AN ADAPTIVE NEURO-FUZZY BASED ON A FRACTIONAL-ORDER PROPORTIONAL INTEGRAL DERIVATIVE DESIGN FOR A TWO-LEGGED ROBOT WITH AN IMPROVED SWARM ALGORITHM

Mustafa Wassef* **– Nizar Hadi**

Department of Electrical Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq

ARTICLE INFO

***Article history*:**
Received:15.12.2021.
Received in revised form: 10.03.2022.
Accepted:06.02.2023.

Keywords:
Adaptive neural fuzzy fractional order PID (ANFFOPID)
Improved slime mould algorithm (ISMA)
Two-legged robot (TLR)
Proportional integral derivative (PID)

DOI: https://doi.org/10.30765/er.1916

Abstract:

In this paper, an adaptive neuro-fuzzy based on fractional-order proportional-integral-derivative (ANFFOPID) controller with an improved slime mould algorithm (ISMA) for the two-legged robot (TLR) is proposed to achieve the minimum angular displacement error of the joint motors. Achieving such error is considered a challenging and time-consuming process due to the gain values set for the FOPID controller. Thus the neural-fuzzy network is used to provide the FOPID input signals by the adaptive magnitude gains. The adaptive mechanism depends on the ISMA to train the neural network weights. The outstanding properties of the ANFFOPID controller are evaluated by comparing the proposed controller with other existing work that is modified chaotic invasive weed optimization based on neural network (MCIWO-NN) for various walking terrains that are flat surface, stair ascending, and stair descending. Finally, the results obtained show the effectiveness of the ANFFOPID controller.

1 Introduction

In recent years, the two-legged robot industry has been overgrowing due to its benefits in scientific and social life applications such as cleaning applications for specific types of industries or health care applications, serving applications, etc. Two-legged robots should handle different terrains and surfaces in natural environments, e.g. flat surfaces, slop surfaces, staircases, roads, sands, and stones. Although of such features for two legged-robots, it is considered challenging to obtain the task mentioned above. Thus, most of the researchers provide different solutions and techniques to solve a certain problem.

These techniques and solutions depend on designing new controlling approaches to enhance the balance of the two-legged robot. The uneven terrain requires a new control to increase system stability [1]. In two-legged robots, the walking patterns related directly to the robot balance, the walking problem is considered a crucial problem that should be solved with high reliability. In [2], the authors present a zero moment point (ZMP) concept to obtain a dynamically balanced gait for the two-legged robot due to the significant criterion for robot walking stability. In [3], the authors introduce a new concept for online gait adaptation to the leg robots capabilities. In [4], the authors proposed a suitable proportional integral derivative (PID) controller for the two-legged robot to achieve the required positions for the joints to achieve stable walk patterns. In [5], the authors present a fractional-order proportional derivative (FOPID) controller for a hexapod robot to achieve the best performance for different ground properties.

The tuning process of different controllers such as PID and FOPID need to be highly taken care of to present the best performance of autonomous robotic systems. Thus, many researchers proposed different controller approaches such as the state-dependent Riccati equation [6], Kalman filter and backstepping design

* Corresponding author
E-mail address:mustafawassef@yahoo.com

[7], adaptive neural network design [8], a neural approximation based on model predictive tracking control [9], genetic algorithm (GA) and fuzzy approach [10, 11].

Moreover, tuning the gains of the proposed controller's for two-legged robots were highly researched through the recent years, such that the authors in [12] present particle swarm optimization to tune the parameters of the PID controller, in [13], the authors proposed a hybrid salp swarm algorithm with grey wolf optimizer algorithm (HSSGWOA) to tune the integral sliding mode controller (ISMC), in [14], the authors present a hybrid dragonfly optimization and differential evolution algorithm for obstacle avoidance trajectory planning problem, in [15], the authors introduce a modified chaotic inverse weed optimization (MCIWO) for training the neural network with PID controller, in [16] the authors proposed an improved slime mould algorithm (ISMA) to adjust the parameters of FOPID controllers, while the authors in [17] present a slime mould algorithm to tune a magnetic levitation system.

Related to this work, many researchers across the globe present and develop different controllers such as [18–20], where the authors proposed on gait generation for a 7-DOF robot. While in [21, 22], they present a PID controller with ZMP to test the various joints of motors. Motivated by the above analysis, we proposed a new technique to reduce the error between the desired angle of the joint motor and the final displacement angle of the joints. This technique ensures that the value of the error will be smaller and optimized. The most challenging problem is to tune the two-legged robot gains controller with the proper values to present the best effort for the different terrains. Thus, this problem is solved by presenting a neuro-fuzzy network to present the best gains possible to the FOPID controller, where the fuzzy network firstly selects the best fuzzy rules for the optimized error techniques and then provides the neural network with the required results inputs weights. At the same time, the neural network weights are trained using an improved slime mould algorithm adaptively. The improved slime mould algorithm is tested for its effectiveness to the TLR by comparing it with other optimization algorithms via the fitness function and the required torque by the motors joints.

The remainder of this work is structured as follows. Section 2 presents the two-legged robot model used in this work. The fractional-order proportional integral derivative (FOPID controller design are stated in Section 3. The Neural fuzzy network with FOPID controller design is stated in Section 4. In Section5, the improved slime mould algorithm is presented. Then, the fundamental result of this work and some numerical simulations to verify the efficiency of the presented controller are proposed in Section 6. At the end, the conclusion part is presented in Section 7.

2 Modelling of two-legged robot

In this work, an 18-DOF of two legged-robot is used, where each leg is 6-DOF, while each hand has 3-DOF. The two-legged robot dynamics were obtained using the Lagrange-Euler formulation. The Lagrange-Euler formulation is used to supply the joints of the two-legged robot with the required torques [15], [23]. Figure 1 show the two-legged robot structure. The frontal plane is considered for the motion of the two-legged robot. The inverse kinematics concept has been used to calculate the direction of each joint for the TLR. To define the joint angles of TLR, the following mathematical expression is used [23]:

$$\Omega_4 = \sin^{-1}\left(\frac{A_1 B_3 \sin\beta + L_1 B_1 \cos\beta\,(B_4 + B_3\cos\beta)}{(B_4 + B_3\cos\beta)^2 + (B_3\sin\beta)^2}\right) \tag{1}$$

where $A_1 = B_4\cos(\Omega_4) + B_3\cos(\Omega_3)$, $L_1 = B_4\sin(\Omega_4) + B_3\sin(\Omega_3)$, Ω_4 and Ω_3 are the joint angles of the swing leg.

$\beta = \Omega_4 - \Omega_3 = \cos^{-1}\left(\frac{{A_1}^2 + {L_1}^2 - {B_4}^2 - {B_3}^2}{2B_4B_3}\right)$. While $\Omega_3 = \Omega_4 - \beta$, similarly the joint angles of the standing leg (Ω_{10} and Ω_9) can be obtained as follows:

$$\Omega_{10} = sin^{-1}\left(\frac{A_2 B_9\, sin\,\beta + L_2(B_{10} + B_9\, cos\,\beta)}{(B_{10} + B_9\, cos\,\beta)^2 + (B_9\, sin\,\beta)^2}\right) \tag{2}$$

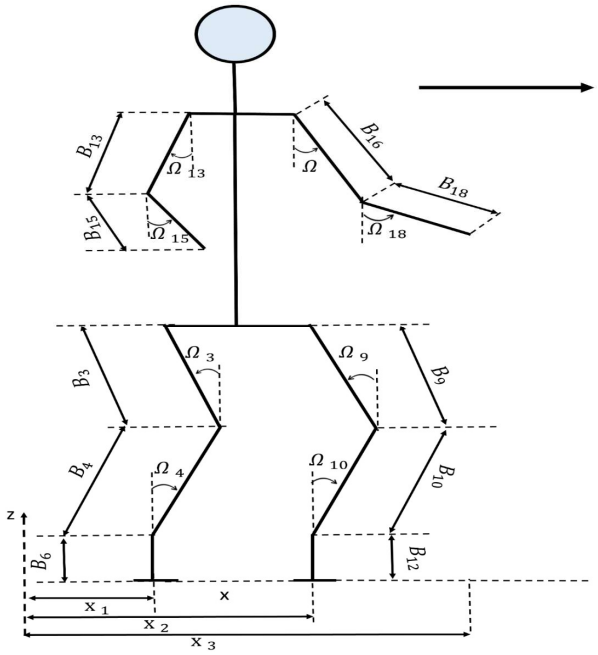

Figure 1. Two-legged robot structure.

where $A_2 = B_9\cos(\Omega_9) + B_{10}\cos(\Omega_{10})$, $L_2 = B_9\sin(\Omega_9) + B_{10}\sin(\Omega_{10})$,
$\beta = \Omega_{10} - \Omega_9 = \cos^{-1}\left(\frac{{A_2}^2+{L_2}^2-{B_{10}}^2-{B_9}^2}{2B_{10}B_9}\right)$. While $\Omega_9 = \Omega_{10} - \beta$,

Now, the ankle joint angle is assumed to equal to zero concerning the vertical axis, thus $\Omega_6 = \Omega_{12} = 0$. Next, to determine the joint angles of the frontal plane leg, the following equation can be considered:

$$\Omega_2 = \Omega_8 = tan^{-1}\left(\frac{f_w}{A_1}\right) \tag{3}$$

$$\Omega_5 = \Omega_{11} = tan^{-1}\left(\frac{0.5f_w}{A_2}\right) \tag{4}$$

where (Ω_2 and Ω_5) are the joint angles of the swing leg in the frontal plane, while (Ω_8 and Ω_{11}) are the joint angles of the standing leg in the frontal plane, f_w represent the width of the TLR foot.

As mentioned before, the Lagrange-Euler formulation has been used for obtaining the dynamics of TLR. Thus, the torque based FOPID controllers for each joint of the TLR are designed while the robot moves in different terrains, as shown in the following equation [23]:

$$\mathcal{T}_i = \sum_{j=1}^{n} R_{ij}(q)\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} U_{ijk}\dot{q}_j\dot{q}_k + V_i \tag{5}$$

For $i, j, k = 1, 2, \dots, n$.
where $\mathcal{T}_i$ denote the torque required for joint i, q represent the movement in (m), $\dot{q}_j$ denote the joint of the velocity in (m/sec), while $\ddot{q}_j$ denote the joint acceleration in (m/sec^2), and the R_{ij} term denotes the inertia term and given as follows:

$$R_{ij} = \sum_{z=max(i,j)}^{n} T_r\left[d_{zj} I_z d_{zi}^T\right] \quad i,j,k = 1 \dots n \tag{6}$$

While the U_{ijk} term defined as the coriolis/centrifugal coefficient and given as follows:

$$U_{ij} = \sum_{z=max(i,j,k)}^{n} T_r\left[\frac{\partial(d_{zk})}{\partial q_z} I_z d_{zi}^T\right] \quad i,j,k = 1 \dots n \tag{7}$$

The V_i term represents the gravity term and given as follows:

$$V_i = -\sum_{z=i}^{n} m_z g d_{zi}{}^{z}\bar{e} r_z \quad i,j,k = 1 \dots n \tag{8}$$

where m_z denote the mass center of the z^{th} link in (m), while ${}^{z}\bar{e} r_z$ and I_z denotes the moment of inertia tensor in $(Kg.m/sec^2)$, and the term g represent the acceleration due to gravity in (m/sec^2).
Now, the acceleration of the links plays an enormous role in controlling each joint of the TLR. Thus, by rearranging Eq. (5) above, the acceleration equation of the links can be obtained as follows [23]:

$$\ddot{q}_j = \sum_{j=1}^{n} R_{ij}(q)^{-1}\left[-\sum_{j=1}^{n}\sum_{k=1}^{n} U_{ijk}\dot{q}_j\dot{q}_k - V_i\right] + \left(\sum_{j=1}^{n} R_{ij}(q)^{-1} \times \mathcal{T}_i\right) \quad i,j,k = 1,2,\dots,n \tag{9}$$

Next, let us consider the following term:

$$\sum_{j=1}^{n} R_{ij}(q)^{-1} \times \mathcal{T}_i = \hat{\mathcal{T}} \tag{10}$$

Eq. (10) can be written as follows:

$$\ddot{q}_j = \sum_{j=1}^{n} R_{ij}(q)^{-1}\left[-\sum_{j=1}^{n}\sum_{k=1}^{n} U_{ijk}\dot{q}_j\dot{q}_k - V_i\right] + \hat{\mathcal{T}} \quad i,j,k = 1,2,\dots,n \tag{11}$$

3 Design of the FOPID controller

During the last years, many researchers across the glope proposed a new thoughts to improve typical controllers. Such that, the authors in [24] provide a nonlinear PID controller to reduce the error to minimum value to increase the system stability for different models. Thus, we proposed the same technique for the neural-fuzzy network input with the FOPID controller [25, 26] to present the best results for the TLR and increase the system stability.

The following expression represents this technique [24]:

$$K_{e(\Omega_i)} = K_0 + K_1\left\{\frac{2}{1+\exp(-k_2 e(\Omega_i)} - 1\right\} \tag{12}$$

Figure 2 below show the general design for Eq. (12) above when $K_0 = 1$, $K_1 = 1$, and $K_2 = 2$.

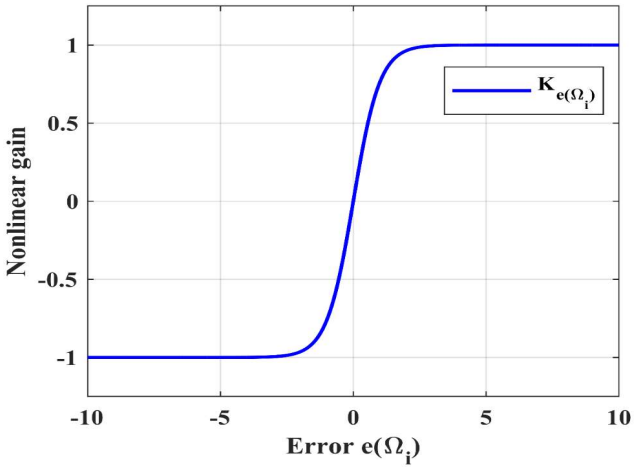

Figure 2. Nonlinear gain function.

Now, the actual torque of FOPID controller required at the different joints of TLR can be explained as the following equation [27-29]:

$$u(\Omega_i) = K_p \acute{e}(\Omega_i) + K_i \frac{d^{-\lambda}\left(\acute{e}(\Omega_i)\right)}{dt^{-\lambda}} + K_d \frac{d^{\mu}\left(\acute{e}(\Omega_i)\right)}{dt^{\mu}} \tag{13}$$

where $\acute{e}(\Omega_i) = e(\Omega_i) \times K_{e(\Omega_i)}$, while K_p is the proportional gain of the FOPID controller, K_i defined as the integral gain of the FOPID controller, K_d defined as the derivative gain of the FOPID, λ and μ represent the proportional and derivative orders, respectively.

The meaning of $e(\Omega_i)$ equal the following formula:

$$e(\Omega_i) = \Omega_{if} - \Omega_{is} \tag{14}$$

where Ω_{is} represent the initial angular position, while Ω_{if} represent the final angular position for the moving joints.

Now, for the fuzzy inference system (FIS), the multi inputs single-output (MISO) involve a set of fuzzy rules, where the rules are given in the following form:

$Rule\ i: IF\ \xi_1(\Omega_i)\ is\ T_1^i\ AND\ \ldots AND\ \xi_n(\Omega_i)\ is\ T_n^i\ THEN\ \delta\left(\xi(\Omega_i)\right) = \delta_i(\Omega_i)$,

where $\xi_j(\Omega_i)$ defined as the linguistic variable, $\delta\left(\xi(\Omega_i)\right)$ defined as the output of the fuzzy inference system, $\delta_i(\Omega_i)$ is the output of the ith rule, while T_j^i defined as the jth linguistic variable of the fuzzy term.

Thus, the final output of the FIS is

$$\delta\left(\xi(\Omega_i)\right) = \sum_{i=1}^{P} w_i\left(\xi(\Omega_i)\right)\delta_i(\Omega_i) \tag{15}$$

where P is the total number of fuzzy rules, $w_i\left(\xi(\Omega_i)\right)$ is the weight of the ith rule.

Now, the fuzzy FOPID controller can be expressed by the following equation:

$$Rule\ i: IF\ \acute{e}(\Omega_i)\ is\ T_1^i\ AND\ \frac{d^{-\lambda}\left(\acute{e}(\Omega_i)\right)}{dt^{-\lambda}}\ is\ T_2^i\ AND\ \frac{d^{\mu}\left(\acute{e}(\Omega_i)\right)}{dt^{\mu}}\ is\ T_3^i\ THEN\ u\left(\xi(\Omega_i)\right) = u_i(\Omega_i)$$

The final output of the fuzzy FOPID controller can be explained as follows:

$$u\big(\xi(\Omega_i)\big) = \sum_{i=1}^{P} w_i\big(\xi(\Omega_i)\big)u_i(\Omega_i) \tag{16}$$

Figure 3 below represent the fuzzy-FOPID design.

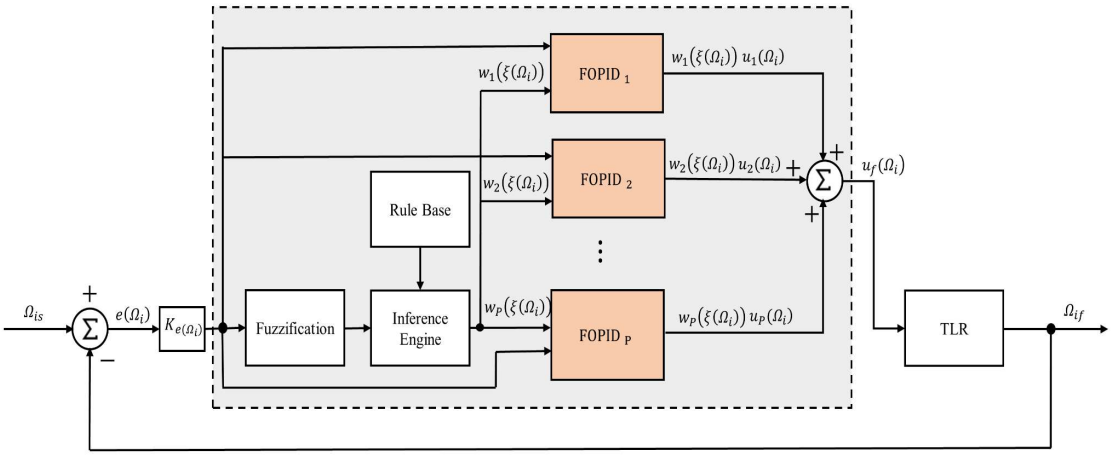

Figure 3. Fuzzy with FOPID network design.

4 Adaptive neuro-fuzzy FOPID network

This section discusses the ANFFOPID controller design work based on ISMA. As shown in Figure 4, the adaptive neuro-fuzzy FOPID network consists of the neuro-fuzzy network, which is shown in the yellow block, while the neuro FOPID network is observed in the orange block. Both of these networks are depicted in Figure 5.

The states of the TLR model $\lambda_i(\Omega_i)$ are the input to the neuro-fuzzy network, which will generate the fuzzy states λ_n and merge the information of the FIS in the current state. The neuro FOPID network will take the fuzzy states λ_n and generates the active gain values $Q(\Omega_i)$ which is then multiplied by the factor G to generate the required control signal $u(\Omega_i)$. The neural network weights are adaptively trained using ISMA.

The structure of the neuro-fuzzy network is firstly predefined by the fuzzy system, where the input layer consists of twelve variables, which are selected to present the state of the TLR system and expressed as $\lambda_i = \xi_i$ where $\xi_i = \big[e(\Omega_i)K_{e(\Omega_i)}\big]$, for $(i = 1,2,\dots,12)$. As depicted in Figure 5, the inputs of the neuro-fuzzy network are expressed as $\lambda_n = [\lambda_1, \cdots, \lambda_{12}]$. The input linguistic layer is connected to the input nodes layer, the inputs nodes represent the fuzzy terms of each linguistic variable.

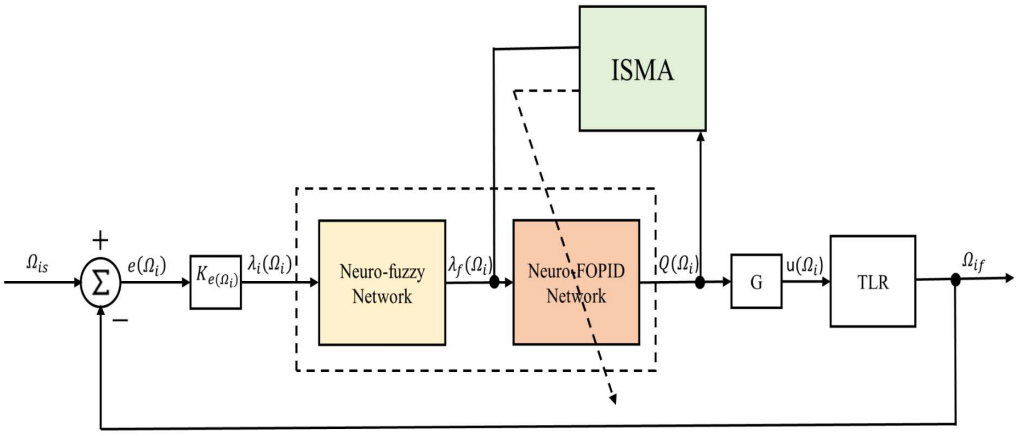

Figure 4. System controller design.

The number of fuzzy terms defines the inputs nodes, each linguistic variable has three fuzzy terms for each of the λ_i that are SMALL, MEDIUM, and LARGE. The membership function of λ_i are shown in Figure 6 below, where S-shaped membership functions are used for the fuzzy terms of λ_i. Then, the rule node numbers can be found by multiplying the fuzzy terms (3) by neural network inputs λ_i. Thus, the number of fuzzy rules nodes are $R_n = 3 \times 12 = 36$. Each rule node presents a corresponding weight, where this weight related to the joint angles are in the form of

$$w_i(\lambda_n) = \frac{\prod_{j=1}^{12} \mu_{M_j^i}(\lambda_n)}{\sum_{k=1}^{R_n} \left(\prod_{j=1}^{12} \mu_{M_j^i}(\lambda_n) \right)} \tag{17}$$

where $i = 1,2, \dots, R_n$, where R_n is the number of fuzz rules related to λ_i. The fuzzy states are given as $\lambda_f = \left[w_1\lambda_1, w_1\lambda_2, w_1\lambda_3, \cdots, w_1\lambda_{12}, \cdots, w_{R_n}\lambda_1, w_{R_n}\lambda_2, w_{R_n}\lambda_3, \cdots, w_{R_n}\lambda_{12} \right]$.

In the following layer, the number of output nodes of the neuro-fuzzy network can be expressed by multiplying the corresponding states by fuzzy rules numbers. Thus the number of the out nodes in this layer is equal to $O_n = 12 \times 36 = 432$.
Now, for the neuro FOPID network, the input layer is the same as the output neuro-fuzzy network layer. As depicted in Figure 5, the output of the first hidden layer is expressed as follows:

$$h_i^1 = \sum_{j=1}^{12} w_{ij}^1 w_i(\lambda_n)\, \lambda_j \quad i = 1,2, \dots, R_n \tag{18}$$

The weights of w_{ij}^1 are obtained and trained using an improved slime mould algorithm (ISMA) which will be discussed in the next section. The output of the second hidden layer is expressed as follows:

$$Q = \sum_{j=1}^{R_n} w_{1j}^2\, h_j^1 \tag{19}$$

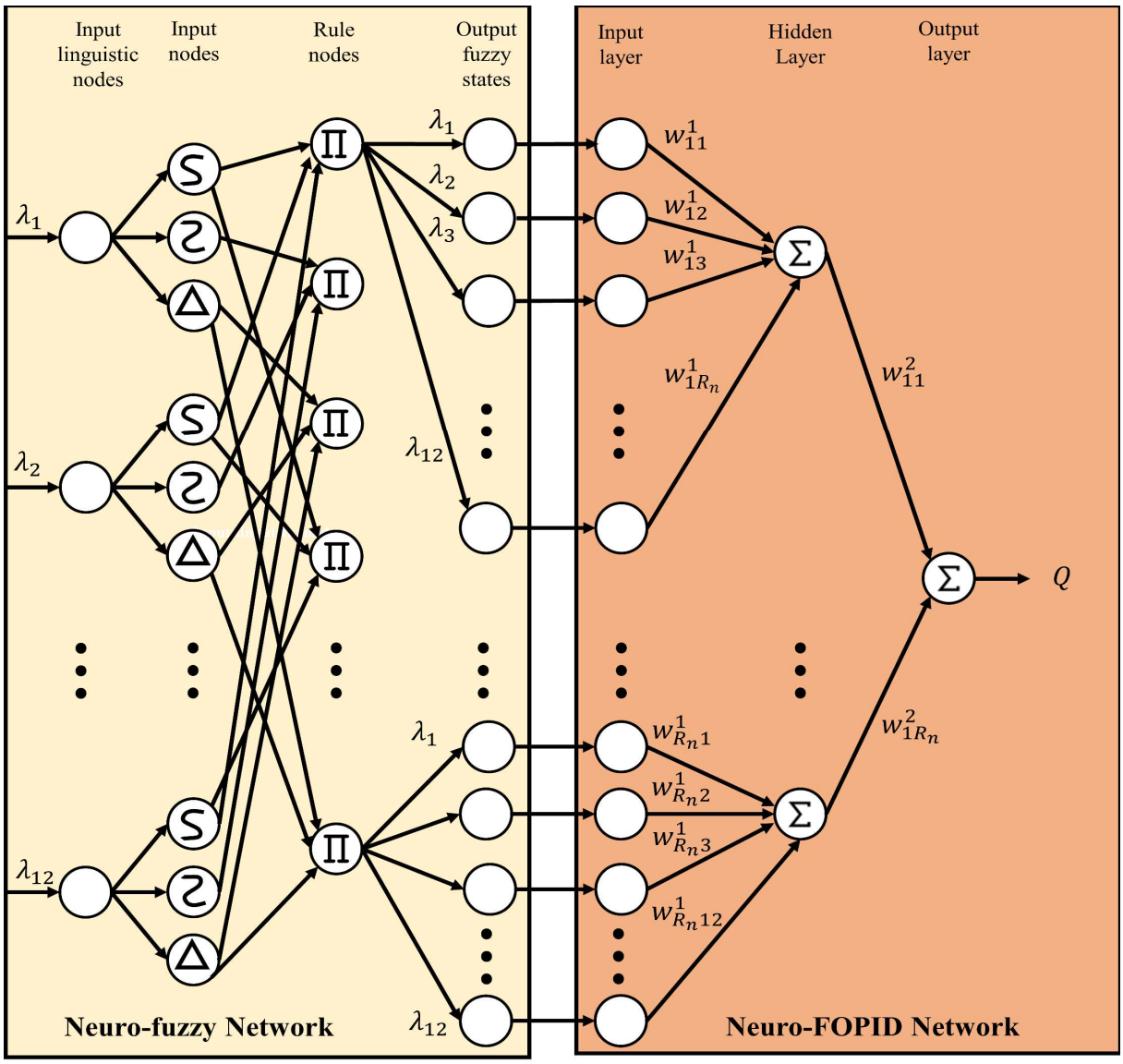

Figure 5. Neuro-fuzzy with FOPID network design.

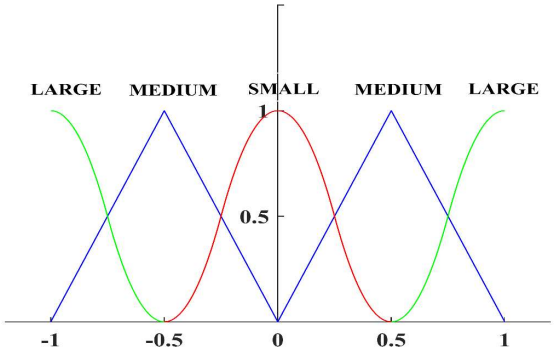

Figure 6. Membership function of λ_n.

The force applied in the TLR can be expressed as:

$$u = GQ \tag{20}$$

where $G = 0.01$ is a constant gain selected after deep experiments.

$$u = G \sum_{k=1}^{R_n} w_{1k}^2 \sum_{j=1}^{12} w_{kj}^1 w_k(\lambda_n)\, \lambda_j \tag{21}$$

$$\begin{aligned} u = G \Bigg(\sum_{k=1}^{R_n} w_k(\lambda_n) \big(& w_{k1}^1 w_{1k}^2 \lambda_1 + w_{k2}^1 w_{1k}^2 \lambda_2 + w_{k3}^1 w_{1k}^2 \lambda_3 + w_{k4}^1 w_{1k}^2 \lambda_4 + w_{k5}^1 w_{1k}^2 \lambda_5 \\ & + w_{k6}^1 w_{1k}^2 \lambda_6 + w_{k7}^1 w_{1k}^2 \lambda_7 + w_{k8}^1 w_{1k}^2 \lambda_8 + w_{k9}^1 w_{1k}^2 \lambda_9 + w_{k10}^1 w_{1k}^2 \lambda_{10} \\ & + w_{k11}^1 w_{1k}^2 \lambda_{11} + w_{k12}^1 w_{1k}^2 \lambda_{12} \big) \Bigg) \end{aligned} \tag{22}$$

$$\begin{aligned} u = \sum_{k=1}^{R_n} w_k(\lambda_n) \big(& K_{P_{1k}} \lambda_1 + K_{D_{1k}} \lambda_2 + K_{I_{1k}} \lambda_3 + K_{P_{2k}} \lambda_4 + K_{D_{2k}} \lambda_5 + K_{I_{2k}} \lambda_6 + K_{P_{3k}} \lambda_7 \\ & + K_{D_{3k}} \lambda_8 + K_{I_{3k}} \lambda_9 + K_{P_{4k}} \lambda_{10} + K_{D_{4k}} \lambda_{11} + K_{I_{4k}} \lambda_{12} \big) \end{aligned} \tag{23}$$

where $K_{P_{1k}} = G w_{k1}^1 w_{1k}^2$, $K_{D_{1k}} = G w_{k2}^1 w_{1k}^2$, $K_{I_{1k}} = G w_{k3}^1 w_{1k}^2$, $K_{P_{2k}} = G w_{k4}^1 w_{1k}^2$, $K_{D_{2k}} = G w_{k5}^1 w_{1k}^2$, $K_{I_{2k}} = G w_{k6}^1 w_{1k}^2$, $K_{P_{3k}} = G w_{k7}^1 w_{1k}^2$, $K_{D_{3k}} = G w_{k8}^1 w_{1k}^2$, $K_{I_{3k}} = G w_{k9}^1 w_{1k}^2$, $K_{P_{4k}} = G w_{k10}^1 w_{1k}^2$, $K_{D_{4k}} = G w_{k11}^1 w_{1k}^2$, $K_{I_{4k}} = G w_{k12}^1 w_{1k}^2$.

Thus, Eq. (23) above can be expressed as follows:

$$u = \sum_{k=1}^{R_n} w_k(\lambda_n) U_{ak} + w_k(\lambda_n) U_{bk} + w_k(\lambda_n) U_{ck} + w_k(\lambda_n) U_{dk} \tag{24}$$

where $U_{ak} = K_{P_{1k}} \lambda_1 + K_{D_{1k}} \lambda_2 + K_{I_{1k}} \lambda_3$, $U_{bk} = K_{P_{2k}} \lambda_4 + K_{D_{2k}} \lambda_5 + K_{I_{2k}} \lambda_6$, $U_{ck} = K_{P_{3k}} \lambda_7 + K_{D_{3k}} \lambda_8 + K_{I_{3k}} \lambda_9$, $U_{dk} = K_{P_{4k}} \lambda_{10} + K_{D_{4k}} \lambda_{11} + K_{I_{4k}} \lambda_{12}$.

Thus, these variables stand for the outputs u_i in the concept of the fuzzy FOPID controller that is given in Eq. (16). As mentioned before, the neuro-fuzzy and the neuro-FOPID are merged to present the adaptive technique, while the wights of the neuro-FOPID are trained using the following improved algorithm in the next section.

5 Slime mould algorithm

Slime Mould Algorithm (SMA) [30-32], an stochastic optimization invented by [33]. The oscillation mode of slime mould inspires the SMA in nature. SMA employs a unique mathematical model that uses adaptive weights to simulate the propagation wave of SMA. The SMA depends on connecting food with excellent exploratory ability and exploitation propensity.
The SMA used the odour of the air to approach the food, where the following formula described the approaching mechanism [33]:

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{X_b(t)} + \overrightarrow{vb}.\left(\overrightarrow{W}.\overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right), r < p \\ \overrightarrow{vc}.\overrightarrow{X(t)}, r \geq p \end{cases} \tag{25}$$

where r is a random number of interval (0,1), while p given as shown in the following formula:

$$p = tanh|S(i) - DF| \tag{26}$$

where $S(i)$ is the fitness of the location of the slime mould ($\vec{X}$), DF defined as the best fitness function of all iteration. t represent the current iteration number, $\overrightarrow{vc}$ is linearly decrease number from [1⇒0], $\overrightarrow{X_A}$ and $\overrightarrow{X_B}$ defined as two individuals selected randomly from slime mould, $\overrightarrow{X_b}$ defined as the individual with the highest odour concertation currently found, $\overrightarrow{W}$ defined as the weight of slime mould, $\overrightarrow{vb}$ defined as parameter with range [-a, a] where a found as follows:

$$a = arctanh(-\left(\frac{t}{max\_t}\right) + 1) \tag{27}$$

where (max $\_t$) is the maximum iteration number for SMA.

Despite the advantages of the SMA algorithm, such as the good speed in conducting a simulation of the robot and obtaining the best parameters for the controller, but after conducting several simulations with the presence of the adaptive neuro-fuzzy controller with the proposed terrain, it was discovered that the robot suffers from a lack of implementation of the proposed path due to the huge amount of information that must be processed from the ANFFOPID controller which employs the SMA algorithm. For this reason, we proposed an improvement to this algorithm that depends on replace one of the functions to increase the ability of the robot to perform accurate calculations in the presence of the proposed controller.

The improvements developed on the SMA are explicitly chosen to be appropriate for the ANFFOPID controller with the URV model, where the first improvement depends on improved Eq. (26), such that ($tanh$) function working on the interval (-1, 1). Thus, instead of this function, the following function is developed for the ISMA:

$$p = \left(\frac{randi([-1000,1000])}{1000} * |S(i) - DF|\right) \tag{28}$$

where ($randi$) refers to a random integer number in the interval (-1000,1000) and divided by (1000) to provide a number of one order; this improvement allows the TLR for a better search of the best weights for the neuro-FOPID controller. To summarize the improved algorithm, the following pseudo-code represents the SMA algorithm procedure.

Algorithm 1. Pseudo-code of the Improved SMA algorithm.

Generate the initial SMA population, $Max_{iteration}$;
Generate the positions of SMA $X_i(i = 1, 2, \ldots, N)$;
While $(t < Max_{iteration})$
  Find the fitness of the SMA;
  Update the best fitness, X_b
  Find the weight W of the SMA;
    for *each search operation*
      Update p by **Eq**. (28);
      Update vc and vb;
      Update the positions using **Eq**. (25);
    end for
    $t \leftarrow t + 1$
end while
return the best fitness, X_b;

6 Results and discussion

In this section, the effectiveness of the ANFFOPID controller is evaluated, but first, the performance of the ISMA is evaluated by comparing this algorithm with other optimization algorithms. These algorithms are particle swarm optimization (PSO) [34], genetic algorithm (GA) [35], modified chaotic inverse weed optimization (MCIWO) [15], and the typical slime mould algorithm (SMA). This comparison depends on operating each optimization algorithm with an ANFFOPID controller to update the weights of the neuro-FOPID network adaptively. Each optimization algorithm is running for an average of (30) complete simulations, where Figure 7 below is obtained, which shows the average cost-value versus the number of iterations for the selected optimization algorithms.

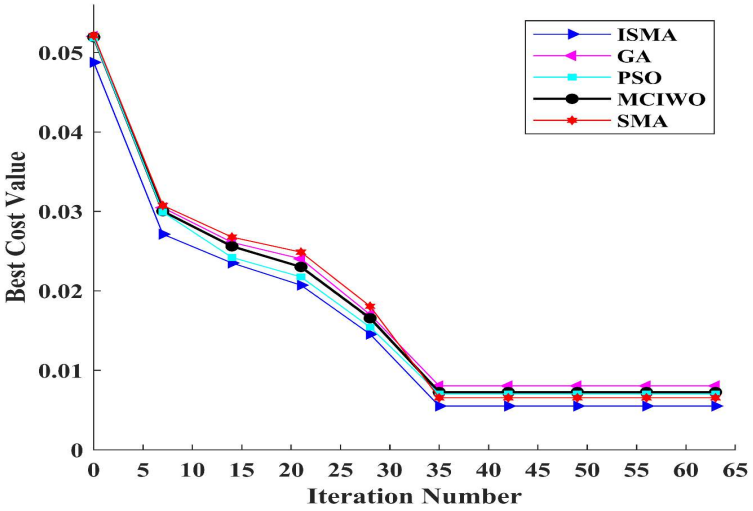

Figure 7. The average of best cost value versus iteration number.

Figure 7 above indicate clearly that the ISMA present the best performance with the minimum cost value compared with other optimization algorithms. Furthermore, to be more sure about the ISMA effectiveness, we set another experiment depending on the distribution of the required torque values for the TLR with the ISMA, PSO, GA, MCIWO, and SMA algorithms, such that the algorithm that presents the least distribution torque will be the most fitted to work with the proposed controller. The required torque represents the actual torque required for the joint motors of the TLR. This experiment has been done using a boxplot technique to describe the collected results for each optimization algorithms, as shown in Figure 8. Each optimization algorithm was also run for (30) complete simulations in order to make this work more accurate.

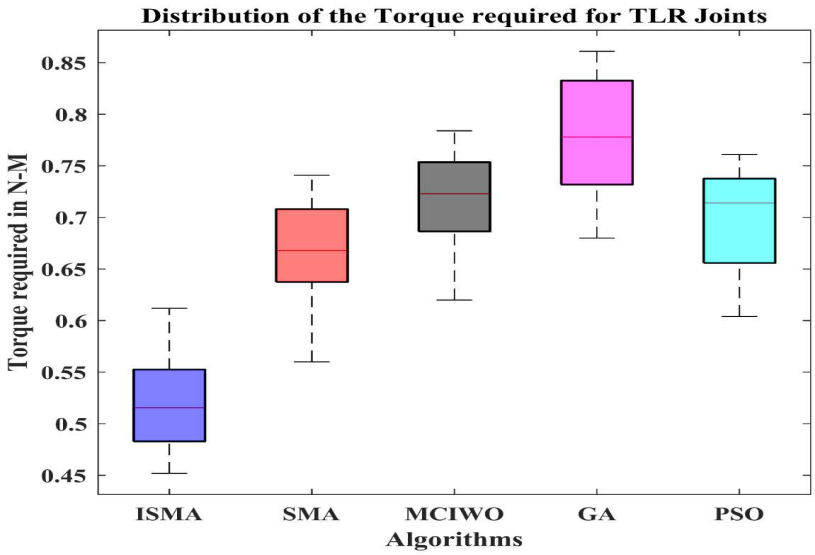

Figure 8. Distribution of the torque required for the joint motors with different optimization algorithms.

Figure 8 indicate clearly that the ISMA present the minimum torque for the joint motors compared to the other optimization algorithms. After ensuring the effectiveness of the ISMA, the error of the displacement angle of the joint motors in TLR are measured to evaluate the performances of the ANFFOPID controller. The ANFFOPID effectiveness is evaluated by comparing the ANFFOPID controller with other existing work, which is adaptive torque based on proportional integral derivative based on modified chaotic invasive weed optimization (MCIWO) with a neural network [15]. The error of the joint angles represents the difference between the desired angle and the actual output joint angle for the TLR. These error magnitudes correspond to the various joints of the swing, and the standing legs of the two-legged robot are shown in Figure 9.

The error is produced for different joints while the TLR moves in: flat surface, ascending, and descending stairs. For all the joints, the error magnitude for the MCIWO-NN controller is very high initially and decreases quietly with time. At the same time, the ANFFOPID controller presents the best performance with the minimum possible error for all of the joints compared to the MCIWO-NN. Moreover, it is clearly shown that the error generated in the flat surface for various joints are less than the other cases that are stairs ascending and descending.

Figure 10 presents the required torque of different joints for the TLR while the robot moves through various terrains. The collected results show that the average torque required has a high value while the TLR ascends the stairs and less value while moving along the flat surface. It is noticed that the torque at joint 3 has the highest torque; this may be because that joint 3 carry all the load while ascending the stairs. At the same time, the results show the effectiveness of the ANFFOPID controller compared to the MCIWO-NN controller, such that the torque required for the ANFFOPID controller has fewer torques for all TLR terrains.

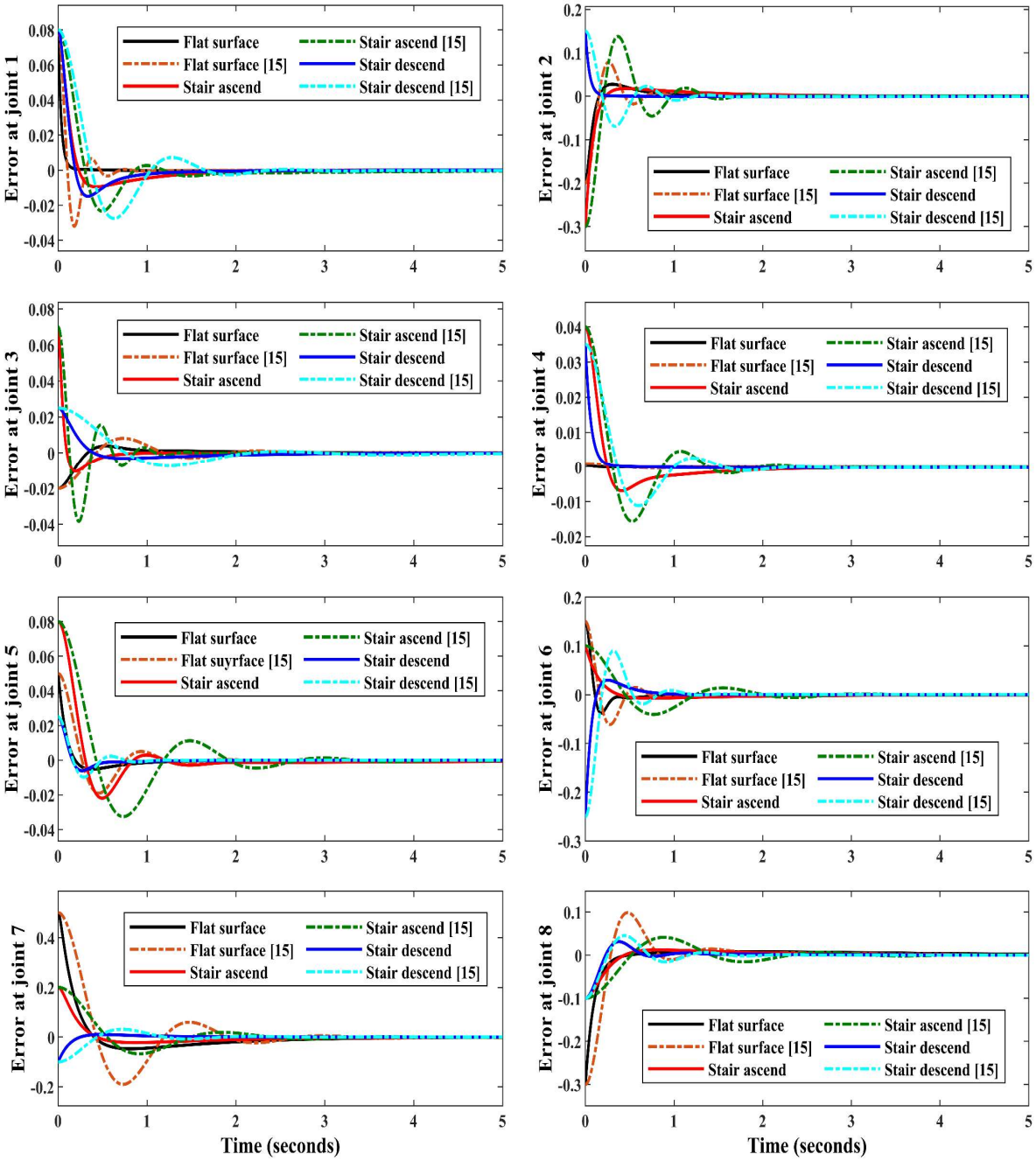

Figure 9. The errors for the various joints.

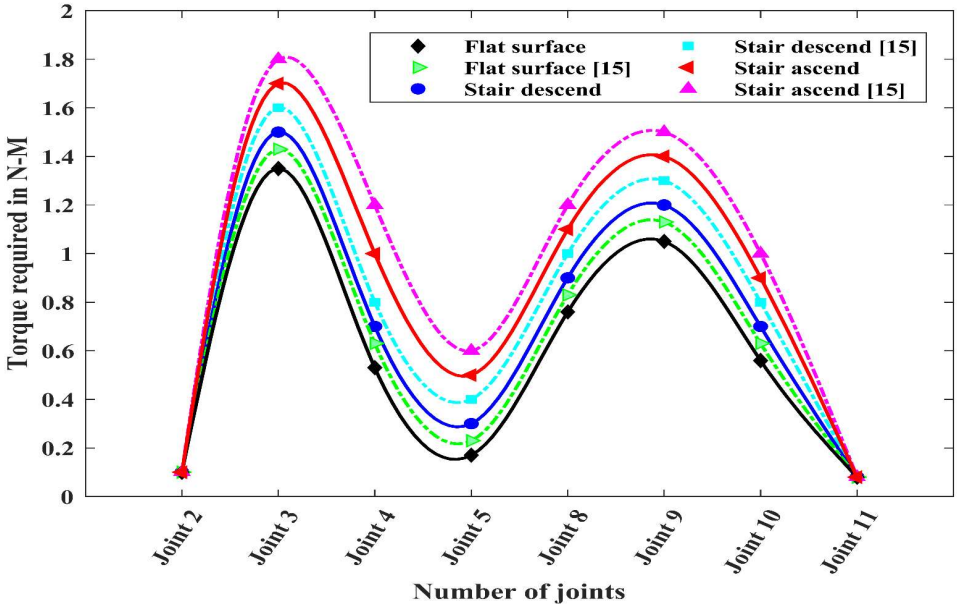

Figure 10. The torque required for various joints.

Now, we present a TLR trajectory for stairs ascending and descending, where Figure 11 show a symbol of the stairs ascending for TLR while the obtained trajectory for the ANFFOPID controller and the MCIWO-NN are shown in Figure 12.

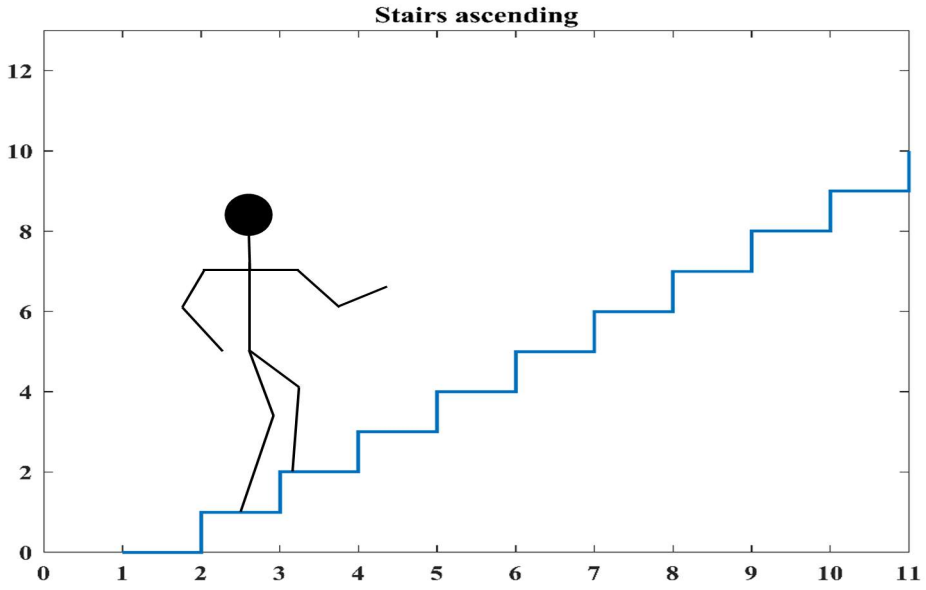

Figure 11. Stairs ascending for a two-legged robot.

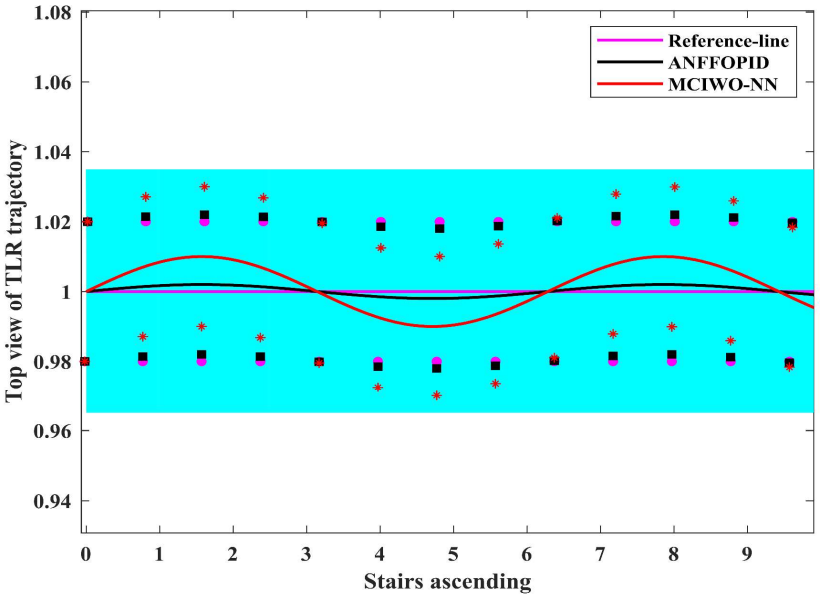

Figure 12. Top-view of TLR line trajectory.

A top view of stairs ascending for the TLR is presented in Figure 12 above, where the width of the stairs is given in a cyan color. Moreover, a middle line is presented in the stairs and considered a reference line trajectory for the TLR model. Each of the trajectories obtained for the ANFFOPID controller and MCIWO-NN has markers which represent the foot position of the TLR for the trajectory obtained. The results obtained indicate clearly that the ANFFOPID controller presents the best foot position with trajectory tracking compared to the MCIWO-NN, which present the worst trajectory tracking compared to the reference and ANFFOPID markers. The foot position of the TLR represents the balance of the robot around a certain trajectory. The concept used to measure the foot balance of the TLR is called zero moment point (ZMP) [22], [36].

The same procedure is used for the stairs descending, where Figure 13 present the stairs descending for TLR while the obtained trajectories for the ANFFOPID controller and the MCIWO-NN are shown in Figure 14. The results show that the ANFFOPID markers present the best foot position with trajectory tracking than the MCIWO-NN. Finally, we present another trajectory for the flat surface terrain, such that the two-legged robot has a start point and ending point and the same concept used for ascending and descending the stairs is used for the flat surface. Figure 15 show the results obtained for the TLR while moving on a flat surface, where the ANFFOPID present the best trajectory in addition to robot balance performance compared to the MCIWO-NN.

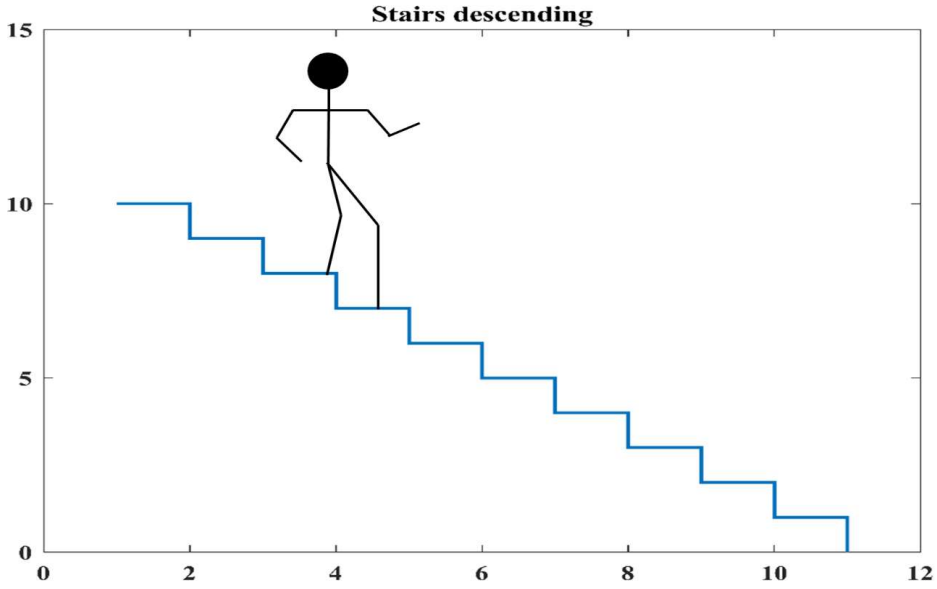

Figure 13. Stairs descending for a two-legged robot.

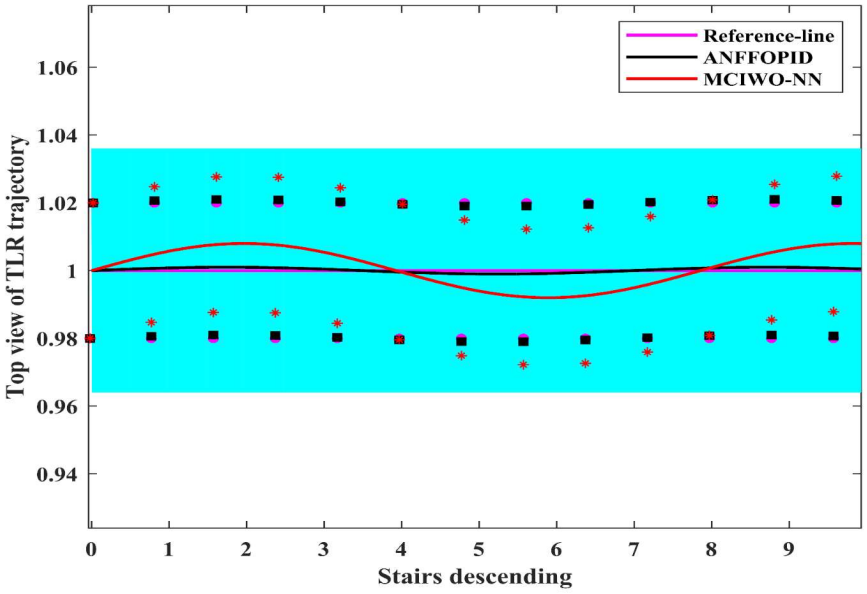

Figure 14. Top-view of TLR line trajectory.

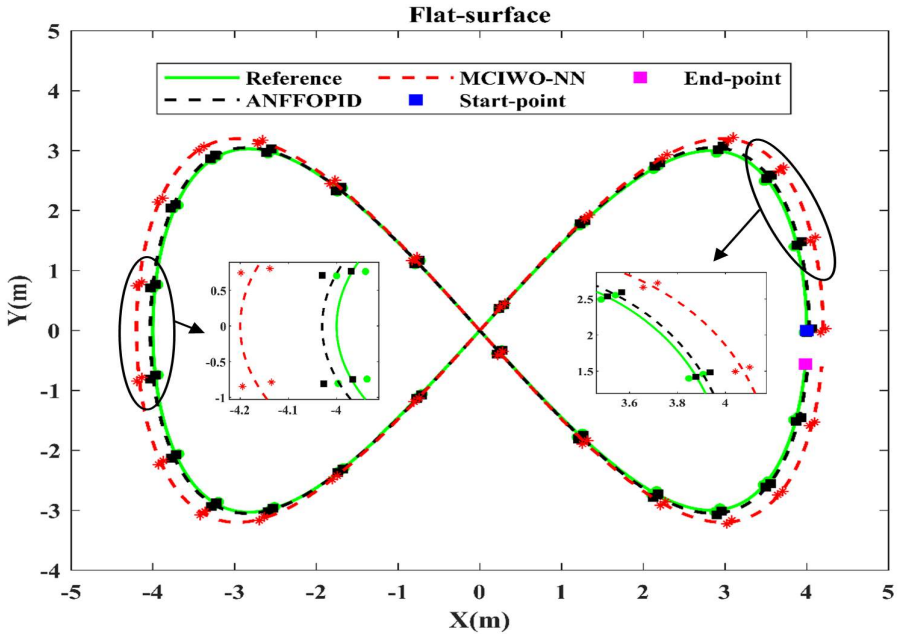

Figure 15. Flat-surface line tracking.

Finally, a zero moment point concept has been used, such that the position of the ZMP is lying at the center of the foot support polygon for the flat surface case, ascending case, and descending case. Thus, the TLR is considered to be dynimcally balance for all of the cases. Figure 16 indicate that the TLR for the flat surface movment is noticed to be more balanced since the location of the ZMP is closer to the center of the robot foot than the ascending and descending cases. Such that, the position of ZMP is kind of away from the foot support polygon center. For this reason, the slope of the ascending and descending cases tend to be more different than the flat surface case.

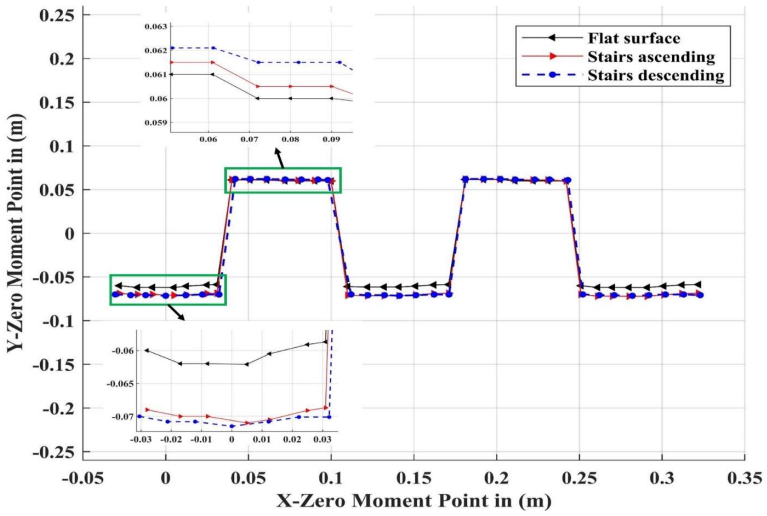

Figure 16. The variation of Zero Moment Point in the directions of X and Y axis.

7 Conclusion

In this work, we proposed an innovative adaptive fuzzy-neural FOPID network design based on an improved slime mould optimization algorithm for a two-legged robot moving along a flat surface, stairs ascending, and stairs descending. The results obtained demonstrate clearly that the ANFFOPID controller can solve the highly nonlinear effects of the two-legged robot and reduce the error fast to zero while the TLR moves on different terrains. In addition, the required torque for the two-legged robot is measured and clearly shows that the ANFFOPID presents the best performance to consume less torque for the TLR joints than the MCIWO-NN, which consume more torque for different joints. The neuro-fuzzy FOPID network has been used, such that the fuzzy network used to supply the neuro-FOPID network with weights based on TLR states, the neuro-FOPID network used to provide the FOPID controller with the best gain values K_P, K_D, and K_I, while the neuro-FOPID weights are trained using an improved slime mould algorithm adaptively. Future works suggested to study the TLR balancing techniques across other complex terrains, different controllers may be used such as adaptive sliding mode controllers, nonlinear proportional integral controllers and etc.

References

[1] K. J. Waldron, "Force and Motion Management in Legged Locomotion," *IEEE J. Robot. Autom.*, vol. 2, no. 4, pp. 214–220, 1986, doi: 10.1109/JRA.1986.1087060.

[2] K. Erbatur, A. Okazaki, K. Obiya, T. Takahashi, and A. Kawamura, "A study on the zero moment point measurement for biped walking robots," *Int. Work. Adv. Motion Control. AMC*, pp. 431–436, 2002, doi: 10.1109/amc.2002.1026959.

[3] C. Boussema, M. J. Powell, G. Bledt, A. J. Ijspeert, P. M. Wensing, and S. Kim, "Online gait transitions and disturbance recovery for legged robots via the feasible impulse set," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1611–1618, 2019, doi: 10.1109/LRA.2019.2896723.

[4] X. Zang, Y. Liu, X. Liu, and J. Zhao, "Design and control of a pneumatic musculoskeletal biped robot," *Technol. Heal. Care*, vol. 24, pp. S443–S454, 2016, doi: 10.3233/THC-161167.

[5] M. F. Silva and J. A. T. MacHado, "Fractional order PDα joint control of legged robots," *JVC/Journal Vib. Control*, vol. 12, no. 12, pp. 1483–1501, 2006, doi: 10.1177/1077546306070608.

[6] E. Najafi, G. A. D. Lopes, and R. Babuška, "Balancing a legged robot using state-dependent riccati equation control," *IFAC Proc. Vol.*, vol. 19, pp. 2177–2182, 2014, doi: 10.3182/20140824-6-za-1003.01724.

[7] A. V. Gulalkari, P. S. Pratama, G. Hoang, D. H. Kim, B. H. Jun, and S. B. Kim, "Object tracking and following six-legged robot system using Kinect camera based on Kalman filter and backstepping controller," *J. Mech. Sci. Technol.*, vol. 29, no. 12, pp. 5425–5436, 2015, doi: 10.1007/s12206-015-1144-4.

[8] C. Sun, W. He, W. Ge, and C. Chang, "Adaptive Neural Network Control of Biped Robots," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 2, pp. 315–326, 2017, doi: 10.1109/TSMC.2016.2557223.

[9] J. Li *et al.*, "Neural Approximation-based Model Predictive Tracking Control of Non-holonomic Wheel-legged Robots," *Int. J. Control. Autom. Syst.*, vol. 19, no. 1, pp. 372–381, 2021, doi: 10.1007/s12555-019-0927-2.

[10] D. K. Pratihar, K. Deb, and A. Ghosh, "Optimal path and gait generations simultaneously of a six-legged robot using a GA-fuzzy approach," *Rob. Auton. Syst.*, vol. 41, no. 1, pp. 1–20, 2002, doi: 10.1016/S0921-8890(02)00273-7.

[11] M. Zabihi and A. Alasty, "Modeling and Fuzzy Control of One-legged Somersaulting Robot," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 2701–2706, 2018, doi: 10.1109/IROS.2018.8593897.

[12] M. B. B. Sharifian, H. Afsharirad, and S. Galvani, "A particle swarm optimization approach for optimum design of PID controller in linear elevator," *2010 9th Int. Power Energy Conf. IPEC 2010*, vol. 19, no. 2, pp. 451–455, 2010, doi: 10.1109/IPECON.2010.5697038.

[13] R. J. Yahya and N. H. Abbas, "Optimal integral sliding mode controller controller design for 2-RLFJ manipulator based on hybrid optimization algorithm," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 1, pp. 293–302, 2022, doi: 10.11591/ijece.v12i1.pp293-302.

[14] S. Liang, B. Song, and D. Xue, "Landing route planning method for micro drones based on hybrid optimization algorithm," *Biomim. Intell. Robot.*, vol. 1, no. May, p. 100003, 2021, doi: 10.1016/j.birob.2021.100003.

[15] R. K. Mandava and P. R. Vundavilli, "Design and development of an adaptive-torque-based proportional-integral-derivative controller for a two-legged robot," *Soft Comput.*, vol. 25, no. 16, pp. 10953–10968, 2021, doi: 10.1007/s00500-021-05811-4.

[16] D. Izci, S. Ekinci, H. L. Zeynelgil, and J. Hedley, "Performance evaluation of a novel improved slime mould algorithm for direct current motor and automatic voltage regulator systems," *Trans. Inst. Meas. Control*, vol. 44, no. 2, pp. 435–456, 2022.

[17] D. Izci, S. Ekinci, E. Eker, and A. Dündar, "Assessment of Slime Mould Algorithm Based Real PID Plus Second-order Derivative Controller for Magnetic Levitation System," in *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2021, pp. 6–10.

[18] P. R. Vundavilli, S. K. Sahu, and D. K. Pratihar, "Dynamically balanced ascending and descending gaits of a two-legged robot," *Int. J. Humanoid Robot.*, vol. 4, no. 4, pp. 717–751, 2007, doi: 10.1142/S0219843607001266.

[19] P. R. Vundavilli and D. K. Pratihar, "Balanced gait generations of a two-legged robot on sloping surface," *Sadhana - Acad. Proc. Eng. Sci.*, vol. 36, no. 4, pp. 525–550, 2011, doi: 10.1007/s12046-011-0031-7.

[20] P. R. Vundavilli, S. K. Sahu, and D. K. Pratihar, "AND DESCENDING GAIT GENERATIONS OF A," vol. 4, no. 4, pp. 777–814, 2007.

[21] B. Aghbali, A. Yousefi-Koma, A. Ghasemi Toudeshki, and A. Shahrokhshahi, "ZMP trajectory control of a humanoid robot using different controllers based on an offline trajectory generation," *Int. Conf. Robot. Mechatronics, ICRoM 2013*, pp. 530–534, 2013, doi: 10.1109/ICRoM.2013.6510161.

[22] J. J. Alcaraz-Jiménez, D. Herrero-Pérez, and H. Martínez-Barberá, "A simple feedback controller to reduce angular momentum in ZMP-based gaits," *Int. J. Adv. Robot. Syst.*, vol. 10, pp. 1–7, 2013, doi: 10.5772/52448.

[23] R. K. Mandava and P. R. Vundavilli, "Near Optimal PID Controllers for the Biped Robot While Walking on Uneven Terrains," *Int. J. Autom. Comput.*, vol. 15, no. 6, pp. 689–706, 2018, doi: 10.1007/s11633-018-1121-3.

[24] H. Seraji, "A new class of nonlinear PID controllers with robotic applications," *J. Robot. Syst.*, vol. 15, no. 3, pp. 161–181, 1998, doi: 10.1002/(SICI)1097-4563(199803)15:3<161::AID-ROB4>3.0.CO;2-O.

[25] D. Izci, S. Ekinci, E. Eker, and M. Kayri, "A Novel Modified Opposition-based Hunger Games Search Algorithm to Design FOPID Controller for Magnetic Ball Suspension System," *Adv. Control Appl. Eng. Ind. Syst.*, p. e96.

[26] D. Izci and S. Ekinci, "An Efficient FOPID Controller Design for Vehicle Cruise Control System Using HHO Algorithm," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2021, pp. 1–5.

[27] M. W. Hasan and N. H. Abbas, "An improved swarm intelligence algorithms-based nonlinear fractional order-PID controller for a trajectory tracking of underwater vehicles," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 18, no. 6, p. 3173, 2020, doi: 10.12928/telkomnika.v18i6.16282.

[28] A. Demirören, S. Ekinci, B. Hekimoğlu, and D. Izci, "Opposition-based artificial electric field algorithm and its application to FOPID controller design for unstable magnetic ball suspension system," *Eng. Sci. Technol. an Int. J.*, vol. 24, no. 2, pp. 469–479, 2021.

[29] S. Ekinci, D. Izci, and B. Hekimoğlu, "Optimal FOPID speed control of DC motor via opposition-based hybrid manta ray foraging optimization and simulated annealing algorithm," *Arab. J. Sci. Eng.*, vol. 46, no. 2, pp. 1395–1409, 2021.

[30] D. İzci, S. Ekinci, and S. Ekinci, "Comparative performance analysis of slime mould algorithm for efficient design of proportional–integral–derivative controller," *Electrica*, vol. 21, no. 1, pp. 151–159, 2021.

[31] S. Ekinci, D. Izci, H. L. Zeynelgil, and S. Orenc, "An application of slime mould algorithm for optimizing parameters of power system stabilizer," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2020, pp. 1–5.

[32] D. Izci, "An enhanced slime mould algorithm for function optimization," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2021, pp. 1–5.

[33] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for

stochastic optimization," *Futur. Gener. Comput. Syst.*, vol. 111, pp. 300–323, 2020, doi: 10.1016/j.future.2020.03.055.

[34] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemom. Intell. Lab. Syst.*, vol. 149, pp. 153–165, 2015, doi: 10.1016/j.chemolab.2015.08.020.

[35] A. Jayachitra and R. Vinodha, "Genetic Algorithm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor," *Adv. Artif. Intell.*, vol. 2014, pp. 1–8, 2014, doi: 10.1155/2014/791230.

[36] H. M. Wu and C. L. Hwang, "Trajectory-based control under ZMP constraint for the 3D biped walking via fuzzy control," *IEEE Int. Conf. Fuzzy Syst.*, pp. 706–712, 2011, doi: 10.1109/FUZZY.2011.6007507.