**International Invention of Scientific Journal**
Online ISSN: 2457-0958
Available Online at https://iisj.in **Volume 8, Issue 03 (July-August-September)|2024**|Page: 500-509
Original Research Paper- Electrical Engineering

# Design of PID Controller with Genetic Algorithm to Control the Speed of Remotely Operated Underwater Vehicle

**Corresponding Author**:
Seyyed Mehdi Hosseini
Master of Electrical Engineering, Mechatronics, Golestan University, Gorgan, Iran

Article Received: 28-June-2024 Revised: 19-July-2024 Accepted: 08-August-2024

**ABSTRACT**:
The remotely operated vehicle, abbreviated as ROV, is a group of underwater vehicles that are made of various types and used for various applications. On the other hand, the non-linear dynamic model and time-varying system, the existence of uncertain parameters in the model, and the presence of uncertain environmental disturbances are among the challenges we face when controlling the submarine robot, so designing a suitable controller through which the submarine robot can It is very important to reach the set speed. Normally, the determination of hydrodynamic coefficients of the model is done by advanced laboratory equipment, but due to its high cost, the computational fluid dynamics technique can be used to obtain hydrodynamic coefficients. In this thesis, a control structure is proposed that can properly control the speed of the remotely operated vehicle by using the PID controller along with the genetic optimization algorithm.

***Keywords: remotely operated vehicle, proportional control - integrator - derivative - optimization, genetic algorithm***

## INTRODUCTION:

Over the past few years, intelligent and autonomous underwater robots have emerged in the form of many applications in various industries. Their role is vital in the military industry to find mines, in fisheries and oceanography to explore the sea environment, in the oil industry to monitor and locate oil and gas leaks, etc. [1].

Previous studies have proven that submarine trajectory tracking is of particular importance due to the nonlinear nature of the problem and operational applications in the marine industry. The use of intelligent underwater robots in various projects under the sea has led to the complexity of their structure as a marine vehicle. Also, the application of these vehicles in various and difficult operations such as deep seas, glaciers, etc. makes it difficult for the owners and users to preserve and maintain them [2-3].

The dynamic equations in submarines are nonlinear time variants. This is one of the complications of designing controllers. It is difficult to derive these equations due to the presence of water currents and hydrodynamic damping effects and the many uncertainties in the sea environment. As a result, classical controllers cannot easily handle these conditions. In many different industries, it is customary to rely on guided and controlled equipment without the direct intervention of human force and remotely. For the first time, Ortis and Parenza investigated ways to increase the reliability and safety of intelligent underwater robots. They proposed two protective layers (critical layer and inhibitor layer) for intelligent underwater robots. In the critical layer, the underwater robot is saved from the critical situation. Subsequently, in the inhibitor layer, the necessary command to save before reaching a critical situation is issued by the computer [7]. In [8], Madsen investigated and identified possible defects in underwater robots and presented a logical diagram of underwater robot defects. In [9], Clayton and Gowar investigated the maintenance methods of various propulsion systems in intelligent underwater robots. They also investigated the energy systems and the possibility of their failure as the main cause of the lack of power supply to the propulsion system. Pouder et al. [10] studied increasing the reliability of the intelligent underwater robot called Dorado. They also investigated the defects that occurred in the desired underwater robot for a specific operating time (from 1 to 300 hours). Strutt et al. [11] studied the cause of the loss of the automatic underwater robot (AUTOSUB) in the waters of the areas covered by the glaciers [12]. In [13], Needham presented a fault tree for the controller system software in an intelligent underwater robot. It also examined the mathematical relations of the inter-influence of nodes in this tree and introduced the corresponding solution to increase safety. A research group from Southampton University studied common defects in smart underwater robots. A total of ten reports were compiled, including sixty-three defects and the estimated probability of occurrence for each [14]. Griffiths discusses risk management and the possibility of failure of various parts of an autonomous underwater robot in marine operations. Most of the

failures are related to the construction and connection stages of underwater robot components. Also, the reliability criterion of the system was defined based on the distance traveled [15]. Zhang first presented the fault tree of a submarine and analyzed the reliability of the intelligent underwater robot based on the fuzzy fault tree. Uncertainty is included in the component's failure rate [16].

One of the most common controllers in most industries is the PID controller [3]. This controller can be used as an independent unit or as part of a distributed computer controller system. Adjusting the PID controller parameters to realize desired objectives is challenging. Conventional methods are often unsuccessful in adjusting PID controllers. One of the methods of adjusting the PID controller is to use evolutionary algorithms. These population-based algorithms look for the optimal solution by relying on their random search process. A genetic algorithm is a special type of evolutionary algorithm that uses evolutionary biology techniques such as inheritance, mutation, and Darwin's selection principles to find the optimal formula for predicting or pattern matching. Genetic algorithms are often a good choice for regression-based forecasting techniques [17].

Natural evolution embedded in genetic algorithms (GAs) is a global search method used for optimization. PID controllers have been widely used to control systems. However, the selection based on observation and experimental setting of Kp, Ki, and Kd parameters makes it difficult to realize parameter optimization [4]. To obtain better performance, the actual output must match the target output. For this purpose, a separate controller is required. One of the basic problems of using a controller is setting the parameters. In recent years, various methods have been proposed to adjust PID parameters. The conventional methods for adjusting the PID controller parameters are the Ziegler-Nichols oscillation method, the Ziegler-Nichols reaction curve method, and the Cohen-Coon reaction curve method. However, recent trends show that an effective improvement in parameter setting can be achieved by evolutionary algorithms. These computational algorithms provide better results after completing each iteration. Conventional computational algorithms include ant colony optimization, particle swarm optimization, and genetic algorithm. These methods are based on the behavioral patterns of living organisms [5-8].

Remotely operated vehicles (ROVs) play a vital role in the search, management, and repair of underwater structures related to the oil industry, especially at great depths that are difficult to access. Two important capabilities of industrial ROVs include orientation and dynamic positioning. By definition, the ROV must be able to remain in a specific location for a specified period [18-20]. Also, the dynamic and changing nature of the underwater environment due to the presence of waves at shallow depths causes significant disturbances in the vehicle's performance. In this research, the PID controller adjusted with the genetic algorithm is used to control the ROV system. The key objective is to control the speed of the submarine robot optimally. PID controller makes it difficult to control such processes. A more sophisticated controller such as dead-time compensation or a predictive controller is generally required. Damping processes with complex conjugate poles close to the assumed axis are difficult to control with a PID controller. Processes with such dynamic characteristics are rare in chemical processing industries. However, this is common in mechanical or robotic systems consisting of flexible structures [9-12]. In practical conditions, most underwater industrial robots use PD or PID controllers [7-9]. The reason for this is the simple and efficient structure of such controllers in certain conditions. Typically, pseudo-PID controllers perform well. At the same time, these controllers do not pay attention to the nonlinearities of the system, which in turn causes the destruction of the efficiency and even the instability of the system.

Researchers in [10] have used the PID approach for speed control. Since the linearization requires a vehicle model, the parameters of the robot were identified and used. The results of swimming pool simulations and tests have proven the ability of this controller to control the depth logically. An adaptive control law for underwater vehicles is proposed in [11, 12], which consists of a PD procedure and a suitable adaptive agreement procedure. In this law, the agreement procedure examines the hydrodynamic effects affecting the routing efficiency. Also, two automatic pilot systems with the ability to adjust automatically have been proposed in [13]. The first one is an online two-dimensional linear controller. The latter uses a stable control law based on the first-order approximation of open-loop dynamics and recursive online identification.

The dynamic equations in submarines are nonlinear time-variant. This is one of the complications of designing controllers. Deriving these equations is challenging due to the presence of waves, hydrodynamic damping effects, and high uncertainties in the sea environment. As a result, classical controllers cannot easily handle them. As a result, the proposed controller in combination with the genetic algorithm can guarantee the stability of the robot to some extent. In light of the above, this research aims to design a PID controller based on a genetic algorithm to control the speed of an ROV underwater robot.

**Proposed method**

**Industrial PID algorithm**

In this section, the structure of the time-continuous PID controller is described. The purpose of the PID controller is to measure the process error *e* and calculate the action *u*. Although *u* is considered as the input of the process, it is also considered the output of the controller.

The equation of the time-continuous PID controller is formulated as follows:

$$u = K_c\left(e + \frac{1}{\tau_i}\int e dt + \tau_d \frac{de}{dt}\right) \qquad (1)$$

where the three regulator constants are the proportional gain of the controller ($K_c$), the integral time constant ($\tau_i$), and the derivative time constant ($\tau_d$). The time unit of two other constants for industrial applications is often considered as "minute". The Laplace transform of the ideal PID controller in Relation (1) is defined in the form of Relation (2).

$$\frac{U(s)}{E(s)} = C(s) = K_c(1 + \frac{1}{\tau_i s} + \tau_d s) \quad (2)$$

Figure (1) of the PID controller is also called parallel PID form. We arrange the Relation (2) in the form of a conventional pole-zero transfer function in Relation (3).

$$C(s) = \frac{K_c(\tau_i \tau_d s^2 + \tau_i s + 1)}{\tau_i s} \quad (3)$$

The PID controller is not ideal. The order of the numerator of the fraction (2) is greater than the order of the denominator of the fraction (1). We have a pole in $S = 0$. When we want to build such controllers, we physically need a suitable transfer function. Therefore, we need to slightly change the ideal PID transfer function. The PID transfer function in Relation (2) contains a pure derivative term with $\tau_d s$. Such a term cannot be implemented physically. A change in the set point leads to a large change in the manual variables. The PID block diagram is represented in Figure (1).

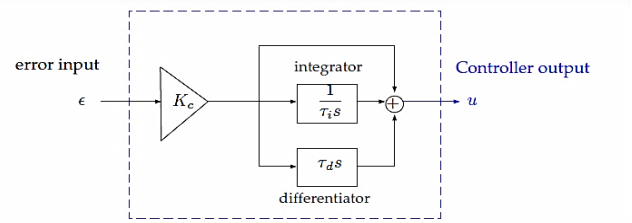


Figure (1): PID block diagram [7]

Many approximations have been proposed in commercial controllers to address derivative problems. Most schemes simply add a minor factory lag term to the derivative term. Therefore, instead of using the derivative term $\tau_d s$, Relation (4) has been used.

$$derivative\ term = \frac{\tau_d s}{\frac{\tau_d}{N} s + 1} \quad (4)$$

where N is a large value ranging from 10 to 100. By using this derivative term, the PID transfer function related to Relation (2) or Relation (3) is modified as follows.

$$C(s) = K_c(1 + \frac{1}{\tau_i s} + \frac{\tau_d s}{\frac{\tau_d}{N} s + 1}) = K_c(\frac{(\tau_i \tau_d + \tau_i \tau_d N)s^2 + (\tau_i N + \tau_d)s + N}{\tau_i s(\tau_d s + N)}) \quad (5)$$

The Relation (5) can be realized physically. When $N \to \infty$, the PID controller related to Relation (6) converges to the state of Relation (4). The PID controller with the derivative filter of Relation (3) is reduced to the PI controller if $\tau_d = 0$.

Another form of series PID controller is sometimes used according to Relation (6).

$$G_c'(s) = K_c'(1 + \frac{1}{\tau_i' s})(1 + \tau_d' s) \quad (6)$$

where the constants of the series PID controller $K_c', \tau_i', \tau_d'$ correspond to the constants of the main PID controller $K_c, \tau_i, \tau_d$. A block diagram of the series PID controller is represented in Figure (2).

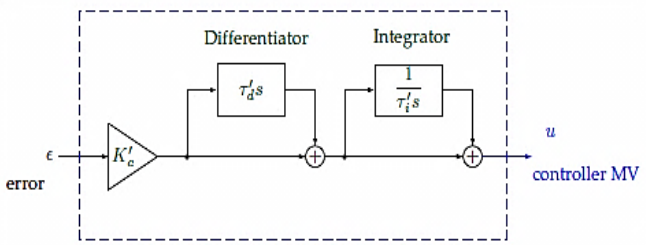


Figure (2): A series PID controller [7]

The series PID controller defined in Relation (6) can be represented in parallel form as follows [1]:

$$K_c = K_c' \frac{\tau_i' + \tau_d'}{\tau_i'}, \tau_i = \tau_i' + \tau_d', \tau_d = \frac{\tau_i' \tau_d'}{\tau_i' + \tau_d'} \quad (7)$$

However, the reverse is not required.

$$K_c' = \frac{K_c}{2}(1 + \sqrt{1 - 4\tau_d / \tau_i}) \quad (8)$$

$$\tau_i' = \frac{\tau_i}{2}(1 + \sqrt{1 - 4\tau_d / \tau_i}) \quad (9)$$

$$\tau_d' = \frac{\tau_i}{2}(1 - \sqrt{1 - 4\tau_d / \tau_i}) \quad (10)$$

The series form can only be realized when $\tau_i > 4\tau_d$. For this reason, the series form of the PID controller is less commonly used. Note that both forms are similar when the derivative part is not used.

Genetic algorithm

A genetic algorithm is a search algorithm based on natural selection and genetic characteristics. The GA method is a computational algorithm inspired by the genetics of the human body. After each iteration, a better result is expected. The error of the results is constantly evaluated. The best solution is considered

for the next iteration based on the selection criteria. GA randomly generates the initial population of PID controller parameters based on the calculation of selection, cross-over, and mutation. Therefore, the values of the controller parameters are optimized. The value of the mean square error (MSE) is included as an efficiency criterion. The general principles of the algorithm of this software are based on the principles of the genetic algorithm. The general steps of the genetic algorithm are as follows:

- Random generation of an initial population of solutions including *n* chromosomes
- Examining the fitness function for each chromosome in the population (usually the fitness function is proportional to the inverse of the squared error. A solution with a small squared error has a high fitness value).
- Creating a new population based on repeating the following four steps:
    - Selection of two parent chromosomes from a population based on their fitness.
    - Applying the cross-over operator on the parents to create new children (that is, generating the next generation of solutions from among the existing solutions).
    - Considering the possibility of mutation and change of children to produce the next generation of solutions.
    - Replacement of new children in the new population.
- Using the new population for the next iterations of the algorithm.
- Stopping the algorithm if the termination conditions are fulfilled (such as the algorithm execution time, the number of generated generations, the objective function reaching a certain threshold, the convergence of the error criterion, and others).

**Simulation and results**

In this section, the simulation results of PID controller design for speed control for an underwater robot system are presented. Subsequently, the simulation results related to the adjustment of the PID controller parameters based on the genetic algorithm are presented. Finally, the simulation results of the two scenarios are compared. The parameters of the Coxswain underwater robot are presented in the following tables.

$$M = \begin{bmatrix} m-X_{u} & 0 & 0 & 0 & 0 & 0 \\ 0 & m-Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & m-Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx}-K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy}-M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz}-N_{\dot{r}} \end{bmatrix}$$

B- Matrix C

Table (1): hydrodynamic parameters [1]

| | | |
|---|---|---|
| $X_{u} = -72kgs^{-1}$ | $X_{\dot{u}} = -29kg$ | $X_{u\|u\|} = 0kgm^{-1}$ |
| $Y_{v} = -77kgs^{-1}$ | $Y_{\dot{v}} = -30kg$ | $Y_{v\|v\|} = 0kgm^{-1}$ |
| $Z_{w} = -95kgs^{-1}$ | $Z_{\dot{w}} = -90kg$ | $Z_{w\|w\|} = 0kgm^{-1}$ |
| $Kp = -40kgms^{-1}$ | $K\dot{p} = -5.2kgm$ | $K_{p\|p\|} = 0kgm^{-1}$ |
| $Mq = -30kgms^{-1}$ | $M\dot{q} = -7.2kgm$ | $M_{q\|q\|} = 0kgm$ |
| $N_{r} = -30Kgms^{-1}$ | $N_{\dot{r}} = -3.3Kgm$ | $N_{r\|r\|} = 0kgm$ |

Table (2): moment of inertia [1]

| | |
|---|---|
| $I_{xx} = 1.32Kgm^{2}$ | $I_{xy} = I_{yx} = 0Kgm^{2}$ |
| $I_{yy} = 2.08Kgm^{2}$ | $I_{yz} = I_{zy} = 0Kgm^{2}$ |
| $I_{zz} = 2.32Kgm^{2}$ | $I_{yz} = I_{zy} = 0Kgm^{2}$ |

Table (3): center of gravity and center of buoy [1]

| Center of gravity | Center of buoy |
|---|---|
| $x_{G} = 0m$ | $x_{B} = 0m$ |
| $y_{G} = 0m$ | $y_{B} = 0m$ |
| $z_{G} = 0m$ | $z_{B} = -0.1m$ |

Table (4): Position of propellers [1]

| Rear left | Rear right |
|---|---|
| $x_{F1} = 0m$ | $x_{F2} = 0m$ |
| $y_{F1} = -0.175m$ | $y_{F2} = 0.215m$ |
| $z_{F1} = -0.1m$ | $z_{F2} = -0.1m$ |
| Sideways (F3) | Vertical (F4) |
| $x_{F3} = 0.135m$ | $x_{F4} = -0.022m$ |
| $y_{F3} = 0m$ | $y_{F4} = 0m$ |
| $z_{F3} = 0.07m$ | $Z_{F4} = 0m$ |

Table (5): Coxswain robot parameters [1]

| |
|---|
| $\rho = 1024Kgm^{3}$ |
| $m = 98.5kg$ |
| $g = 9.81ms^{-2}$ |

The parameters of the Coxswain hydrodynamic model are set as follows [1]:

A- Matrix M

$$C(v)=\begin{bmatrix} 0 & 0 & 0 & 0 & m\omega+Z_{\dot{\omega}}\omega & -mv+Y_{v}v \\ 0 & 0 & 0 & -m\omega+Z_{\dot{\omega}}\omega & 0 & mu-X_{u}u \\ 0 & 0 & 0 & mv-Y_{v}v & -mu+X_{u}u & 0 \\ 0 & 0 & 0 & 0 & -mu+X_{u}u & -I_{yy}q-M_{q}q \\ 0 & 0 & 0 & -I_{zz}r+N_{r}r & 0 & I_{yy}q-M_{q}q \\ 0 & 0 & 0 & I_{yy}q-M_{q}q & I_{xx}p+K_{p}p & 0 \end{bmatrix}$$

C-Matrix D

$$D(v)=-diag[X_{u}+X_{u|u|}|u|,Y_{v}+Y_{v|v|}|v|,Z_{w|w|}|w|,K_{p}+K_{p|p|},M_{q|q|},N_{r}+N_{r|r|}|r|]$$

D-Vector $g(\eta)$

$$g(\eta)=\begin{bmatrix} 0 \\ 0 \\ 0 \\ -\bar{B}\bar{G}_{y}Wc\theta s\phi \\ \bar{B}\bar{G}_{z}Ws\theta+\bar{B}\bar{G}_{z}Wc\theta\phi \\ -\bar{B}\bar{G}_{x}Wc\theta s\phi-\bar{B}\bar{G}_{y}Ws\theta \end{bmatrix}$$

where

$\bar{B}\bar{G}=[\bar{B}\bar{G}_{x} \quad \bar{B}\bar{G}_{y} \quad \bar{B}\bar{G}_{z}]^{T}=[X_{G}-X_{B},\ Y_{G}-Y_{B},\ Z_{G}-Z_{B}]^{T}$ and $c\theta=\cos\theta, s\theta=\sin\theta$. Also, the weight of the robot is defined as $W=mg$ where $g$ is the gravitational acceleration constant.

**Simulation results of speed control of an underwater robot system**

This section presents the simulation results of the conventional PID controller and PID controller adjusted by a genetic algorithm to control the speed of the ROV underwater robot. The key objective here is speed control. The speed value changes from an initial state to a final state. It is assumed that the robot starts its movement with the initial speed and reaches the desired final speed equal to $[u,vlinear,w,p,q,r]=[1.1,0.1,1.3,1.4,1.5,1.4][m/s,rad/s]$. The objective is to reach the desired final speed. The proposed relation for speed control is presented in (11).

$$\tau_{PID}=C_{\eta}\eta_{2}+D_{\eta}\eta_{2}+g_{\eta}-k_{p}\tilde{v}-k_{d}\int\tilde{v}-k_{i}\dot{\tilde{v}} \qquad (11)$$

where $\tau_{PID}$ is the PID controller vector and $C_{\eta}=J^{-1}(C_{v}-M*J^{-1}*\dot{J})J^{-1}$, $g_{\eta}=J^{-1}g_{1}$ and $\tilde{v}=v-v_{desired}$ is the speed error. The parameter $\eta_{2}$ is obtained by solving the following relation.

$$\dot{\eta}_{2}=(M^{-1}_{\eta})(\tau-C_{\eta}\eta_{2}-D_{\eta}\eta_{2}-g_{\eta}) \qquad (12)$$

We can control the underwater robot by relying on these equations and the equations given in the second section.

**A. Speed control with conventional PID controller**

First, we set the parameters of the PID controller arbitrarily without optimization so that appropriate responses are obtained. The adjusted PID controller parameters are considered as follows.

$$kp=[25 \quad 25 \quad 10 \quad 5 \quad 5 \quad 4]$$

$$kd=[5 \quad 5 \quad 4 \quad 5 \quad 5 \quad 5]$$

$$ki=[0.01 \quad 0.01 \quad 0.01 \quad 0.01 \quad 0.01 \quad 0.01]$$

The simulation time and time step size are set to 20 and 0.01 seconds, respectively. The PID controller has been applied to control the speed of the underwater robot and the state variables and control forces have been obtained. The results are presented below. The speed curves of u, Vlinear, and w along with their desired values including ud, vd, and wd are reflected in Figure (3). The key objective is to tend the speed values towards zero.

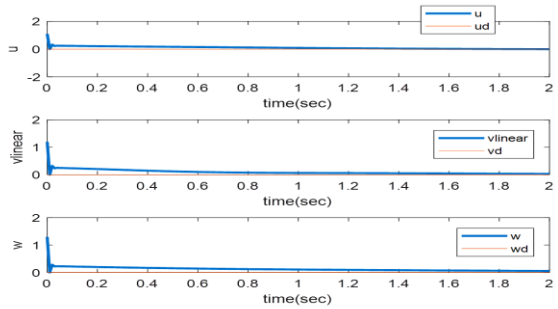


Figure (3): u, vlinear, and w speeds resulting from the conventional PID controller

According to Figure (3), the state variable u has reached zero after 3 seconds. The state variable Vlinear has reached a constant value after 3 seconds. The state variable w has reached zero in about 3 seconds. The curves of the state variables including the speeds r, q, and P are represented in Figure (4).

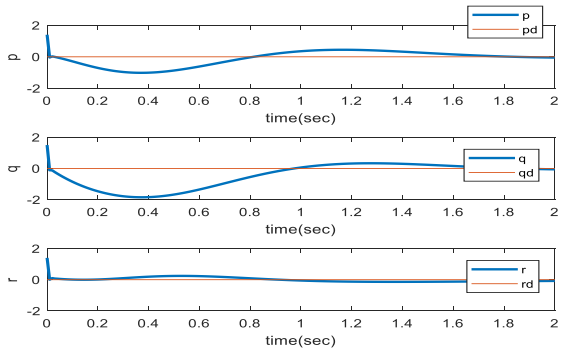


Figure (4): r, q, P speeds resulting from the conventional PID controller

According to Figure (4), the state variable p has reached zero after 2 seconds. The speed values q and r have reached a constant value after 2 seconds. Also, X, Y, Z, and K, M, and N control forces are shown in Figures (4) and (5) respectively.

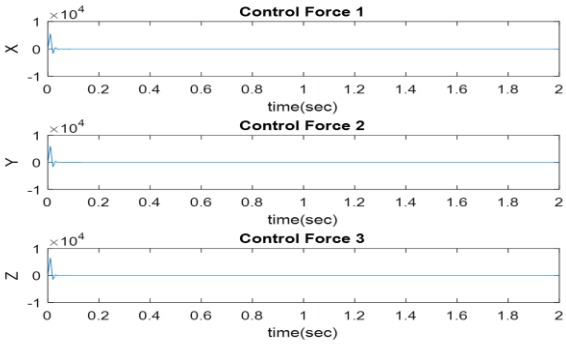


Figure (5): X, Y, and Z control forces resulting from conventional PID

According to Figure (5), control forces X, Y, and Z initially had large values, which have gradually tended to zero over time.

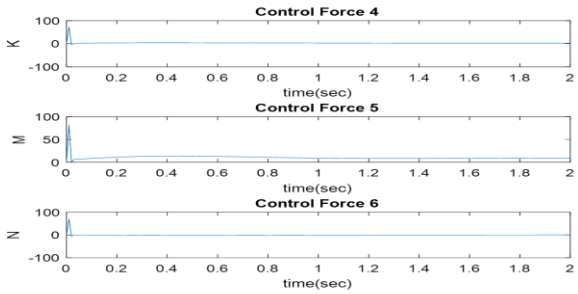


Figure (6): control forces K, M, N resulting from conventional PID

According to Figure (6), the value of the control force K was initially large, which then decreased and reached a constant value. The control force M has faced many changes at first and gradually increased, finally, this variable has reached the value of 10. Also, the value of control force N was initially large. However, it decreased after 0.05 seconds and finally reached zero.

B. Speed control with PID controller adjusted with the proposed genetic algorithm

A cost function is defined for speed errors. Speed errors are indicated by $\tilde{v}$. The proposed cost function includes the mean square errors of the speed. The speed vector of the underwater robot is formulated as $v = [u, vlinear, w, p, q, r]$. Also, the speed errors are defined in the form in $\tilde{v} = v - v_{desired}$ where $v_{desired}$ is optimal speed and is considered equal to $v_{desired} = [0,0,0,0,0,0]$. The proposed cost function includes the minimization of the square error of the speed state variables. The cost function is formulated in the form of Relation (13).

$$J = mean(\tilde{v})^2 = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{6}(v_i^j - v_d^j)^2 \qquad (13)$$

where $mean$ is mean operator and $\tilde{v}$ indicates the speed error. Indices *i* and *j* indicate the sample number and state number, respectively. Also, $v_i^j$ and $v_d^j$ respectively indicate the speed of the i[th] sample of the j[th] state of the underwater robot and the optimal speed of the j[th] state of the underwater robot. Also, N is the total number of simulation samples. This number depends on the size of the time step. The index *i* is calculated from the 1[st] to the N[th] sample. The index *j* reflects the state number calculated from 1 to 6. In the cost function, the squared errors of all 6 states are calculated and aggregated for all samples. Finally, the mean value is calculated.

The role of the genetic algorithm is to determine the PID coefficients. Due to the existence of 6 states of the differential equation, 6 states can be defined in the controller. Also, kp, ki, and kd coefficients are defined for each state. Therefore, a total of 18 coefficients must be evaluated. These 18 coefficients are initialized by the genetic algorithm so that the lowest value for the cost function is obtained. At the point where the cost function is minimized, the coefficients found are stored as optimal PID coefficients. Then these coefficients are fed to the robot model and the program is executed. Finally, the outputs including the robot's speeds are recorded and the corresponding graphs are drawn.

In light of the above, the input of the genetic algorithm includes the PID controller robot model and the calculation of the cost function. PID coefficients are the outputs of the genetic algorithm. Based on the generated outputs, the program is re-executed and the results are recorded.

In the genetic algorithm, the maximum number of iterations is set to 100 and the population size is set to 30. The probability of mutation and cross-over were considered equal to 0.1 and 0.8, respectively. It should be said that the probability of cross-over in a genetic algorithm is usually a number close to 1. Also, the mutation probability is chosen to be very small. Population size and number of iterations are chosen arbitrarily. The values are chosen in such a way that the optimal solution can be reached. We execute the genetic algorithm by adjusting the parameters of the PID controller. The upper and lower bounds of the PID parameters are considered as follows:

Lb= [0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.001 0.001 0.001 0.001 0.001 0.001];
Ub= [10 10 10 10 10 10 10 10 10 10 10 10 1 1 1 1 1 1 ];
Figure (7) shows the plot of the best cost resulting from the genetic algorithm regarding the number of iterations.

**Upper and lower bounds:**

Upper and lower bounds for kp:
Lb(kp): [0.01 0.01 0.01 0.01 0.01 0.01];
Ub(kp): [10 10 10 10 10 10];
Upper and lower bounds for ki:
Lb(ki): [0.01 0.01 0.01 0.01 0.01 0.01];
Ub(ki): [10 10 10 10 10 10];
Upper and lower bounds for kd:
Lb(kd): [0.001 0.001 0.001 0.001 0.001 0.001];
Ub(kd): [1 1 1 1 1 1];
The resolution for calculating the parameters is set to three decimal places.

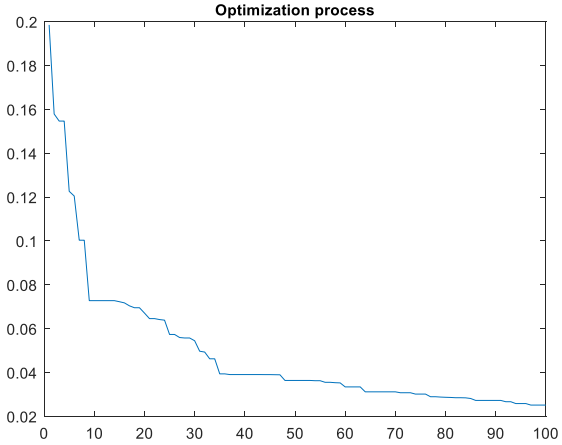


Figure (7): The curve of the best cost regarding the number of iterations using the genetic algorithm

According to Figure (8), the value of the cost function varies from 0.2 to 0.03. The PID parameters adjusted using a genetic algorithm are shown below.

$Kp = [6.2404 \quad 7.4732 \quad 6.4056 \quad 7.2841 \quad 8.3092 \quad 7.5546]$
$Kd = [0.4878 \quad 1.6407 \quad 0.6944 \quad 0.1452 \quad 0.9377 \quad 0.3984]$
$Ki = [0.5786 \quad 0.7179 \quad 0.6967 \quad 0.5258 \quad 0.2447 \quad 0.5167]$

According to these PID controller parameters, we get the results.
In Figures (8) and (9), the plots of u, vlinear, and w speeds and p, q, and r speeds are represented, respectively.

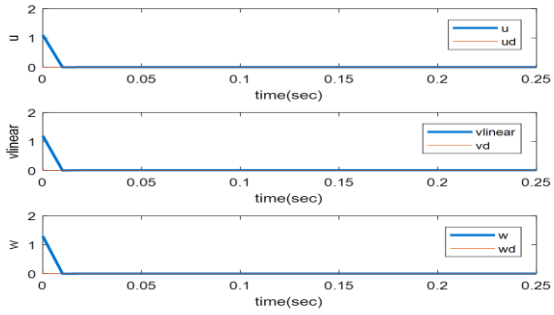


Figure (8): state variables of u, vlinear, and w speeds resulting from the genetic PID controller

According to Figure (8), *u* and *w* have reached zero after 0.02 seconds. Also, the vlinear speed reached zero after 0.02 seconds. In Figure (9), the speeds p, q, and r resulting from the PIDGA controller are represented.

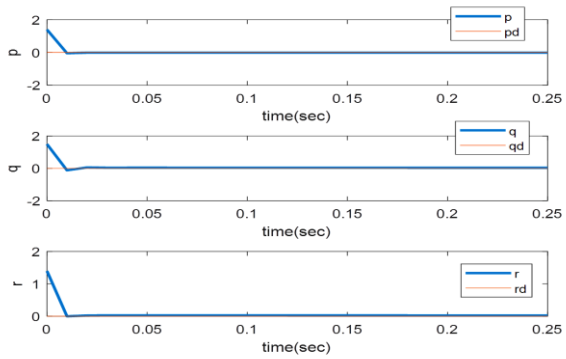


Figure (9): Speed variables p, q, and r resulting from genetic PID controller

According to Figure (9), the values of q and r have reached zero in about 0.02 seconds. Also, the speed of p has reached zero in about 0.02 seconds. In Figure (8), the control forces X, Y, and Z resulting from PIDGA control are represented.

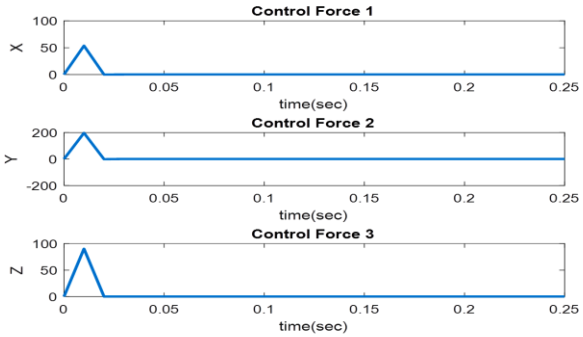


Figure (10): Curve control forces X, Y, and Z regarding time resulting from PID controller based on genetic algorithm

According to Figure 10, all three plots X, Y, and Z have reached zero after 0.02 seconds.


Figure (11): Curve of control forces K, M, and N regarding time resulting from PID controller using genetic algorithm

According to Figure (11), the value of K has reached a constant value after 0.02 seconds. Curve M has reached 100 in less than 0.02 seconds. The curve N has reached the value of 25 in about 0.02 seconds. For a more accurate comparison, the plots obtained from conventional PID and PID adjusted with genetic algorithm are drawn below.

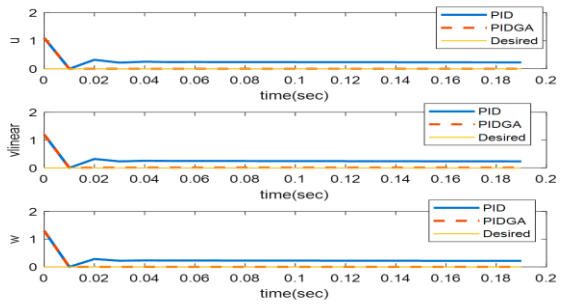


Figure (12): Comparison of plots of u, vlinear, and w speeds obtained from two conventional PID and genetic PID controllers

According to Figure (12), the state variables of u, vlinear, and w speeds obtained from the proposed controller converged to zero earlier than the conventional PID controller.

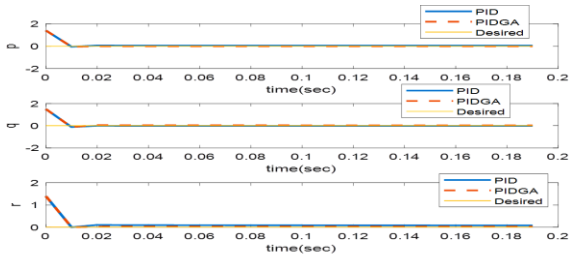


Figure (13): Comparison of p,q, and r plots obtained from two conventional PID and genetic PID controllers

According to Figure (13), the state variables p, q, and r resulting from the proposed controller converge to zero earlier than the conventional PID controller.
According to the obtained results, the PID controller adjusted with the genetic algorithm is more successful than the conventional PID, because the convergence of the speed state variables to the desired values occurred faster. However, the error of the cost function in the proposed controller is 0.0356, while the same value for the conventional PID controller is equal to 0.7583. In short, the proposed method has performed better than the conventional PID regarding the mean square error and the speed of convergence to the desired objective.

Table (6): Comparison between PID and PIDGA controller methods

| Disadvantages | Advantages | Controller methods |
|---|---|---|
| ▪ It is difficult to set the coefficients.<br>▪ Less accuracy and more error compared to PIDGA<br>▪ Slower response compared to PIDGA | ▪ Simple implementation<br>▪ Lack of controller complexity<br>▪ Acceptable solution in most cases | Conventional PID |
| Longer runtime than conventional PID | ▪ Higher speed and accuracy of output response<br>▪ Less error and fluctuations than conventional PID | PIDGA |

Table (7): Comparison of PID and PIDGA controller methods

| $MSE_r$ | $MSE_q$ | $MSE_p$ | $MSE_w$ | $MSE_{\text{Vlinear}}$ | $MSE_u$ | Cost function value (mean squared error( | Controller methods |
|---|---|---|---|---|---|---|---|
| 0.0187 | 0.5323 | 0.1629 | 0.0173 | 0.0129 | 0.0143 | 0.7583 | Conventional PID |
| 0.0067 | 0.0078 | 0.0065 | 0.0056 | 0.0049 | 0.004 | 0.0356 | PIDGA |

According to Table (7), the value of the cost function including the mean square error of the speed, resulting from the proposed PIDGA controller, is lower. It also provides more advantages compared to the conventional PID controller in various applications.

## CONCLUSION

Remotely operated underwater vehicles (ROVs) are widely used in many subsea applications, from environmental inspection to repairing subsea structures associated with the power and oil industries. Often the ROV has to constantly change its operating tool or release the load, which subsequently causes a change in its behavior. In this study, the setting of PID controller parameters for the nonlinear model of an underwater vehicle was investigated based on a genetic algorithm. The underwater vehicle under study is a robot with 6 degrees of freedom. First, the robot model was defined and the basics of PID controller and genetic algorithm were described. Since it is difficult to adjust PID controller gains manually or by trial and error, a genetic evolutionary optimization algorithm was used for this purpose. It is a population-based algorithm based on random search, which is free of gradient information of the cost function. Due to the

characteristics of cross-over and mutation, this algorithm can escape from local optima. As a result, this algorithm was used to solve the problem of adjusting PID controller parameters for remote underwater robot systems. Subsequently, the optimal controller benefits were calculated offline with less computational effort using optimal and actual state variables. To evaluate the performance of the proposed optimization scheme, its results were compared with the conventional PID controller. The results proved that the performance of the PID controller adjusted with the genetic algorithm is better compared to the conventional controller regarding the speed of convergence to optimal values and error minimization. In the future, it is suggested to use new meta-heuristic optimization algorithms to control this type of robot to adjust basic controller parameters such as PID and types of sliding mode. Due to the proper performance of the proposed design, the proposed controller can be used for other underwater robot dynamic models.

## REFERENCES


[1] Rúa, Santiago, and Rafael E. Vásquez. "Development of a low-level control system for the ROV Visor3." International Journal of Navigation and Observation 2016 (2016).

[2] Prasad, M. P. R., and Akhilesh Swarup. "Position and velocity control of remotely operated underwater vehicle using model predictive control." (2015).

[3] M. Kamarlouei, H. Ghassemi, Robust control for horizontal plane motions of autonomous underwater vehicles, Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol. 38, No. 7, pp. 1921-1934, 2016.

[4] F. Repoulias, E. Papadopoulos, Planar trajectory planning and tracking control design for underactuated ROV, Ocean Engineering, Vol. 34, No. 11, pp. 1650-1667, 2007.

[5] D. Qi, J. Feng, J. Yang, Longitudinal motion control of ROV based on fuzzy sliding mode method, Journal of Control Science and Engineering, Vol. 2016, pp. 5, 2016.

[6] R. Cristi, F. A. Papoulias, A. J. Healey, Adaptive sliding mode control of autonomous underwater vehicles in the dive plane, IEEE Journal of Oceanic Engineering, Vol. 15, No. 3, pp. 152-160,1990.

[7] I. Khan, A. Bhatti, Q. Khan, Q. Ahmad, Sliding mode control of lateral dynamics of an ROV, Proceedings of 2012 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST), Islamabad, Pakistan, pp. 27-31, 2012.

[8] T. Elmokadem, M. Zribi, K. Youcef-Toumi, Terminal sliding mode control for the trajectory tracking of under actuated ROV, Ocean Engineering, Vol. 129, pp. 613-625, 2017.

[9] B. K. Sahu, B. Subudhi, Adaptive tracking control of an autonomous underwater vehicle, International Journal of Automation and Computing, Vol. 11, No. 3, pp .299-307 ,2014

[10] J. Kim, H. Joe, S.-c. Yu, J. S. Lee, M. Kim, Time-delay controller design for position control of autonomous underwater vehicle under disturbances, IEEE Transactions on Industrial Electronics, Vol. 63, No. 2, pp. 1052-1061, 2016.

[11] F. Rezazadegan, K. Shojaei, F. Sheikholeslam, A. Chatraei, A novel approach to 6-DOF adaptive trajectory tracking control of an ROV in the presence of parameter uncertainties, Ocean Engineering, Vol. 107, pp. 246-258, 2015.

[12] B. Geranmehr, S. R. Nekoo, Nonlinear suboptimal control of fully coupled non-affine six-DOF autonomous underwater vehicle using the state-dependent Riccati equation Ocean Engineering, Vol. 96, pp. 248-257, 2015.

[13] T. Herlambang, H. Nurhadi, Design of a Sliding PID Controller for The Surge and Roll Motion Control of UNUSAITS ROV, International Journal of Computing Science and Applied Mathematics, Vol. 3, No. 2, pp. 61-64, 2017.

[14] M. Narasimhan, S. N. Singh, Adaptive optimal control of an autonomous underwater vehicle in the dive plane using dorsal fins, Ocean Engineering, Vol. 33, No. 3, pp. 404-416, 2006.

[15] M. S. Naik, S. N. Singh, State-dependent Riccati equation-based robust dive plane control of ROV with control constraints, Ocean Engineering, Vol. 34, No. 11, pp. 1711-1723, 2007.

[16] C. Yu, X. Xiang, Q. Zhang, G. Xu, Adaptive fuzzy trajectory tracking control of an under-actuated autonomous underwater

vehicle subject to actuator saturation, International Journal of Fuzzy Systems, pp. 1-11, 2017.

[17] Z. Chu, D. Zhu, S. X. Yang, Observer-based adaptive neural network trajectory tracking control for remotely operated Vehicle, IEEE transactions on neural networks and learning systems, 2017.

[18] C. Yu, X. Xiang, L. Lapierre, Q. Zhang, Nonlinear guidance and fuzzy control for three-dimensional path following of an under actuated autonomous underwater vehicle, Ocean Engineering, Vol. 146, pp. 457-467, 2017.

[19] X. Xiang, C. Yu, Q. Zhang, Robust fuzzy 3D path following for autonomous underwater vehicle subject to uncertainties, Computers & Operations Research, Vol. 84, pp. 165-177, 2017.

[20] Z. Peng, J. Wang, Output-Feedback Path-Following Control of Autonomous Underwater Vehicles Based on an Extended State Observer and Projection Neural Networks, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017.