# An improved swarm intelligence algorithms-based nonlinear fractional order-PID controller for a trajectory tracking of underwater vehicles

**Mustafa Wassef Hasan, Nizar Hadi Abbas**
Department of Electrical Engineering, College of Engineering, University of Baghdad, Iraq

**Article Info**

***Article history:***

Received Apr 8, 2020
Revised Jun 20, 2020
Accepted Jul 6, 2020

***Keywords:***

Improved GWOA
Improved WOA
Nonlinear-FOPID
PID controller
Swarm intelligence algorithm
Underwater vehicles

**ABSTRACT**

This paper presents a nonlinear fractional order proportional integral derivative (NL-FOPID) for autonomous underwater vehicle (AUV) to solve the path tracking problem under the unknown disturbances (model uncertainty or external disturbances). The considered controller schemes are tuned by two improved swarm intelligence optimization algorithms, the first on is the hybrid grey wolf optimization with simulated annealing (HGWO-SA) algorithm and an improved whale optimization algorithm (IWOA). The developed algorithms are assessed using a set of benchmark function (unimodal, multimodal, and fixed dimension multimodal functions) to guarantee the effectiveness of both proposed swarm algorithms. The HGWO-SA algorithm is used as a tuning method for the AUV system controlled by NL-FOPID scheme, and the IWOA is used as a tuning algorithm to obtain the PID controller's parameters. The evaluation results show that the HGWO-SA algorithm improved the minimal point of the tested benchmark functions by 1-200 order, while the IWOA improved the minimum point by (1-50) order. Finally, the obtained simulation results from the system operated with NL-FOPID shows the competence in terms of the path tracking by 1-15% as compared to the PID method.

***Corresponding Author:***

Mustafa Wassef Hasan,
Department of Electrical Engineering,
University of Baghdad,
Al-Jadriya, Baghdad, Iraq.
Email: m.hasan0902@coeng.uobaghdad.edu.iq

## 1. INTRODUCTION

Due to the importance of AUV in several fields such as industrial applications in oil and gas companies or explore the oceans for creature researches, and on the military applications, many studies have been proposed to solve specific technical problems in AUV. One of these inconveniences the unknown disturbances that may be caused by external impacts such as strong ocean current or the high pressure on the AUV body and internal purposes such as sensor and device's noise or signal interference that may be caused by communication tools and many other causes.

Numerous controllers are presented for the AUV's system. In [1], a model-predictive controller MPC is proposed to control depth signals using quadratic programming, and in [2] the researcher discusses a model-free reinforcement learning algorithm for the AUV. Additionally, in [3] applied a neural network for a consensus multiple tracking AUV problems, in [4] a routing protocol is proposed to solve the end-to-end delay in AUV System, and finally on [5] discussed the adaptive AUV system design. In this research work, an nonlinear fractional order proportional integral derivative (NL-FOPID) AUV is intended to solve the unknown disturbances' problems that affect system response and compared to the relative PID controller.

In order to determine the optimum parameters for both considered controllers, a hybrid grey wolf optimization algorithm with simulated annealing (HGWO-SA) algorithm and an improved whale optimization (IWO) algorithm are proposed. The GWO algorithm is developed by Mirjalili [6], which a wolf structure consists of different types of wolfs (alpha, beta, delta, and omega) and a hunting mechanism is discussed and mimics the leadership hierarchy, the hunting process deals with three main steps that are searching, encircle, and attacking the target. The GWOA has many benefits, such as vast search territory, speed, and accessible to apply. The whale optimization algorithm (WOA) is invented by Mirjalili and Lewis [7], which is based on bubble-net hunting of whales, this method used to mimic the whale creature in exploring for its necessities. The SA algorithm is proposed by Kirkpatrick [8], that is used to enhance the algorithm solution beside the recommended objective function and the Boltzman probability to avoid the local optima trapped in a while exploring the search space. The GWO, SA, and WO algorithms are metaheuristic optimization methods inspired from the behaviour of animals, and other physical phenomena, which are part from other popular swarm optimization algorithms such as a particle swarm optimization [9], artificial bee colony [10], genetic algorithm [11], ant colony optimization [12], and firefly algorithm [13].

The Contribution of the proposed HGWO-SA and improved whale optimization algorithm (IWOA) algorithms is to solve the trajectory tracking problem by finding the optimum parameters for the proposed controller (NL-FOPID) and for the classical PID controller. Additionally, the hybrid and improved swarm algorithms are tested using many benchmark functions to show their effectiveness in comparison with other classical swarm algorithms. The mean reason behind using a PID controller is to demonstrate the capabilities of the introduced controlling scheme through results comparison for underwater vehicle trajectory tracking. The remaining sections of this paper are organized as follows; section 2 establishes the autonomous underwater vehicle model. In section 3, the theoretical basics for the controlling methods are explained. The details of the proposed swarm intelligence algorithms are demonstrated in section 4. Section 5 dedicated for the effectiveness of the proposed algorithms' performance, and the simulation results of the AUV system based on both controlling schemes. Section 6 summarizes the conclusions and the key aspects of this research work

## 2. AUTONOMOUS UNDERWATER VEHICLE MODEL

AUVs model can be described as body-fixed reference (BRF), an inertial reference frame (IRF) or earth fixed frame, AUV founded as translational components and rotational components (suge, sway, heave, roll, pitch, yaw) as shown in Figure 1 [14, 15]. AUV dynamics presented by vector velocity $vv = [X_1, X_2]^T$ where $X_1 = [\text{u, v, w}]^T$ which refer to linear velocities and $X_2 = [p, q, r]^T$ which refer to angular velocities of (suge, sway, heave, roll, pitch, yaw) respectively, while IRF can express as the vector $\eta = [\eta_1, \eta_2]^T$ where $\eta_1 = [X, Y, Z]^T$ and $\eta_2 = [\phi, \theta, \psi]^T$ both $\eta_1$ and $\eta_2$ represents the position and rotational coordinate of AUV.

The transformation of translational velocities between the body-fixed frame and earth fixed coordinates,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = J_1(\eta_2) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{1}$$

where,

$$J_1(\eta_2) = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{2}$$

where, $J_1(\eta_2)$ is an orthogonal matrix, hence, $\left(J_1(\eta_2)\right)^{-1} = \left(J_1(\eta_2)\right)^T$. And the other transformation of rotational velocities between the body-fixed frame and earth fixed frame is

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J_2(\eta_2) \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3}$$

where,

$$J_2(\eta_2) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{4}$$

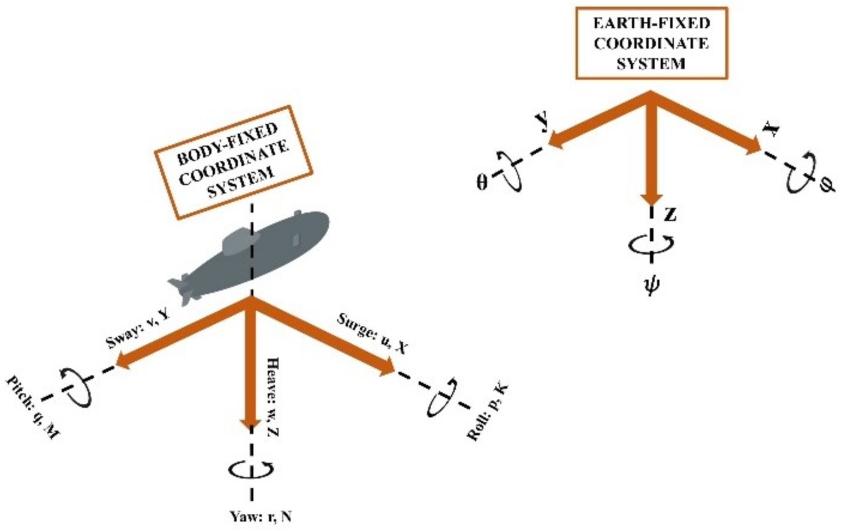

Figure 1. AUV reference frame

Note that when $\theta = \pm 9\mathring{0}$, $J_2$ will be undefined. The locations of the vehicle center of gravity and buoyancy are defined in terms of the body-fixed coordinate system as follows:

$$r_G = \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} \quad r_B = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \tag{5}$$

Vehicle dynamics described by T. I. Fossen (1994) [16] as follows:

$$M\dot{v}v + C(vv)vv + D(vv)vv + g(\eta) = F + D \tag{6}$$

where $M$ is the inertia matrix that consists of a rigid body mass ($M_{RB}$) and added mass ($M_A$) respectively, M Є $\mathbb{R}^{6x6}$, C (vv) is the Coriolis and Centripetal matrix which also consists of a rigid body ($C_{RB}(vv)$) and added mass ($C_A(vv)$), C(vv) Є $\mathbb{R}^{6x6}$, While D (vv) is the hydrodynamic damping of the AUV and consists of linear drag term ($D_l(vv)$) and quadratic term ($D_q(vv)$), D(vv) Є $\mathbb{R}^{6x6}$, F Є $\mathbb{R}^{6x6}$ is the torque force applied on the AUV, and D Є $\mathbb{R}^{6x6}$ are the disturbances that imposed on the system. For the dynamic model given in (6), the system transformed into earth fixed coordinate as:

$$M_\eta(\eta)\ddot{\eta} + C_\eta(vv,\eta)\dot{\eta} + D_\eta(vv,\eta)\dot{\eta} + g_\eta(\eta) = F_\eta(\eta) + D_\eta(t) \tag{7}$$

$$M_\eta(\eta) = J_{(\eta)}^{-T}\, M\, J_{(\eta)}^{-1}$$

$$C_\eta(\eta) = J_{(\eta)}^{-T}\left[C(vv) - M\, J_{(\eta)}^{-1}\, \dot{J}_{(\eta)}\right] J_{(\eta)}^{-1}$$

$$D_\eta(\eta) = J_{(\eta)}^{-T}\, D(vv)\, J_{(\eta)}^{-1}$$

$$g_\eta = J_{(\eta)}^{-T}\, g_{(\eta)}$$

$$F_\eta(\eta) = J_{(\eta)}^{-T}\, F$$

$$D_\eta(t) = J_{(\eta)}^{-T}\, D$$

## 3. CONTROLLING SCHEMES

### 3.1. FOPID controller

Fractional order PID controller mainly formed as five parameters which differ from ordinary PID controller, that generally involves three parameters to enlarge the search space and therefore, achieve robust performance [17, 18]. The transfer function representation of FOPID controller is given by:

$$C(s) = K_p + \frac{K_I}{s^{\lambda}} + K_D S^{\mu} \quad (8)$$

where, $K_p$ is the proportional gain, $K_D$ is a derivative gain, and $K_I$ is integral gain, and (λ, μ) is the FOPID parameters. In case of λ=1 & μ=1, FOPID will act as standard PID controller, when λ=0 &μ=1 will provide a PD controller, and when λ=1 & μ=0 provides a PI controller. Figure 2 shows the FOPID plane.

### 3.2. NL-FOPID controller

The nonlinear FOPID controlling scheme is introduced to enhance the controller capability towards better results, where a nonlinear term is cascaded with the traditional FOPID to improve the nonlinearity behaviour that changes with time or by external effects. The NL-FOPID [19], will work as a self-tuning to handle system complexity due to the disturbances, such that it will decrease system overshoot and neutralize time rising. Therefore, the controller design is given as shown in the following formula,
Let $e_i(t)$ denoted as the error of the system, where $i = 1, 2$
Let $y_i(t)$ indicated as the output response of the AUV, and let the nonlinear term called as $N_i(t)$, where

$$N_i(t) = e_i(t). \exp\left((\tau_{1i}e_i(t) + \tau_{1i}y_i(t).\left(-\tau_{2i}e_i(t) - \tau_{2i}y_i(t)\right)\right) \quad (9)$$

where $(\tau_i)$ is a real number, $\tau_i \in (0, R)$, where $R < \infty$ . The AUV system with NL-FOPID is illustrated in Figure 3.

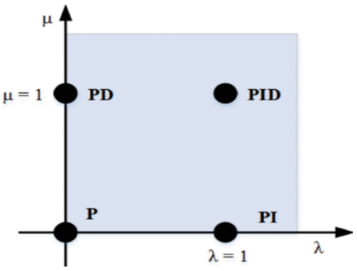

Figure 2. $\lambda$ $\mu$-Plane of FOPID controller

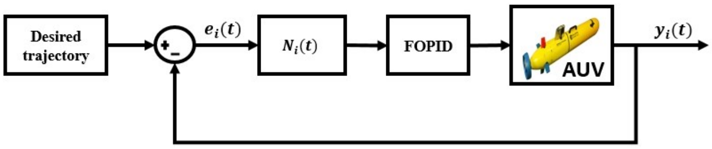

Figure 3. AUV system with NL-FOPID controlling scheme

## 4. PROPOSED SWARM INTELLIGENCE OPTIMIZATION ALGORITHM

### 4.1. Hybrid GWO-SA algorithm

The GWOA tends to mimics the leadership hierarchy were the wolves group divided into (alpha, beta, delta, and omega), where alpha is the fittest solution, beta is the second-best solution, and delta is the third-best solution. In contrast, omega is the remaining wolves that follow the best three individuals [6]. The first step of GWOA is encircling the prey were represented as following mathematical:

$$\vec{D} = \left|\vec{C}\,.\,\vec{X}_P(t) - \vec{X}(t)\right|v \quad (10)$$

$$\vec{X}(t+1) = \vec{X}_P(t) - \vec{A}\,.\,\vec{D} \quad (11)$$

where *t* represents the current iterations, $\vec{X}_P(t)$ is the position of the prey, $\vec{X}(t)$ is the position of the wolf, $(\vec{A}\&\vec{C})$ is the coefficient vector and found by the following:

$$\vec{A} = 2\vec{a}\,.\,\vec{r} \quad (12)$$

$$\vec{C} = 2\,.\,\vec{r} \quad (13)$$

where, $\vec{r}_1$ and $\vec{r}_2$ random numbers from (0 to 1), $\vec{a}$ is linearly decrease coefficient from (2 to 0) during the running iterations. The second step in standard GWO algorithm is hunting were usually the hunting process leading by the alpha wolf with the involvement of beta and delta, but due to unknown location of the prey the best three individuals obtained so far are saved and update the position of the other search agent (omega) by the following:

$$\vec{D}_\alpha = \left|\vec{C}_1\,.\,\vec{X}_\alpha - \vec{X}\right| \quad (14)$$

$$\vec{D}_\beta = \left|\vec{C}_2 \,.\, \vec{X}_\beta - \vec{X}\right| \tag{15}$$

$$\vec{D}_\delta = \left|\vec{C}_3 \,.\, \vec{X}_\delta - \vec{X}\right| \tag{16}$$

$$\vec{X}_1 = \left|\vec{X}_\alpha - \vec{A}_1 \,.\, (\vec{D}_\alpha)\right| \tag{17}$$

$$\vec{X}_2 = \left|\vec{X}_\beta - \vec{A}_2 \,.\, (\vec{D}_\beta)\right| \tag{18}$$

$$\vec{X}_3 = \left|\vec{X}_\delta - \vec{A}_3 \,.\, (\vec{D}_\delta)\right| \tag{19}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1+\vec{X}_2+\vec{X}_3}{3} \tag{20}$$

where, $\vec{C}_1$, $\vec{C}_2$, $\vec{C}_3$, $\vec{A}_1$, $\vec{A}_2$, and $\vec{A}_3$ are randomly generated vectors, $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$ are the positions of alpha, beta, and delta. The GWOA is improved by inserting three modifications, first one focus on enhancing the main wolves (alpha, beta, and delta) locations, such that it is assumed that alpha is the near point to the target, so is supposed to have a wight of (1) and decrease to (1/3) as the number of iteration increase. In contrast (beta, and delta) assumed to be far from alpha and have weight equal to (0) and rise to (1/3) as the number of iteration increase. The improvements are formulated as show in (21-23).

$$\alpha = 1 - \left(\frac{1}{3}\right) * \frac{t}{Max_{iteration}} \tag{21}$$

$$\beta = \frac{1}{3} - \frac{t}{Max_{iteration}} \tag{22}$$

$$\delta = 1 - \alpha - \beta \tag{23}$$

where (21-23) are applied in (20) and yields.

$$\vec{X}(t+1) = \alpha * \vec{X}_1 + \beta * \vec{X}_2 + \delta * \vec{X}_3 \tag{24}$$

The second modification assumed to neglect 10% of the incapable wolves (ill's or relatively old) that have a higher objective function value from the search space to increase the exploitation ability. For that purpose, the SA algorithm is introduced to avoid the GWOA from stacking in the same search arena, where a new solution obtained, which is a neighbour to the best solution obtained so far at every iteration, and the worse solution is developed through the following,

$$p = e^{\frac{-\gamma}{T}} \tag{25}$$

where, ɣ is the change between the objective function for the best solution and the trial solution, while $T$ is the temperature factor and equal to

$$T = (T_0 * alpha) \tag{26}$$

where, $T_0$ is the initial temperature, and $alpha$ is the reduction factor used to reduce $T$ after each iteration.

The new solution obtained (neighbour number) use a DE mechanism to collect the unique number which corresponding to the second modification as shown in,

$$\vec{Z}(t+1) = \vec{X}_1 + K * \left(\vec{X}_2 - \vec{X}_3\right) \tag{27}$$

where,

$$K = rand(\underset{number}{\max}\,,\ \underset{number}{\min}\,, dimension\ size) \tag{28}$$

$K$ is a random number value that in-between ( $\underset{number}{\max}$ & $\underset{number}{\min}$ ) and have size equal to dimension size. After that, a crossover operation introduced to make sure that the new value of $\vec{Z}(t+1)$ effective in comparison with the standard value $\vec{X}(t+1)$ a new variable called $\vec{\omega}(t+1)$ represent the final choice between either case and equal to,

$$\vec{\omega}(t+1) = \begin{cases} \vec{Z}(t+1) & if \left( rand(j) == randi\left( size\left( row\left( \vec{X}(t+1) \right) \right) \right) \right) or\ rand \leq Pcr \\ \vec{X}(t+1) & elsewhere \end{cases} \tag{29}$$

where, $(rand)$ is a random number between (0,1), $j = 1, 2, 3, \ldots, \left( size\left( row\left( \vec{X}(t+1) \right) \right) \right)$, $(randi)$ is a random integer number, and $Pcr$ is the mutation rate.

Finally, the last improvement is achieved by using the SA algorithm to enhance the best solution obtained so far $(alpha_{pos}, beta_{pos}, and\ delta_{pos})$ of each wolf after each iteration to maximize the exploration and exploitation capabilities. The detailed steps of HGWO-SA algorithm are demonstrated as follows,

Input: The HGWO-SA algorithm external parameters.  
Step 1: Specify the LB, UB and Dim of the selected fitness function and the initial a, A, and C;  
Step 2: Evaluate the fitness of each search agent, where $\mathrm{X}_\alpha, \mathrm{X}_\beta$ and $\mathrm{X}_\delta$ are the first, second, third best individuals in search agent in series;  
Step 3: Begin the main loop and for each search agent calculate α, β and δ by (21 − 23) then Update position of each solution by (24);  
Step 4: Find the worst wolves locations;  
Step 5: select 10% of position size then Update position using SA ;  
Step 6: enhance $\mathrm{X}_\alpha$← using SA , enhance $\mathrm{X}_\beta$← using SA , enhance $\mathrm{X}_\delta$← using SA;  
Step 7: Repeat Step 3 until it reaches the maximum number of iterations;  
Output: The optimum solution.

### 4.2. Improved whale optimization algorithm

The WOA algorithm is used to mimic whale organism in Nature, where the whales hunt in a shrinking circle and on a spiral path as shown in [7],

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \,.\, \vec{D} & if\ p < 0.5 \quad \text{(a)} \\ \overrightarrow{D'} \,.\, e^{bl} .\ \cos(2\pi l) + \vec{X}^*(t) & if\ p \geq 0.5 \quad \text{(b)} \end{cases} \tag{30}$$

where,

$$\vec{A} = 2\,\vec{a} \,.\, \vec{r} - \vec{a} \tag{31}$$

$$\vec{D} = \left| \vec{C} \,.\, \vec{X}_{rand} - \vec{X} \right| \tag{32}$$

where (30 a) represent the shrinking circle and (30 b) represent the spiral path, where $\overrightarrow{X^*}$ is the best solution in position vector, $(\overrightarrow{A}, \vec{C})$ are coefficient numbers, $\vec{D}$ is a global search, $\vec{a}$ is a specific number that decreased in the period of (2,0), $\vec{r}$ a random number between (1,0), b is a constant number for defining the shape of a logarithmic spiral, $l$ is a random number in the range (-1,1), $p$ is a random number between [0,1], and $\overrightarrow{D'}$ is the distance between the whale and the target (prey) and given in the following,

$$\overrightarrow{D'} = \left| \vec{X}^*(t) - \vec{X}(t) \right| \tag{33}$$

The humpback whale locates their prey and encircle them, and represented in (34),

$$\vec{D} = \left| C \,.\, \vec{X}^*(t) - \vec{X}(t) \right| \tag{34}$$

The main problems in WOA are its trail to departure from a large number of local solutions in nonlinear search spaces and the stabilizing issue between the exploration and exploitation. The procedure of the update is that when the algorithm updates the position in each iteration, the values of these updated values can reach a higher value. Such that it may give value that beyond the upper or lower bound of the required functions. Therefore, a random matrix-vector (random whale) is suggested to be initiated from each vector at every loop, as explained below in the following statement,

$$X_{rand} = Array(Integer[random_{number}]) \tag{35}$$

where the $X_{rand}$ represent a random hunter (whale) vector that has only one value to be selected at each $(A \geq 1)$ and neglect the remaining values. And this value might be a not optimal value that makes the search domain converge to the prey for the shrinking circle in the exploration phase. In order to solve this problem, a search domain is updated to obtain the best value as in (36),

$$\overrightarrow{D_{new}} = \left|\vec{C} \,.\, \vec{X}_{best} - \vec{X}\right| \tag{36}$$

Let the developed position of the shrinking encircling step is $h$,

$$\mathrm{h} = X_{best} - A{*}\overrightarrow{D_{new}} \tag{37}$$

A comparison between the value of $h$ and the position of each search agent should be achieved; therefore, if new values of $h$ are smaller than the old position of search agent, it will be updated as a new position. To guarantees that the values of the position will be within the upper and lower bounds for the positive and negative values except when the value of position is positive, and the new value acquired from $h$ is negative so that the recently updated position will update the absolute $h$. The following algorithm represents the running procedure o the IWOA:
Input: The IWO algorithm external parameters.
Step 1: Define the LB, UB and the dimension of the chosen fitness function;
Step 2: calculate the fitness of each search agent and chose the best one;
Step 3: start the main loop at that point for each search agent update the interval range (a), (A, C), (L), and (p);
Step 4: If p below (0.5) and (A) below (1) then update the current position as show in (34);
Step 5: If A exceeding (1), find the value of (36), and the value of (h) using (37) according to the selection process of each search agent using (35);
Step 6: test the current position if it's over the UB or LB define in Step 1, then replace it by another value (h) such that the new value should be between the range of (UB, and LB) otherwise return to Step 5 and choose a new value;
Step 7: If p ≥ (0.5) then update the current position using (30 b);
Step 8: Repeat Step 3 until it reaches the maximum number of iterations;
Output: The optimum solution.

## 5. RESULTS AND DISCUSSION

### 5.1. Performance analysis for the proposed algorithm

In this section, a concise performance comparison is presented, where the proposed algorithm is implemented using Matlab R2018b such that it runs for 30 times within 500 iterations to calculate the average (AVG) and standard deviation (STD) for a set of benchmark functions. The chosen benchmark problems are unimodal, multimodal and fixed dimension's multimodal functions [20-23]. The collected numerical results for proposed algorithms are compared with other basic optimization algorithms that are particle swarm optimization (PSO), differential evolution (DE) [24] and gravitational search algorithm (GSA) [25].

The statistical results for evaluating the proposed HGWO-SA algorithm based on selected benchmark functions are tabulated in Table 1. In Table 1, the functions (F1, F2, F3, and F4) has a dimension size equal to (30), while function (F5) has a dimension size of (4). Figure 4 depicts the best objective function registered for F3 function based on HGWO-SA algorithm and other basic optimization algorithms.

From the result obtained in Table 1 and elucidated in Figure 4, it can be seen that, for example, the average for (F3) using HGWO-SA algorithm is decreased to (16) order compared to standard GWO algorithm that has a (14) order that means the HGWO-SA is improved by (2) order with a minimum number of iteration (around 65 iterations). Therefore, the HGWO-SA algorithm is the nearest one compared to other algorithms towards ($F3_{min} = 0$). Furthermore, the functions (F1, F2, F4, and F5) from Table 1, also demonstrated that the proposed algorithm has the nearest points towards the minimum function points and for different orders.

Table 1. Comparison of HGWO-SA with GWOA, PSO, DE and GSA algorithms

| Function | Metric | HGOWA | GWOA | PSO | DE | GSA |
|---|---|---|---|---|---|---|
| F1 | avg | 4.6586e-13 | 26.9358 | 82.3811 | 165.4646 | 0.4106 |
| Rosenbrock | std | 1.0983e-12 | 0.7633 | 99.3448 | 52.3370 | 0.3802 |
| F2 | avg | 4.1614e-04 | 0.0013 | 0.1571 | 0.0529 | 0.1658 |
| Noise | std | 2.7607e-04 | 7.8060e-04 | 0.0509 | 0.0115 | 0.4588 |
| F3 | avg | 8.8818e-16 | 6.0929e-14 | 0.0868 | 0.0061 | 7.0640e-09 |
| Ackley | std | **0** | 9.3462e-15 | 0.3163 | 0.0015 | 1.2032e-09 |
| F4 | avg | 2.4633e-06 | 0.3216 | 0.3723 | 0.2991 | 0.2730 |
| Happy Cat | std | 1.8171e-06 | 0.0547 | 0.0833 | 0.0328 | 0.0964 |
| F5 | avg | -10.1532 | -9.3135 | -8.1434 | -9.7884 | -6.3921 |
| Shekel 5 | std | 6.9708e-05 | 2.2189 | 2.9738 | 1.2199 | 3.6236 |

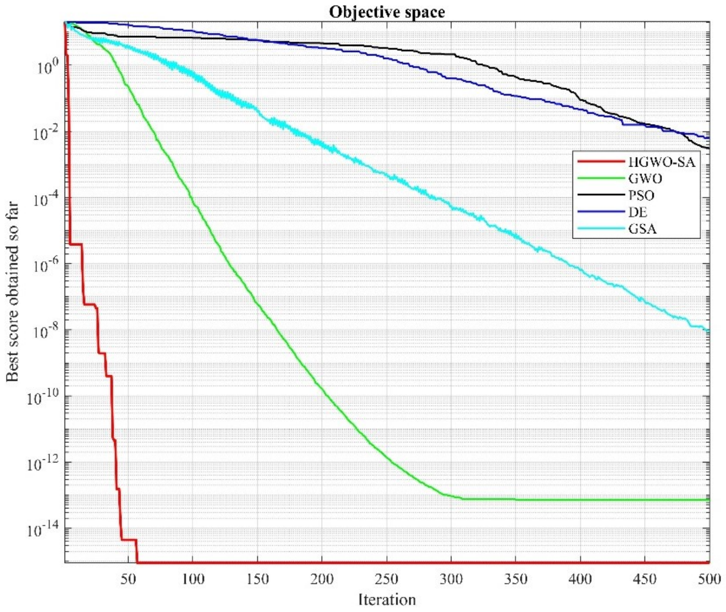

Figure 4. The results of Ackley (F3) function based on HGWO-SA algorithm and other standard algorithms

In Table 2, the numerical results of the proposed second algorithm, which is IWOA, is illustrated based on selected various test functions. The test functions (F1, F2, F3, and F4) used in Table 2, has a dimension size equal to (30), while function (F5) has a dimension size of (4). Figure 5 as shows the best objective function registered for F1 function based on IWO algorithm and other selected standard optimization algorithms. From the result obtained in Table 2 and Figure 5, it can be seen that the average value of the function (F1) using the second proposed algorithm is decreased to (123) order compared to the standard WOA that has (80) order; that's mean the IWOA is improved by (43) order. Therefore, the IWO algorithm is the optimum algorithm compared to other algorithms towards ($F1_{min} = 0$). Moreover, the results of the functions (F2, F3, F4, and F5) also indicated that the improved algorithm had achieved global best values.

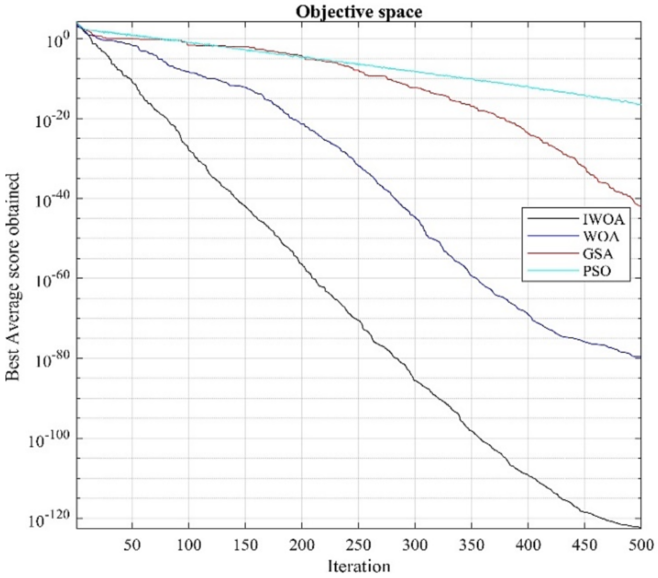

Figure 5. The results of the sphere (F1) function based on IWOA and other standard algorithms

Table 2. Comparison of IWOA with WOA, PSO and GSA algorithms

| Function | Metric | IWOA | WOA | PSO | GSA |
|---|---|---|---|---|---|
| F1 | avg | 3.5275E-123 | 2.6260e-80 | 1.446595e-18 | 8.341100e-41 |
| Sphere | std | 4.963833e-117 | 7.328033e-79 | 8.766226e-19 | 2.218018e-40 |
| F2 | avg | 1.1230E-25 | 46.2694 | 1.0902 | 8.0052 |
| Noise | std | 3.0857E-25 | 31.2569 | 0.2153 | 2.5682 |
| F3 | avg | **0** | 3.7896E-15 | 53.1311 | 29.3181 |
| Ackley | std | **0** | 1.4422E-14 | 11.0473 | 7.3381 |
| F4 | avg | **0** | 0.0040 | 0.0067 | 25.9758 |
| Schwefel | std | **0** | 0.0217 | 0.0086 | 6.4710 |
| F5 | avg | -8.2746 | -6.5921 | -6.8053 | -6.4268 |
| Shekel 7 | std | 2.4917 | 2.8524 | 3.3115 | 3.6015 |

### 5.2. Results and discussions for the designed system

For obtaining the results for the designed system, it's assumed that the AUV is operating at low speed and move at (YAW(ψ)=0.785 deg) so that the Coriolis matrix showed in (7) will not designate to vehicle dynamics as a result $C_\eta(vv,\eta)=0$, while the desired trajectory equal to $r_d=[\sin wt \quad 0 \quad 0 \quad 0 \quad 0 \quad \cos wt]$, the disturbances assumed to equal $D=[-180.3\sin(wt+89) \quad 0 \quad 0 \quad 0 \quad 0 \quad -9.85\sin(wt+86.5)]$, and finally, from [26] applied the following AUV parameters which are shown in Table 3.

Table 3. The Parameters values for the underwater vehicle

| Parameters | Values | Parameters | Values |
|---|---|---|---|
| m | 159 | $Z_{w\|w\|}$ | 90 |
| $I_{xx}$ | 14 | $K_{p\|p\|}$ | 10 |
| $I_{yy}$ | 14 | $K_{q\|q\|}$ | 9 |
| $I_{zz}$ | 14 | $N_{r\|r\|}$ | 14 |
| $X_u$ | 58 | W | 0 |
| $Y_v$ | 58 | B | 0 |
| $Z_w$ | 70 | $X_g$ | 0 |
| $K_p$ | 12 | $Y_g$ | 0 |
| $K_q$ | 13 | $Z_g$ | 0 |
| $N_r$ | 14 | $X_b$ | 0 |
| $X_{u\|u\|}$ | 88 | $Y_b$ | 0 |
| $Y_{v\|v\|}$ | 86 | $Z_b$ | 50 |

After Applying both of the proposed swarm intelligence algorithms to obtain the optimum parameters for the NL-FOPID, and PID controlling schemes, where the objective functions that used to evaluate the performance index (*PI*) are integrated time absolute error (ITAE), integrated absolute error (IAE), and integrated squared error (ISE) [27-29]. The tuned NL-FOPID and PID controllers' parameters based on HGWO-SA algorithm are listed in Table 4, and Table 5 represents the PID parameters that optimized using the IWO algorithm.

In accordance with the values in Tables 3, 4, and 5, the AUV model with controlling schemes is implemented using MATLAB R2018b. Consequently, the simulation results of the designed system are obtained, where Figure 6 discusses the difference between the input for the desired signal ($r_{d6}$) and the output signal for the NL-FOPID and PID controllers which shows that the AUV with NL-FOPID is more stable and close to the reference signal. As an example, for that, the third (+ve) peak for the NL-FOPID has a delay of (0.006) of the reference signal in comparison to the PID that has a delay of (0.012). In contrast, the (-ve) peak has a (0.01) for the NL-FOPID controller and a (0.02) for the PID controller in comparison with the reference signal.

Table 4. NL-FOPID parameter

| NL-FOPID parameters | | | |
|---|---|---|---|
| $Kp_1$ | 53.6345 | $Kp_6$ | -5 |
| $Ki_1$ | -65.8899 | $Ki_6$ | 0 |
| $kd_1$ | -90.5585 | $kd_6$ | -40 |
| $\lambda_1$ | 1.1927 | $\lambda_6$ | 0.9 |
| $\mu_1$ | 0.0642 | $\mu_6$ | 1 |
| $\tau_{11}$ | 0.6420 | $\tau_{16}$ | 0.001 |
| $\tau_{21}$ | -0.4239 | $\tau_{26}$ | -2.5 |
| Number of iterations | | 26 | |

Table 5. PID parameters

| PID parameters | | | |
|---|---|---|---|
| $Kp_1$ | -19.4850 | $Kp_6$ | -19.4850 |
| $Ki_1$ | -19.4850 | $\mathrm{Ki}_6$ | -163.2955 |
| $kd_1$ | -53.2347 | $kd_6$ | -53.2347 |
| Number of iterations | | 28 | |

Figure 7 discusses the difference between the input for the desired signal ($r_{d1}$) and the output signal such that the second (+ve) peak for the designed system with NL-FOPID has a delay of (0.005) of the reference signal in comparison to the PID that has a backlog of (0.03). However, the (-ve) peak for the NL-FOPID very close and stable to the reference signal compared with PID controller that has a delay of (0.02). Figure 8 combines both output signal of ($r_{d1}, r_{d6}$) with the running time *t,* which lead to achieving a helical trajectory, as shown in Figure 8 (a). Moreover, Figure 8 (b) illustrated the circular path it shows that the considered system with NL-FOPID is more stable than the PID controller in spite of some fluctuating at starting due to the disturbance effect.

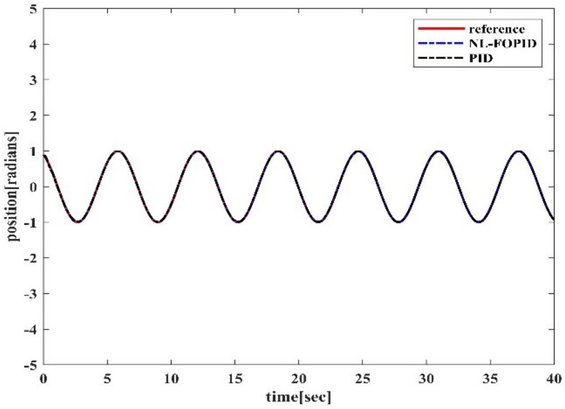

Figure 6. AUV system trajectory of $r_{d6}$

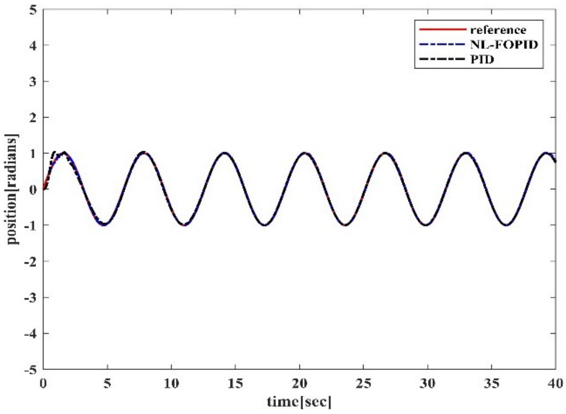

Figure 7. AUV system trajectory of $r_{d1}$

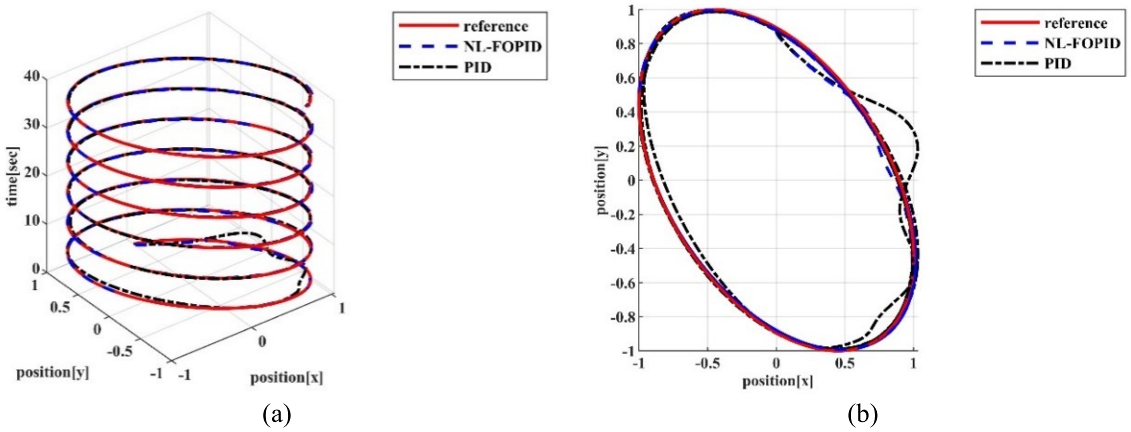

Figure 8. Three-dimensional graph; (a) helical trajectory, (b) circular trajectory

## 4. CONCLUSION

This paper introduced the NL-FOPID controller for the underwater vehicle to solve the path trajectory tracking problem. The proposed controller is compared with the PID controller to show differences betweesn both controllers, such that the proposed controller shows a steady performance that closes to the reference path than the PID controller. An HGWO-SA algorithm presented with three modifications that are modified the wolves positions, neglect 10% of the ineffective wolves in the search space, and enhances the best solution obtained after each iteration for the leading wolves. Moreover, an IWO algorithm is developed with a modified search space to achieve the best individuals in the search domain. Both of the swarm intelligence optimization algorithms are tested using various sets of benchmark functions, where the results show that the HGWO-SA algorithm is improved the minimum point by 20-130% compared to the GWO scheme, while the IWOA improved by 2-50% compared to the WOA. Finally, the results obtained from simulating the system with NL-FOPID controller show that it enhances the system trajectory by 1-15% as compared to the PID controller that not match well with the reference path.

## ACKNOWLEDGEMENTS

We would like to express our gratitude and thanks to the Editors of TELKOMNIKA journal and the anonymous Reviewers for their valuable comments and suggestions.

## REFERENCES

[1] L. V. Steenson, *et al.*, "Experimentally Verified depth regulation for AUVs using constrained model predictive control," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11974-11979 , 2014.

[2] H. Wu, S. Song, K. You, and C. Wu, "Depth control of model-free AUVs via reinforcement learning," *IEEE Transaction on System, Man, and Cybernetics. Systems*, vol. 49, no. 12, pp. 2499-2510, Dec 2019.

[3] J. Cui, L. Zhao, J. Yu, C. Lin, and Y. Ma, "Neural network-based adaptive finite-time consensus tracking control for multiple autonomous underwater vehicles," *IEEE Access*, vol. 7, pp. 33064-33074, March 2019.

[4] W. Farooq, T. Ali, A. Shaf, U. Draz, and S. Yasin, "Atomic-shaped efficient delay and data gathering routing protocol for underwater wireless sensor networks," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 27, no. 5, pp. 3454-3469, 2019.

[5] M. P. Brito, *et al.*, "Adaptive autonomous underwater vehicles: an assessment of their effectiveness for oceanographic applications," *IEEE Trans. Eng. Manag.*, vol. 66, no. 1, pp. 98-111, February 2019.

[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advanced in Engineering Software*, vol. 69, pp. 46-61, March 2014.

[7] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advanced in Engineering Software*, vol. 95, pp. 51-67, May 2016.

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, May 1983.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Network* , vol. 4, pp. 1942-1948, December 1995.

[10] D. Karaboga and B. Basturk, "A Powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization.*, vol. 39, no. 3, pp. 459-471, November 2007.

[11] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," *Addison-Wesley Longman Publishing Co.*, 1989.

[12] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28-39, 2006.

[13] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *International Journal Bio-Inspired Computation*, vol. 2, no. 2, pp. 78-84, March 2010.

[14] C. Shen, Y. Shi, and B. Buckham, "Lyapunov-based model predictive control for dynamic positioning of autonomous underwater vehicles," *in Proc. 2017 IEEE Int. Conf. on Unmanned Syst. (ICUS 2017*, 27-29 Oct. 2017.

[15] N. Fischer, S. Bhasin, and W. E. Dixon, "Nonlinear control of an autonomous underwater vehicle: A RISE-based approach," *in Proc. Am. Control Conf.*, San Francisco, vol. 1, pp. 3972-3977, July 2011.

[16] T. I. Fossen, "Guidance and Control of Ocean Vehicles," Wiley, New York, 1994.

[17] O. W. Abdulwahhab and N. H. Abbas, "A new method to tune a fractional-order PID controller for a twin rotor aerodynamic system," *Arabian Journal for Science and Engineering*, vol. 42, no. 12, pp. 5179-5189, June 2017.

[18] O. W. Abdulwahhab and N. H. Abbas, "A new analytic method to tune a fractional order PID controller," *Journal of Engineering*, vol. 23, no. 12, pp. 1-12, December 2017.

[19] M. Shamseldin, M. A. Ghany, and A. G. Mohamed, "Performance Study of Enhanced Non-Linear PID Control Applied on Brushless DC Motor," *International Journal of Power Electronics and Drive System*, vol. 9, no. 2, pp. 536-545, 2018.

[20] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82-102, 1999.

[21] D. Simon, "Evolutionary optimization algorithms.," John Wiley & Sons, 2013.

[22] M. Molga and C. Smutnicki, "Test functions for optimization needs," *Test Funct. Optim. needs*, vol. 101, 2005.

[23] N. H. Abbas, "Tuning of different controlling techniques for magnetic suspending system using an improved bat algorithm," *Internatinal Journal Electrrical and Computer Engineering*, vol. 10, no. 3, pp. 2402-2415, June 2020.

[24] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal Global Optimization.*, vol. 11, no. 4, pp. 341-359, 1997.

[25] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Information Science.*, vol. 179, no. 13, pp. 2232-2248, June 2009.

[26] J. Vervoort, "Modeling and control of an unmanned underwater vehicle," *Master traineeship report*, New Zealand, 2008.

[27] R. A. Krohling and J. P. Rey, "Design of Optimal disturbance rejection PID controllers using genetic algorithms," *IEEE Transactions on Evolutionary Computer*, vol. 5, no. 1, pp. 78-82, February 2001.

[28] T. Kawabe and T. Tagami, "A real coded genetic algorithm for matrix inequality design approach of robust PID controller with two degrees of freedom," *Proceedings of 12th IEEE International Symposium on Intelligent Control*, 1997.

[29] Y. Mitsukura, T. Yamamoto, and M. Kaneda, "Design of self-tuning PID controllers using a genetic algorithm," *in Proc. American Control Conf.*, vol. 2, pp. 1361-1365, June 1999.