the performance requirement of the ROV is increased.

*Corresponding email: zysee2@graduate.utm.my

The researchers then introduced different types of artificial intelligent controllers to enhance the robustness of the system. The SIFLC (Single Input FLC) has been applied to control the depth of ROV by Ishaque *et al.* (2010) [3]. The simulation reveals that the SIFLC has excellent performance and it exactly resembles conventional FLC in terms of its response. Xu and Wang (2018) proposed a control scheme that apply Genetic Algorithm (GA) to a fuzzy adaptive PID controller, where the fuzzy control rule of a fuzzy PID controller is optimized using GA, and results show improvements in performances as compared to ordinary fuzzy-PID controller [4]. Sliding mode control (SMC) has also been successfully employed in depth control for underwater robots owing to its insensitivity to model imprecision, parametric uncertainty, and external disturbances [5, 6]. In this control scheme, the dynamics of the system are altered by the application of high-speed switching control. However, SMC has the problem of chattering effect caused by its switching inputs, and it is difficult to transform the dynamical model into a standard form. An adaptive fuzzy sliding mode controller was also implemented by Bessa *et al.* (2010) to deal with the stabilization and trajectory tracking problems [7]. Also, an adaptive neuro-fuzzy sliding mode based genetic algorithm control system was designed in order to achieve a high-accuracy position control for a remotely operated vehicle [8]. On this basis, researchers have proposed different adaptive fuzzy SMC scheme to better improve the control accuracy A robust adaptive self-organizing neuro-fuzzy control (RASNFC) scheme for trajectory tracking of ROV is proposed in [9] but the neural network control learning speed is slow, easy to fall into local minimum, network layer number is limited and overfitting.

There are many control techniques that have been successfully applied by other researchers to the UROV for the past few years. However, the intelligent AFC scheme using evolutionary computation (EC) has not been implemented to the UROV control system. AFC is more reliable and practically applicable in real-time due to the lesser computational burden incurred for signal processing. With the aid of EC technique, the inertial parameter in the AFC loop can be well approximated.

2.0 THEORETICAL PRELIMINARY

2.1 Active Force Control (AFC)

The research on active force control (AFC) is initiated based on the principle of invariance and the classic *Newton's* second law of motion. Mailah *et al*. (1996) extended the usefulness of the method by introducing intelligent mechanisms to approximate the mass or inertia matrix of the dynamic system to trigger the compensation effect of the controller, thereafter making the control system very robust and effective [10].

AFC-based is practically applicable in real-time due to the much lesser computational burden incurred for signal processing [11]. However, it is a well-known fact that the AFC scheme could not work or compensate alone without the classical control PID or other accompanying feedback controller in the outermost control loop as the supporting backbone of the scheme [12]. In this work, AFC was introduced as an internal positive feedback loop in the PID control system. The AFC acts as an adaptive mechanism to approximate the inertial parameter of the dynamic system to trigger the compensation effect of the PID controller. Hence, AFC is able to optimize the system responses as well as overcome the weaknesses of the basic PID controller. Note that one great advantage of using the AFC-based scheme is that once the PID control gains were satisfactorily and initially tuned for a specific loading/operating condition, the fixed gains need no further tuning regardless of any subsequent disturbance conditions once the AFC mechanism is in place. In other words, instead of having to tune the typical PID terms (K_p, K_i and K_d) for any parametric/non-parametric changes in loading/operating conditions or even uncertainties, one needs to tune only a single parameter in the AFC loop, i.e., the estimated inertial/mass parameter.

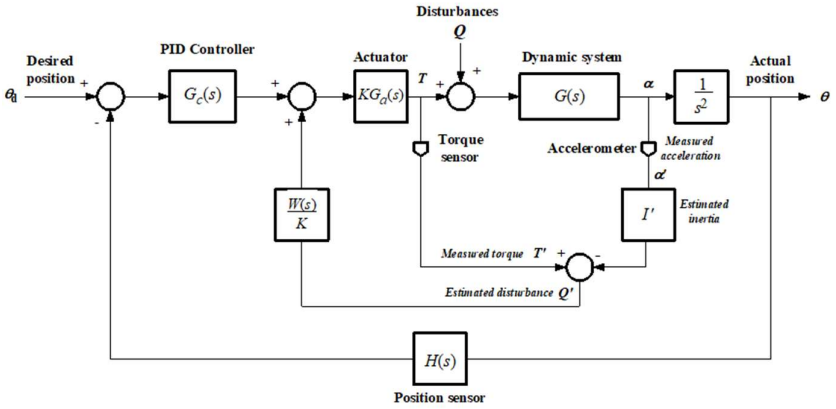

Figure 1: AFC scheme

The estimated disturbance torque *Q'* can be obtained as follows:

$$Q' = T' - I'\ddot{\theta}' \quad (1)$$

where

T'	:	measured torque applied to ROV system
I'	:	estimated mass moment of inertia
$\ddot{\theta}'$	:	measured angular acceleration

The AFC in Figure 1 estimates the inertial or mass parameters of the ROV and measure the acceleration and force signals induced by the system. Then, the estimated disturbance, *Q'* is computed and passed through a weighting function W(s) to give the ultimate AFC signal to be embedded with the outer control loop.

2.2 Evolutionary Computation (EC)

EC is a family of algorithms for global optimization inspired by biological evolution, and the subfield of artificial intelligence (AI) and soft computing studying these algorithms. In technical terms, they are a family of population-based trial and error problem solvers with a metaheuristic or stochastic optimization character [13].

The EC techniques mostly involve metaheuristic optimization algorithms such as ant colony optimization (ACO), particle swarm optimization (PSO), genetic algorithm (GA), evolutionary algorithm, evolutionary programming, etc. This study specifically focuses on the application of GA and PSO methods for optimization of the estimated inertial parameter, *I*' of the AFC

GA is devised based on the principle of survival of the fittest. The algorithm generates different sets of candidates or population, *P*(n) of designated size for each generation based on their fitness determined through the selection process, evaluated through fitness function. Selected individuals are parents that will be used to produce the next generation. Crossover process will combine chromosomes, depicted by the bits in data, of pairs of parents to produce children. With certain probability as designated, some of the offspring will then undergo mutation process which randomly alters the chromosome. The process will then loop until optimized solution is produced or the stopping criteria is met.

The general concept of the PSO technique is based on the social behavior of a bird flock and bee swarm. Each particle in the search space is driven by a velocity vector to search for the best position (optimal solution) [14]. Similar to GA, the evaluation for a local and global best location is determined by a fitness function. After acquiring the best location, the velocity vector of the particles will be updated, which then cause them to travel to the

selected location. The process will be iterated until the stopping criteria is satisfied, decided either by maximum iteration number or fitness tolerance.

The optimization of the estimated inertia parameter using GA and PSO, are carried out in the workspace by running the simulation by iteration. The method is based on the study of applying GA for tuning of PID gains as reported in [15, 16]. For each iteration, error output, *e*(t) from the simulation will be retrieved from the simulation to workspace. The error output as a six values array will then be processed through the algorithms to produce the updated values of estimated inertia, *I'* for the cycle. The implementation of EC to AFC is represented as block diagram in Figure 2. The GA optimization was carried out using *Optimization Tool App* in MATLAB, with the *Multiobjective Genetic Algorithm Solver* and the related graphical user interface (GUI) for the app is shown in Figure 3.

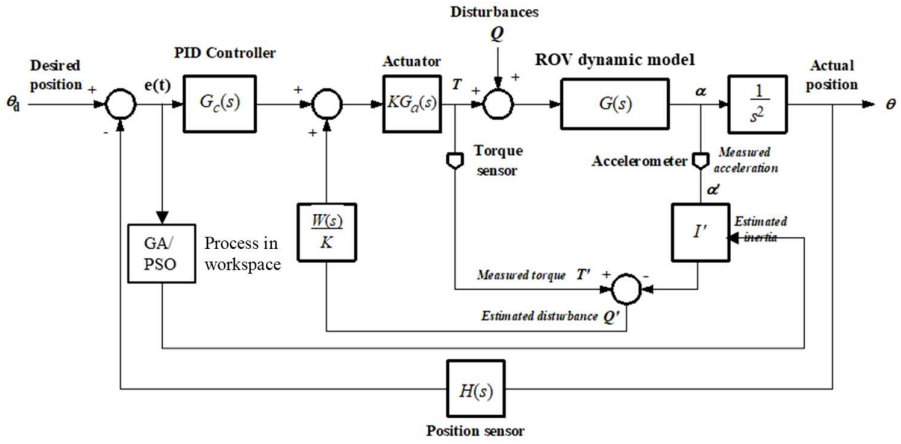

Figure 2: AFC scheme integrated with EC

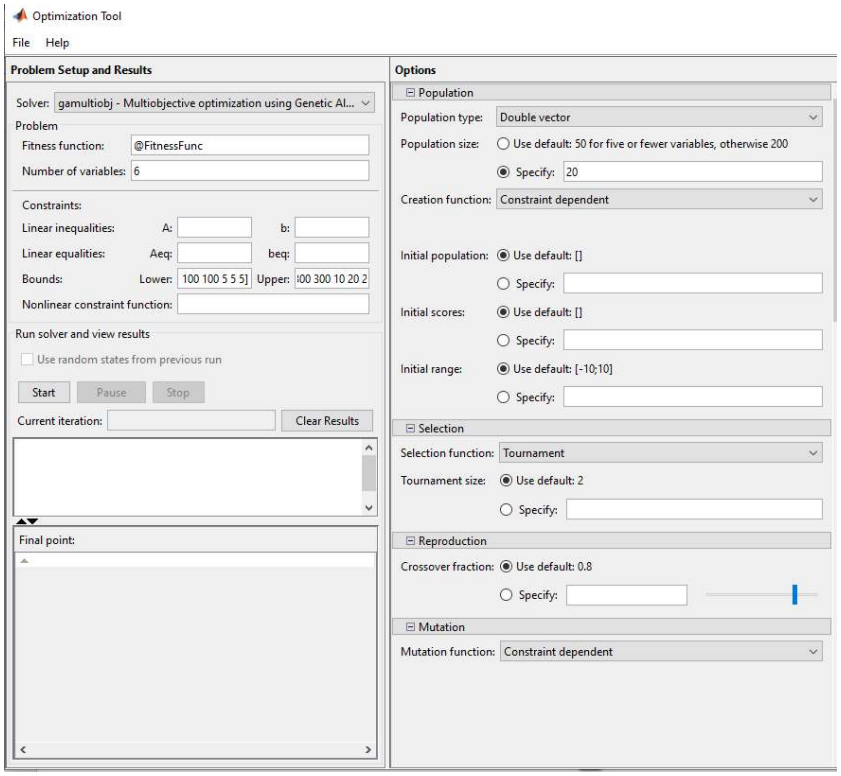

Figure 3: GUI of the *Optimization Tool App*

The PSO was then implemented by executing an algorithm in MATLAB script. The fitness or objective function used for the evaluation is the integral absolute error (IAE), which is the integral of the absolute value of the tracking errors, $e(t)$.

3.0 MODELING THE UROV

3.1 UROV Dynamic Model

The proposed UROV mathematical modeling was obtained based on works in [17] and this model was used as the system dynamic model throughout the simulation works. The UROV used in this study was the UROV designed by Robotics Research Centre (RRC) at *Nanyang Technological University* (NTU) to carry out the underwater pipeline inspection such as locating pipe leakages or cracks. The 6-DOF UROV were conventionally defined as shown in Figure 4.

Notations used in the figure:

$\boldsymbol{\eta}$: position and orientation in inertia frame
$\boldsymbol{\eta} = [\eta_1 \quad \eta_2]^{\mathrm{T}} = [x \quad y \quad z \quad | \quad \phi \quad \theta \quad \psi]^{\mathrm{T}}$

$\mathbf{v}$: linear and angular velocities in body-fixed frames
$\mathbf{v} = [\mathrm{v}_1 \quad v_2]^{\mathrm{T}} = [u \quad v \quad w \quad | \quad p \quad q \quad r]^{\mathrm{T}}$

$\boldsymbol{\tau}$: force and moments acting on the vehicle in body-fixed frames
$\boldsymbol{\tau} = [\tau_1 \quad \tau_2]^{\mathrm{T}} = [\tau_x \quad \tau_y \quad \tau_z \quad | \quad \tau_\phi \quad \tau_\theta \quad \tau_\psi]^{\mathrm{T}}$

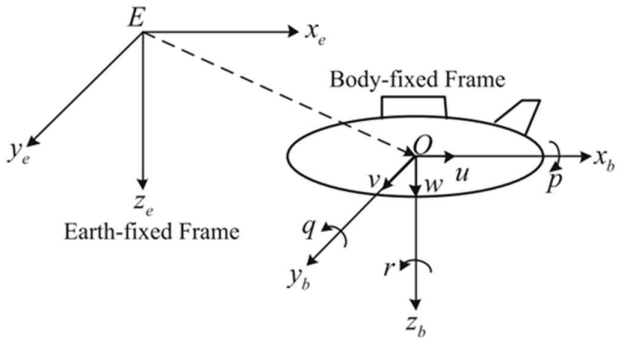

Figure 4: Notations of the UROV model [17]

The mathematical model of an underwater vehicle can be expressed with respect to a local body-fixed reference frame, by non-linear equations of motion in matrix form [18]:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{2}$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta}_2)\mathbf{v} \tag{3}$$

Where,

$\mathbf{M}$	:	mass inertia matrix for rigid body and added mass
$\mathbf{C}(\mathbf{v})$	:	*Coriolis* and centripetal matrix for rigid body and added mass
$\mathbf{D}(\mathbf{v})$	:	linear and quadratic damping matrix
$\mathbf{g}(\boldsymbol{\eta})$	:	gravitational and buoyancy vector
$\boldsymbol{\tau}$	:	input force and moment vector
$\mathbf{J}(\boldsymbol{\eta}_2)$	:	*Euler* transformation matrix

The detailed derivation of the model can be found in [17], the derived and simplified (based on assumptions made) matrices involved is as follows:

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & m - Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} - K_{\dot{p}} & -I_{xy} & -I_{xz} \\ 0 & 0 & 0 & -I_{yx} & I_{yy} - M_{\dot{q}} & -I_{yz} \\ 0 & 0 & 0 & -I_{zx} & -I_{zy} & I_{zz} - N_{\dot{r}} \end{bmatrix} \tag{4}$$

$$\mathbf{C}(\mathbf{v}) = \begin{bmatrix} 0_{3\times3} & C_{12}(v) \\ -C_{12}^{T}(v) & C_{22}(v) \end{bmatrix} \tag{5}$$

Where

$$C_{12}(v) = \begin{bmatrix} 0 & mw - Z_{\dot{w}}w & -mv + Y_{\dot{v}}v \\ -mw + Z_{\dot{w}}w & 0 & mu - X_{\dot{u}}u \\ mv - Y_{\dot{v}}v & -mu + X_{\dot{u}}u & 0 \end{bmatrix} \tag{6}$$

$$C_{22}(v) = \begin{bmatrix} 0 & -I_{yz}q - I_{xz}p + I_z r - N_{\dot{r}}r & I_{yz}r + I_{xy}p - I_y q + M_{\dot{q}}q \\ I_{yz}q + I_{xz}p - I_z r + N_{\dot{r}}r & 0 & -I_{xz}r - I_{xy}q + I_x p - K_{\dot{p}}p \\ -I_{yz}r - I_{xy}p + I_y q - M_{\dot{q}}q & I_{xz}r + I_{xy}q - I_x p + K_{\dot{p}}p & 0 \end{bmatrix} \tag{7}$$

$$\mathbf{D}(\mathbf{v}) = \begin{bmatrix} -X_u - X_{u|u|}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & -Y_v - Y_{v|v|}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & -Z_w - Z_{w|w|}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & -K_p - K_{p|p|}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & -M_q - M_{q|q|}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & -N_r - N_{r|r|}|r| \end{bmatrix} \tag{8}$$

$$\mathbf{g}(\boldsymbol{\eta}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ (z_G - z_B)W cos\theta sin\phi \\ (z_G - z_B)W sin\theta \\ 0 \end{bmatrix} \tag{9}$$

$$\mathbf{J}(\boldsymbol{\eta}_2) = \begin{bmatrix} \mathrm{J}_1(\eta_2) & 0_{3\times3} \\ 0_{3\times3} & \mathrm{J}_2(\eta_2) \end{bmatrix} \tag{10}$$

Where

$$\mathrm{J}_1(\eta_2) = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi) + c(\psi)c(\phi)s(\theta) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\phi)s(\theta)s(\psi) & -c(\psi)s(\phi) + s(\theta)s(\psi)c(\theta) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad (11)$$

$$\mathrm{J}_2(\eta_2) = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & s(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix} \quad (12)$$

Note: c = cos, s = sin, t = tan

The parameter values such as the coefficients were obtained through experimental and simulation works done in [19] as shown in Table 1.

Table 1: The UROV parameters [19]

Parameter	Value
m	113.5 kg
W	1110.5 N
$z_B - z_G$	0.048 m
I_{xx}	6.100 kgm^2
I_{yy}	5.980 kgm^2
I_{zz}	9.590 kgm^2
I_{xy}	-0.00016 kgm^2
I_{xz}	-0.185 kgm^2
I_{yz}	0.0006 kgm^2
X_u	-252.98 kgs^{-1}
Y_v	-1029.51 kgs^{-1}
Z_w	-1029.51 kgs^{-1}
K_p	-97.78 kgs^{-1}
M_q	-142.22 kgs^{-1}
N_r	-71.11 kgs^{-1}
$X_{u\|u\|}$	-423 Nm^2
$Y_{v\|v\|}$	-747 Nm^2
$Z_{w\|w\|}$	-735 N.m^2
$K_{p\|p\|}$	-99 Nm^2
$M_{q\|q\|}$	-126 Nm^2
$N_{r\|r\|}$	-62 Nm^2
$X_{\dot{u}}$	-0.6 kg
$Y_{\dot{v}}$	-107 kg
$Z_{\dot{w}}$	-107 kg
$K_{\dot{p}}$	-0.0023 kg
$M_{\dot{q}}$	-6.23 kg
$N_{\dot{r}}$	-6.23 kg

Where

m	:	rigid body mass of the UROV
W	:	gravitational force due to the UROV weight
$z_B - z_G$	:	distance between center of buoyancy and center of gravity

$I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}$: moment of inertias
$X_u, Y_v, Z_w, K_p, M_q, N_r$: linear damping coefficients
$X_{u|u|}, Y_{v|v|}, Z_{w|w|}, K_{p|p|}, M_{q|q|}, N_{r|r|}$: quadratic damping coefficients
$X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}$: added mass and moment of inertia coefficients

3.2 MATLAB/Simulink Computing Platform

The simulation of the UROV control system was carried out in MATLAB/Simulink. The Simulink block diagram of the control system is shown in Figure 5. The dynamic model of the UROV is marked by the orange dotted line and was constructed based on Equations (2) to (12) using the *ROV Design and Analysis Toolbox* [20]. The control schemes were then implemented into the model as proposed earlier and are represented by the PID (green) and AFC (Blue) schemes. The application of the EC components into AFC is not shown in the block diagram since the parameter optimization process for the estimated inertia was done in MATLAB workspace based on the codes developed for the EC algorithms. Note that the UROV system adopts a 6-DOF configuration related to motions to be controlled in order to achieve the desired output. A perfect modeling was also assumed for the actuator and sensor with the thruster configuration disregarded for the simulation. A saturation block was deliberately added in the schematics to limit the input to the actuator so that it does not operate beyond its maximum capability.

The simulation was first run without disturbances based on the basic PID controller so that the tuning of the PID gains can be carried out. Later, a pulsating type of disturbance was introduced to test the system effectiveness and robustness. The gains of the PID controller for all 6-DOF were tuned heuristically using a trial-and-error method (TEM). The optimized gain values and the desired outputs for each DOF are shown in Table 2 based on works done in [19]. The entire simulation was performed using the variable step integral solver *ODE45*. The PID controller shall serve as the basis for comparison with the other AFC-based schemes considered in the study.

The next stage involves the application of the AFC-based control scheme by simply adding it in series with the conventional PID controller. The EC methods based on GA and PSO algorithms were embedded into the AFC loop to estimate the inertial parameters of the UROV system for the triggering of the AFC control action. They were also similarly tested like the PID schemes for comparison purpose.

Table 2: UROV parameters [19]

Degree of Freedoms	Desired output	PID gains		
		K_p	K_i	K_d
Surge, x	1 m	35	0.001	55
Sway, y	1 m	60	0.0001	95
Heave, z	1 m	95	0.0001	90
Roll, Φ	0.5 rad	0.1	35	15
Pitch, θ	0.4 rad	15	35	15
Yaw, ψ	0.3 rad	30	8	12

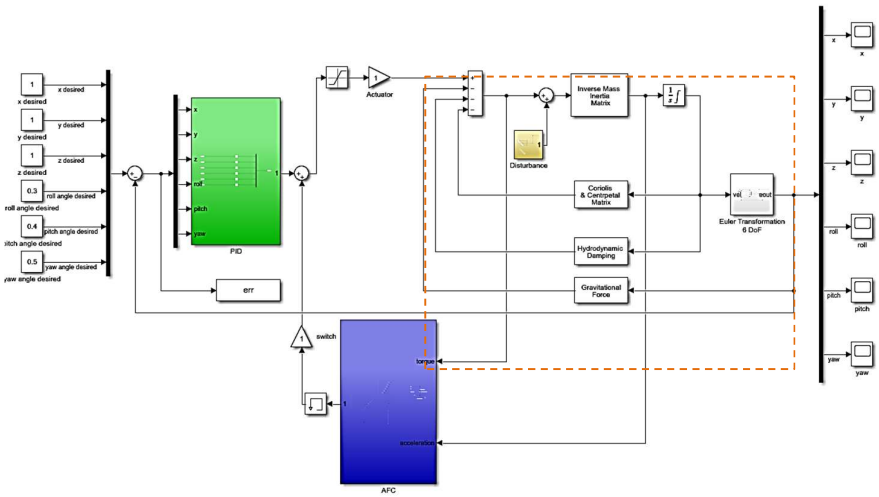

Figure 5: The UROV control scheme

4.0 RESULTS AND DISCUSSION

Figures 8 shows the results of the system response to the constant and step input references for the given PID gains. Without the presence of any disturbances, the PID controller performed satisfactorily with slight overshoot but fast settling time.

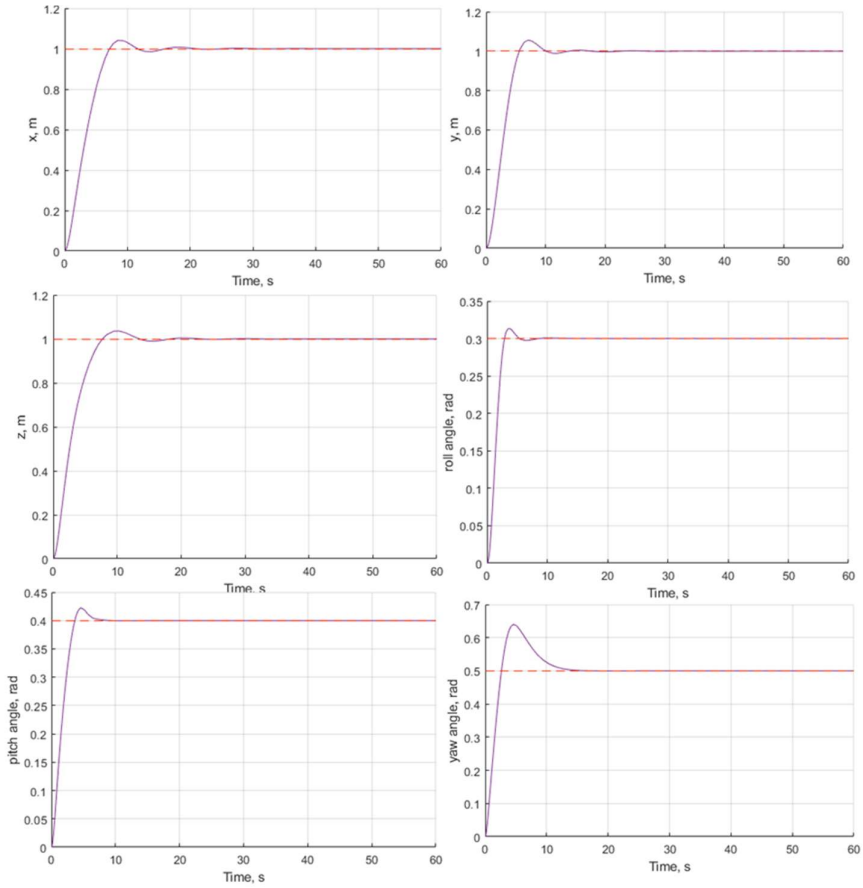

Figure 8: System response without disturbance for the PID controller

Figure 9 clearly shows the existence of a significant steady state error with a series of 'hills and valleys' pattern when a pulse disturbance was applied to the system for all 6-DOF motions. This condition is similar to a minor collision incident with an underwater object at a constant time interval. The result indicates that the system performance is considerably affected at the point of applied disturbance to the UROV system. This further implies that the conventional PID control method is not able to compensate for this adverse condition, thereby failing to achieve the desired output response.

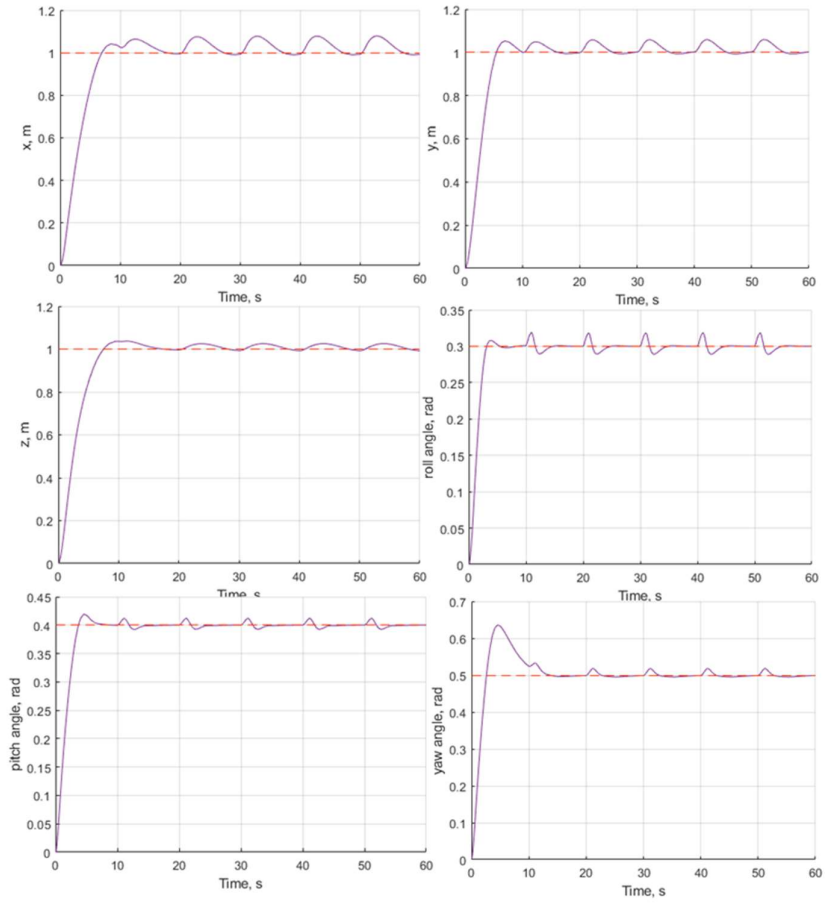

Figure 9: PID with pulse disturbance for all 6-DOF UROV

The AFC control scheme was later integrated with the conventional PID to form a two DOF control strategy, by cascading it with the PID control loop to form a PID-AFC scheme, noting that the PID controller gains (fixed) need no further tuning. The AFC was implemented to test for its ability for disturbance compensation. The AFC's estimated inertial parameters, *I*' of the respective DOF were crudely approximated based on the computed values in the inertia matrix of the UROV model.

The results shown in Figure 10 exhibits the robustness of AFC compared to the PID counterpart, implying that the effect of disturbances is greatly compensated. The steady state error is much reduced, and the response has low overshoot with fast settling time for some DOF. In general, the AFC produced better response under pulse disturbance compared to that of PID.

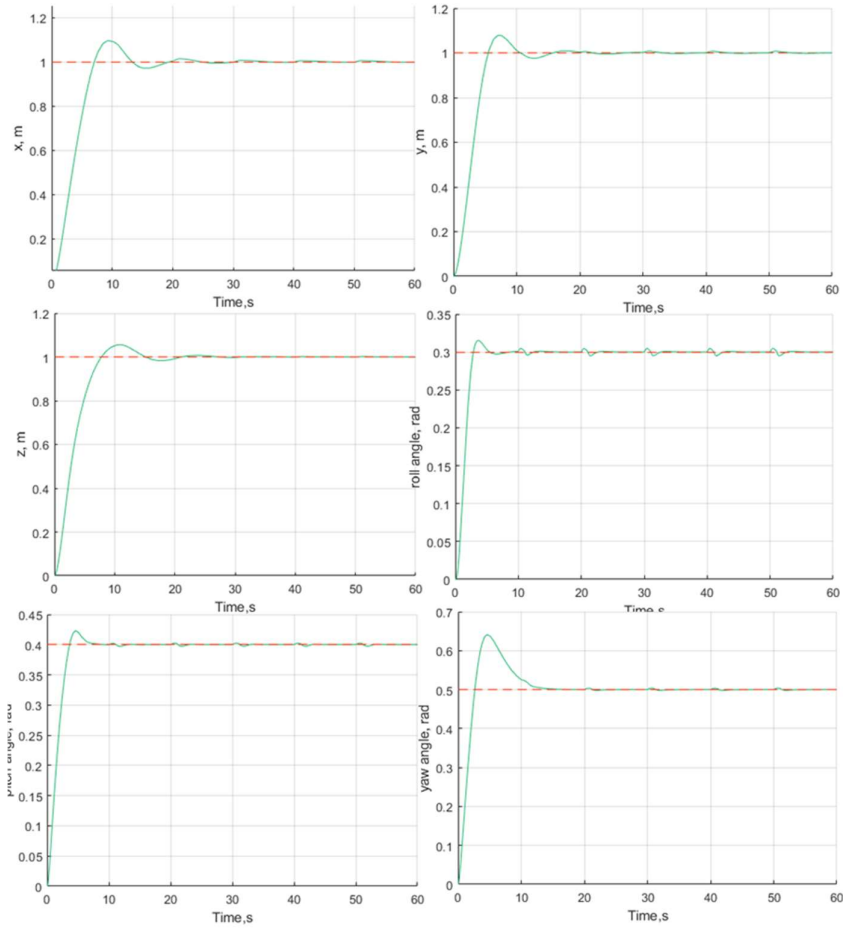

Figure 10: AFC with impulse disturbance for all 6-DOF UROV

In order to optimize and obtain better estimation of the value of inertial parameters, EC methods were then applied to the PID-AFC control scheme. The specific EC algorithms used for the simulation are GA and PSO. For GA, using the *Multiobjective Genetic Algorithm Solver*, the *gamultiobj* function in the *Optimization Tool Apps* was selected for the simulation, because optimization involves an array input of error for all 6-DOF. The parameters used for the problem setup and options are shown in Table 3.

Table 3: Parameters for the GA

Range of *I'*	$100 < I'_x < 200$	$100 < I'_y < 300$	$100 < I'_z < 300$
	$5 < I'_\Phi < 10$	$0 < I'_\Theta < 20$	$0 < I'_\psi < 20$
Number of generations	50		
Number of populations	20		
Fitness function	$f = \int \lvert e(t) \rvert \, dt$ where $e(t)$ is the error output from simulation		

The results generated using the final value of inertial parameter obtained from the GA optimization are shown in Figure 11. The effect of the disturbances can be seen to be adequately compensated and it resulted in producing a signal with relatively low steady state error. However, the response does not seem to perform better when compared to AFC, since it has higher overshoot and slightly longer settling time.

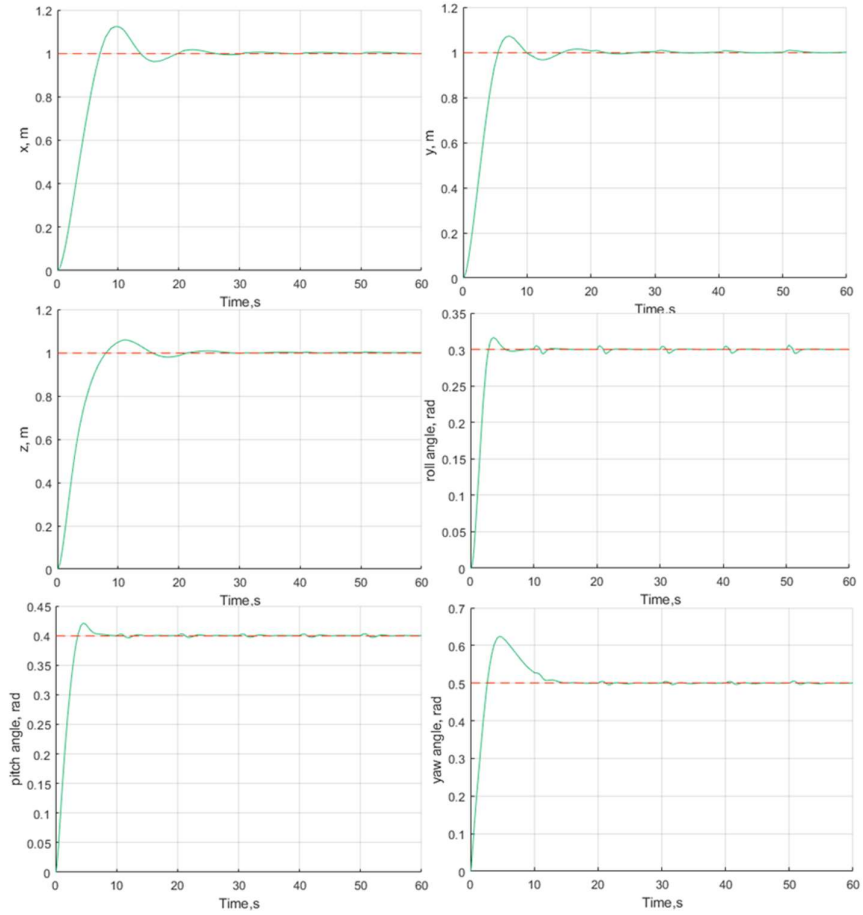

Figure 11: GA-AFC with pulse disturbance

The PSO algorithm was also integrated into the AFC for the optimization of the inertial parameter, *I*'. The parameters used for the algorithm in *m*-script are shown in Table 4.

Table 4: Parameters for the PSO algorithm

Range of *I*'	$100 < I'_x < 200$	$100 < I'_y < 300$	$100 < I'_z < 300$
	$5 < I'_\Phi < 10$	$0 < I'_\Theta < 20$	$0 < I'_\psi < 20$
Number of iterations	50		
Number of particles	20		
Range of weightage value	$0.2 < w < 0.9$		
Acceleration constants	$c_1 = 2$, $c_2 = 2$		
Objective function	$f = \int \lvert e(t) \rvert \, dt$ where $e(t)$ is the error output from simulation		

Figure 12 shows the response of the UROV system using PSO-AFC. The response shows less steady state error, noticeable compensation of the effect of the disturbances and even a fast settling time against pulse disturbance. Although there are no significant differences between each of the AFC scheme tested, the PSO and GA components provide an easier and accurate way to tune the inertia parameter of AFC.

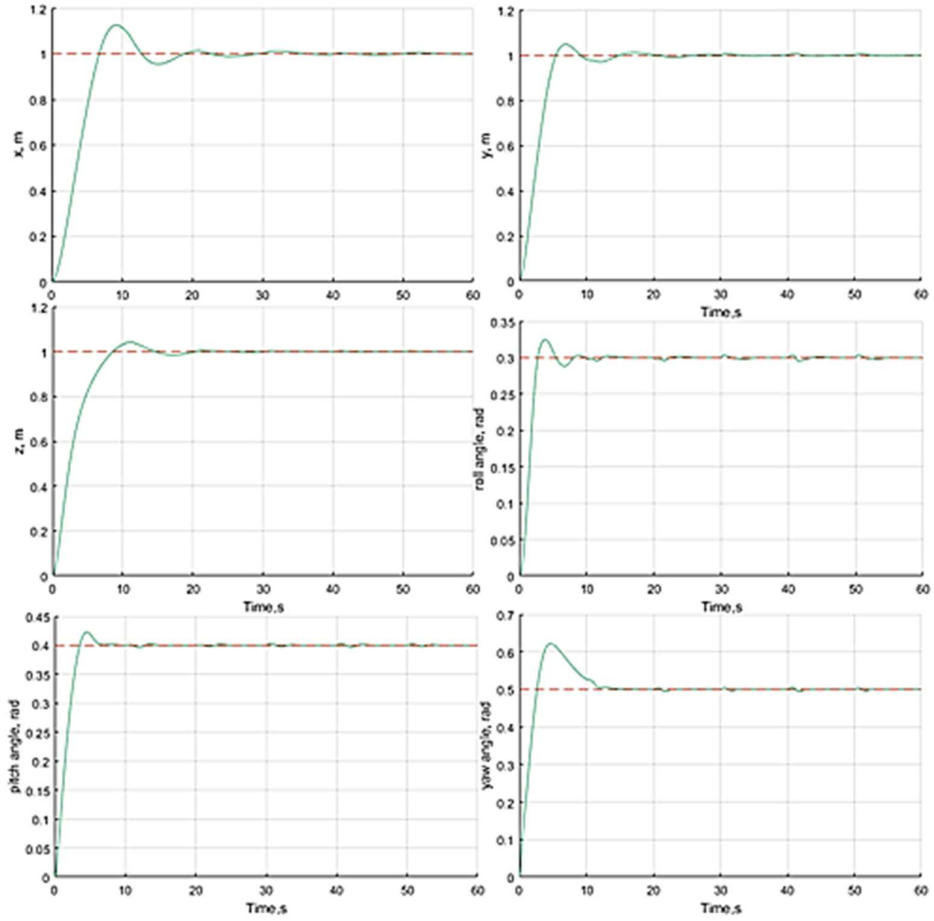

Figure 12: PSO-AFC with pulse disturbance

The values of the inertial parameter estimated through PSO and GA algorithms for each DOF are tabulated along with the crude estimated values in Table 5. The values obtained from each method are quite significantly different. The values of the inertial parameter obtained through PSO and GA may not have converged before the iterations ends; this is because the number of iterations was set to be the stopping criterion of the algorithm.

Table 5: Values of the inertial parameters

DOF	Estimated Inertias, I' (kgm^2)		
	Crude estimate	**GA**	**PSO**
Surge, x	158	183.3099	176.4693
Sway, y	235	212.6464	166.2555
Heave, z	270	273.1276	193.3843
Roll, Φ	6.1023	6.4715	11.1993
Pitch, θ	12.2	11.7253	13.4329
Surge, x	15.83	9.3778	7.2082

Table 6 provides the results related to the responses of the UROV system based on the performance specification curves in terms of the settling times, percentage overshoots and steady state errors. In the study, the steady state errors were assumed and studied at t = 53 s of the simulation period for ease of analysis and comparison. The results show that all the three of the AFC-based schemes show similar trend in performance but obviously exhibit

drastically better performance than the conventional PID controller. For specific DOF, the PSO-AFC scheme seems to give the lowest steady state error and faster settling time.

Table 6: System performance of the different control schemes for all DOF

Control Methods	Criteria	DOF					
		x	*y*	*z*	*Φ*	*θ*	*ψ*
PID	Settling Time (s)	-	-	-	-	-	-
	Overshoot (%)	6.354	5.314	3.731	2.658	4.857	27.286
	Error ($\times 10^{-2}$)	7.882	5.521	2.120	1.004	0.7614	0.1039
AFC	Settling Time (s)	13.852	10.909	11.074	3.284	4.560	10.266
	Overshoot (%)	9.655	7.849	5.609	5.087	5.809	28.229
	Error ($\times 10^{-2}$)	0.4220	0.1607	0.0675	0.0540	0.0212	0.7053
GA-AFC	Settling Time (s)	14.672	11.407	15.968	10.026	4.369	10.259
	Overshoot (%)	12.417	7.169	5.905	5.479	5.252	24.821
	Error ($\times 10^{-2}$)	0.4903	0.3822	0.0532	0.0201	0.0380	0.1255
PSO-AFC	Settling Time (s)	13.997	10.872	10.236	6.042	4.254	10.215
	Overshoot (%)	12.617	4.820	4.226	8.449	5.721	24.540
	Error ($\times 10^{-2}$)	0.4495	0.0070	0.0133	0.1480	0.0585	0.0404

5.0 CONCLUSION

A complete dynamic model of a 6-DOF UROV was investigated for the simulation works through the implementation of a various control schemes related to the PID, PID-AFC and PID-AFC-EC. The simulations were carried out in MATLAB/Simulink computing platform considering a number of different loading and operating conditions. The system responses were simulated, compared and analyzed to review the performance of each control scheme based on their robustness with reference to their settling times, steady-state errors and overshoots. It is clear from the results that the conventional PID controller failed to perform satisfactorily under the effect of disturbances, which exposed its inability to withstand adverse conditions. Applying the AFC scheme yields responses with low steady state errors and the disturbances were effectively compensated. The EC method using GA and PSO was proven as yet another intelligent method to effectively tune the inertial parameter of the UROV. The algorithms served as an optimization tool to compute the parameter in the AFC loop to activate the compensation action. Although all the AFC-based schemes performed equally well compared to the conventional PID, the results of the simulations show that the PSO-AFC scheme can produce response with less steady state error of only 7×10^{-5} m, significantly for the *y*-position.

ACKNOWLEDGMENTS

The author acknowledges and appreciates the Universiti Teknologi Malaysia (UTM) for providing facilities and logistics. The research is supported and funded using a Fundamental Research Grant Scheme (FRGS) grant (FRGS/1/2019/TK03/UTM/01/1).

REFERENCES

1. Azis F.A., Aras M.S.M., Rashid M.Z.A., Othman M.N. and Abdullah S.S., 2012. Problem Identification for Underwater Remotely Operated Vehicle (ROV): A Case Study, *Procedia Engineering*, 41: 554–560.
2. Yamamoto I., Morinaga A. and Ura K., 2019. Development of Remotely Operated Underwater Vehicle and Applications to the Sea, *Procs of the International Offshore and*

Polar Engineering Conference, 1, 1637–1641.
3. Ishaque K., Abdullah S.S., Ayob S.M. and Salam Z., 2010. Single Input Fuzzy Logic Controller for Unmanned Underwater Vehicle, *Journal of Intelligent and Robotic Systems: Theory and Applications*, 59(1): 87–100.
4. Xu J. and Wang N., 2018. Optimization of ROV Control Based on Genetic Algorithm. *Procs of 2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, Kobe.
5. Cui R., Chen L., Yang C. and Chen M., 2017. Extended State Observer-Based Integral Sliding Mode Control for an Underwater Robot With Unknown Disturbances and Uncertain Nonlinearities, *IEEE Transactions on Industrial Electronics*, 64(8): 6785–6795.
6. Soylu S., Proctor A.A., Podhorodeski R.P., Bradley C. and Buckham B.J., 2016. Precise Trajectory Control for an Inspection Class ROV, *Ocean Engineering*, 111: 508–523.
7. Bessa W.M., Dutra M.S. and Kreuzer E., 2010. An Adaptive Fuzzy Sliding Mode Controller for Remotely Operated Underwater Vehicles, *Robotics and Autonomous Systems*, 58(1): 16–26.
8. Javadi-Moghaddam J. and Bagheri A., 2010. An Adaptive Neuro-fuzzy Sliding Mode Based Genetic Algorithm Control System for Under Water Remotely Operated Vehicle, *Expert Systems with Applications*, 37(1): 647–660.
9. Liu S., Liu Y. and Wang N., 2017. Robust Adaptive Self-organizing Neuro-fuzzy Tracking Control of UUV with System Uncertainties and Unknown Dead-zone Nonlinearity, *Nonlinear Dynamics*, 89(2): 1397–1414.
10. Mailah M., Hewit J.R. and Meeran S., 1996. Active Force Control Applied to a Rigid Robot Arm, *Jurnal Mekanikal*, 2(2): 52–68.
11. Priyandoko G., Mailah M. and Jamaluddin H., 2009. Vehicle Active Suspension System using Skyhook Adaptive Neuro Active Force Control, *Mechanical Systems and Signal Processing*, 23(3): 855-868.
12. Mailah M. and Rahim N.I.A., 2000. Intelligent Active Force Control of a Robot Arm using Fuzzy Logic, *Procs of TENCON 2000*, 2: 291–296.
13. Bäck T. and Schwefel H-P., 1993. An Overview of Evolutionary Algorithms for Parameter Optimization, *Evolutionary Computation*, 1(1): 1–23.
14. Kiranyaz S., Ince T. and Gabbouj M., 2014. *Multidimensional Particle Swarm Optimization for Machine Learning and Pattern Recognition*, Adaptation, Learning, and Optimization book series (ALO, volume 15), Springer.
15. Jaen-Cuellar A.Y., Romero-Troncoso R.D.J., Morales-Velazquez L. and Osornio-Rios R.A., 2013. PID-controller Tuning Optimization with Genetic Algorithms in Servo Systems, *International Journal of Advanced Robotic Systems*, 10(9), https://doi.org/10.5772/56697.
16. Mirzal A., Yoshii S., Furukawa M., 2006. PID Parameters Optimization by Using Genetic Algorithm: A Study on Time-delay Systems, *ISTECS Journal*, 8: 34-43.
17. Chin C.S., Lau M.W.S., Low E. and Seet G.G.L., 2006. Software for Modelling and Simulation of a Remotely-Operated Vehicle (ROV), *International Journal of Simulation Modelling*, 5(3): 114–125.
18. Siciliano B. and Khatib O. (Eds.), 2016. *Springer Handbook of Robotics*, Springer.
19. Chin C.S., 2011. Systematic Modeling and Model-based Simulation of a Remotely Operated Vehicle using MATLAB and Simulink, *International Journal of Modeling, Simulation, and Scientific Computing*, 2(4): 481–511.
20. Chin C.S., 2017. ROV Design and Analysis (RDA) - Simulink. Retrieved from: https://www.mathworks.com/matlabcentral/fileexchange/19362-rov-design-and-analysis-rda-simulink. [Accessed: 18 April 2020].