



# An adaptive PID control algorithm for the two-legged robot walking on a slope

Ravi Kumar Mandava<sup>1,2</sup> · Pandu R. Vundavilli<sup>1</sup>

Received: 16 January 2018 / Accepted: 28 June 2019 / Published online: 4 July 2019  
© Springer-Verlag London Ltd., part of Springer Nature 2019

## Abstract

The crux in designing the PID controller lies in determining its gain values, which play a major role in deciding its performance. The gains are fed as inputs to the controller and are to be decided before its run. On the other hand, the effectiveness of the biped walk purely depends on the performance of the PID controller. Initially, the upper and lower body gaits of the two-legged robot are determined using the concept of inverse kinematics. Further, the dynamics of the biped robot is derived by using Lagrange–Euler formulation. The main objective of the present research is to decide the gains of the torque-based PID controller with the help of a neural network trained by using nature-inspired optimization algorithms, namely MCIWO and PSO. The adaptiveness of the algorithm lies in modifying the gains of the controller based on the magnitude of the error in the angular displacement received at the input to the NN. Once the controller is developed, its effectiveness is tested in computer simulations. Finally, the optimum controlled gait angles obtained by the best approach are tested on a real biped robot.

**Keywords** Adaptive control · Sloping surface · NN · Torque-based PID controller

## 1 Introduction

In recent years, researchers are paying more attention in the areas of research related to bipedal robots due to its usability and adaptiveness in accomplishing various tasks. Controlling the locomotion of the biped robot is difficult to achieve in unknown terrains. Several studies [1–6] focused on the control of locomotion of the biped robot after utilizing the concept of zero moment point (ZMP). Most of the research work was focused on the analysis of walking stability of the biped robots on flat and uneven terrains. To maintain the stability, the trajectory-based methods are suitable, as the robots perform their walk based on

predesigned trajectories. However, if the trajectories are fixed and the terrain conditions are varying, the robot may fail to move on those terrains. It is important to note that the polynomial interpolation methods may become incompetent, if the complexity of walking increases. Further, the increase in the order of the polynomial leads to enhanced computational time. To overcome this problem, Shih [1] developed a strategy that uses a cubic spline interpolation to estimate the foot trajectories. In [2], Huang et al. formulated the constraints for the foot trajectories to produce various types of trajectories on different terrain conditions. Further, Park et al. [5] proposed a trajectory generation method based on the combination of third-order polynomial and sinusoidal functions to realize the free gait for the biped walking. Human walking does not show the characteristics of precise trajectory tracking. But, the biological researches show that the human beings walk is a consequence and combination of inherent patterns and reflexes. Later on, researchers worked on the bio-inspired control methods based on the biomechanics of musculoskeletal system and the motors of neural networks [7, 8]. Inspired by the concept of ZMP, many researchers had developed stable walking for biped robots. Several studies

---

✉ Ravi Kumar Mandava  
rm19@iitbbs.ac.in

Pandu R. Vundavilli  
pandu@iitbbs.ac.in

<sup>1</sup> School of Mechanical Sciences, IIT Bhubaneswar, Bhubaneswar, Odisha 752050, India

<sup>2</sup> Mechanical Engineering Department, Vignan's Foundation for Science, Technology and Research, Guntur, Andhra Pradesh 522213, India

were based on the equations of ZMP that express the relationship between the ZMP and centre of gravity (COG). In addition to the COG trajectory, the generation of ZMP equation is useful for the biped robot to walk on flat ground. The ZMP equation with constant COG height on flat ground was very simple. Vukobratovic and Stepanenko [9] proposed the concept of ZMP, which corresponds to the centre of pressure of floor reaction force on the sole of a biped robot. If the ZMP falls inside the foot support polygon, the sole of the foot does not leave the ground and the balanced walk is possible. The foot rotation indicator (FRI) is the extension of the ZMP, and was proposed by Goswami [10]. While the ZMP exists within the boundary of the foot support polygon, the FRI can be used as an indicator of the extent of balance of the robot. However, many researchers had proposed trajectory-planning methods for the balanced walking of biped robots. Kajita et al. [11] generated the walking pattern for the biped robot by the preview control of the ZMP. Here, the COG trajectory is generated by using the cart-table model. Further, Shibuya et al. [12] developed a linear pendulum model (LPM) and applied this to control the COG trajectory generation in the double-support phase. In addition to the above approaches, Erbaturo and Kurt [13] proposed a reference generation algorithm based on the linear inverted pendulum model (LIPM) and moved the ZMP references.

Later on, some of the researchers had dealt with the walking of the biped mechanism on a slope surface to enhance the utility of the biped robot. Ching-Long and Chien-Jung [14] studied the motion control of statically stable biped robot with seven degrees of freedom (DOF) on an uneven floor. To maintain the statically stable walk, the COG of the biped had to fall inside the foot support polygon during the single- and double-support phases. Sung et al. [15] proposed a trajectory generation method for a biped robot to walk on an inclined surface. To generate the walking trajectory, three additional assumptions were used. First, the centre of mass was moved parallel to the slope surface. Second, the ZMP should be fixed with the supporting foot during the single-support phase. Third, when the slope of the surface is changed, the boundary conditions of the robot while walking are also changed. Cristiano et al. [16] proposed the concept of central pattern generator (CPG) for efficient locomotion control of the biped robots while ascending and descending the slope surface. The feedback signals were generated by the robot, i.e. the data from inertial and force sensors were directly fed to the CPG, and allowed the robot to automatically change the locomotion pattern over the flat and inclined terrains in real time. Further, the researchers not only worked on the generation of the dynamically balanced gaits of the biped robot using ZMP, COG, LPM and LIPM but also solved the problems related to the controlling of the generated gait of the biped robot using PID controllers. Tae et al. [17] discussed a hybrid self-

organizing fuzzy (SOF) PID controller to tune the parameters of the multi-input multi-output nonlinear biped robot. They observed a better performance for hybrid SOF-PID controller when compared to SOF-PID controller. Safa et al. [18] developed a predictive PID controller for stable walking of the biped robot. The predictive PID controller was used to imitate the calculated time and reduce the complexity of the control. Moreover, Ho Pham et al. [19] proposed a novel adaptive neural PID (AN-PID) controller, which was suitable for the real-time control of the biped robot. The proposed AN-PID controller had a simpler and self-organising structure, fast online-tuning speed and flexibility in updating the online data. The above algorithm was capable of optimizing the gains and weights of MLPNN model integrated with AN-PID controller.

The PID controller is an electrical element, whose purpose is to reduce the error between a desired and actual process variable. In order to do so, it requires tuning of some of its essential parameters. But, tuning of the PID controller is a time-consuming process. Recently, some researchers are trying with different types of stochastic methods, population-based algorithms, nature-inspired and computational intelligence algorithms, such as artificial neural network, fuzzy systems and artificial immune systems [20], to tune the PID controller and train the biped robot [21]. In [22], the authors had constructed a modified triangular fuzzy PID controller and used a genetic algorithm (GA) to fine-tune the membership function parameters of fuzzy system, and to improve the quality of the system in terms of rise time, overshoot and steady-state error. Hang et al. [23] developed an intelligent tuning algorithm, i.e. bacterial foraging optimization (BFO) technique, to tune the PID controller. The BFO algorithm uses natural selection to eliminate the entities with poor foraging strategies for locating, capturing and ingesting the food. Further, Huseyin and Zafer [24] proposed a new nature-inspired optimization (i.e. ant algorithm) algorithm to tune the parameters of the PID controller. The results of the tuning process were compared with Ziegler–Nichols (ZN), iterative feedback tuning (IFT) and internal model control (IMC) methods. Dusan et al. [25] tuned the parameters of the PID controller using various optimization algorithms such as, differential evolution algorithm, genetic algorithm, bat algorithm, hybrid bat algorithm, particle swarm optimization algorithm and cuckoo search algorithm. The results were compared using statistical analysis and observed that PSO had exhibited better performance when compared with the other methods.

In the present research, the authors have implemented a torque-based PID controller to control the joint motors in a systematic manner. It is important to mention that ZMP-based PID controller, which was used by many of the researchers [26–29], is an indirect method of controlling the motor, as the error signal between the obtained and

reference ZMP trajectory is used to design the gaits for various links of the biped robot. In this process, the controller may suggest some dynamically balanced postures which may not be kinematically feasible. Moreover, finding the most appropriate gait of the biped robot that fulfils the sequence-oriented criterion and repeatability conditions is difficult to achieve, whereas torque-based PID controller considers the error signal of each joint of the biped robot and tries to design the gains of the controller by minimizing the error between two consecutive gait angles of each joint of the biped robot. The gains of the PID controller are obtained by using the neural network (NN) (i.e. adaptive based). Further, the structure of NN is tuned with the help of a nature-inspired optimization algorithm, i.e. MCIWO (which is a variant of standard IWO algorithm). To improve the performance of the algorithm, in the present research the authors have added two new variables, namely cosine and chaotic variables. The cosine variable is used to increase the search space, and the chaotic variable is used to push the solution directly into global optimum position. Finally, the performance of the developed algorithm has been compared with another nature-inspired algorithm such as PSO. Further, the gait angles generated by the best algorithm are fed into the real robot for testing its performance.

The rest of the paper is organised as follows. Section 2 describes the kinematics, dynamics and design of PID controller of the biped robot. Section 3 discusses the structure of neural network. Section 4 explains the nature-inspired optimization algorithms which are used to tune the weights of the NN controller. Section 5 deals with the experiments and the results related to the MCIWO-NN- and PSO-NN-based PID controller algorithms. Further, the paper concludes with a summary of work and the suggestions for further improvement.

## 2 Modelling and design of controller of the biped robot

In this section, the model of the biped robot and the design of the corresponding PID controller are discussed.

### 2.1 Modelling

Day by day, the usage of biped robots in both the industrial and non-industrial applications is increasing enormously. An overview of the biped robot used in this research is shown in Fig. 1. All the joints of the robot are connected in a serial manner, and each joint has only one motor mounted on it. The robot is designed to have 6 DOF in each leg and 3 DOF in each hand.

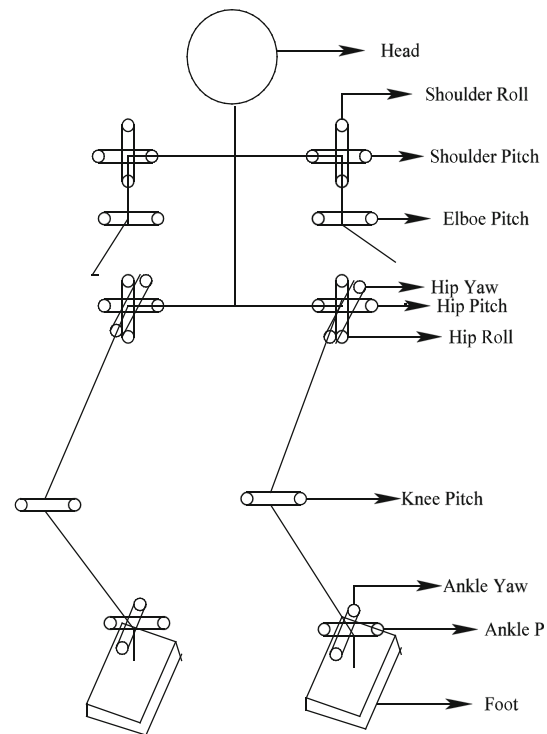


Fig. 1 Schematic diagram showing structure of the Robot

To maintain the balanced walk, the foot of the biped robot is assumed to follow a particular polynomial trajectory. In this research, the authors had conducted a study on different polynomial trajectories, such as quadratic, cubic and fifth-order polynomials, and found that the cubic polynomial trajectory has produced dynamically more balanced gaits when compared with the other trajectories. The maximum angle of slope on which the robot can perform walk without slipping is decided by the value of coefficient friction ( $\mu$ ). To tackle the situation in an appropriate manner, all the joints are allowed to move in a coordinated manner. For moving the links in a systematic way without violating the kinematic constraints, the concept of inverse kinematics is applied. While walking, the foot of the robot is always kept parallel to the ground and the ZMP should fall within the foot support polygon. The equation for calculating the ZMP in X- and Y-directions is given by Eqs. (1) and (2), respectively.

$$x_{ZMP} = \frac{\sum_{i=1}^n (I_i \dot{\omega}_i - m_i \ddot{x}_i z_i - m_i x_i (g + \ddot{z}_i))}{\sum_{i=1}^n (m_i (\ddot{z}_i + g))} \quad (1)$$

$$y_{ZMP} = \frac{\sum_{i=1}^n (I_i \dot{\omega}_i - m_i \ddot{y}_i z_i - m_i y_i (g + \ddot{z}_i))}{\sum_{i=1}^n (m_i (\ddot{z}_i + g))} \quad (2)$$

where  $\dot{\omega}_i$ ,  $I_i$ ,  $g$  and  $m_i$  represent the angular acceleration ( $\text{rad/s}^2$ ), mass moment of inertia ( $\text{kg m}^2$ ) of the  $i$ th link, acceleration due to gravity ( $\text{m/s}^2$ ) and mass ( $\text{kg}$ ) of the link  $i$ , respectively. Further,  $(x_i, y_i, z_i)$  signify the coordinates of

the  $i$ th lumped mass and  $\ddot{z}_i$  and  $\ddot{x}_i$  indicate the acceleration of the  $i$ th link moving in  $z$ - and  $x$ -directions ( $m/s^2$ ), respectively.

Once the ZMP is calculated, the dynamic balance margin of the biped robot in  $X$ - and  $Y$ -directions has been determined by using the mathematical Eqs. (3) and (4), respectively.

$$x_{DBM} = \left( \frac{f_s}{2} \times \cos \alpha - |x_{ZMP}| \right) \tag{3}$$

$$y_{DBM} = \left( \frac{f_w}{2} \times \cos \alpha - |y_{ZMP}| \right) \tag{4}$$

where  $f_w$  and  $f_s$  represent the width and length of the foot, respectively.

The gait generation methodology for the biped while descending the slope is very much similar to the ascending case with only one difference, that is, the acceleration due to gravity “ $g$ ” is acting in the direction opposite to that of the movement of the root.

### 2.2 Controller

Once the model of the robot is generated after generating the balanced gaits on ascending and descending the sloping surfaces, the torque-based PID controllers are designed to control each joint of the biped robot in a coordinated manner. The torque required to move each joint of the biped robot is derived based on the Lagrange–Euler formulation (ref. Eq. 5).

$$\tau_{i,the} = \sum_{j=1}^n M_{ij}(q)\ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n C_{ijk}\dot{q}_j\dot{q}_k + G_i \tag{5}$$

$i, j, k = 1, 2, \dots, \dots, n$

Further, the acceleration of the links plays an important role to control each joint of the biped robot. The expression that represents the acceleration of the link is obtained by rearranging the terms of the above equation and is given as follows.

$$\ddot{q}_j = \sum_{j=1}^n M_{ij}(q)^{-1} \left[ - \sum_{j=1}^n \sum_{k=1}^n C_{ijk}\dot{q}_j\dot{q}_k - G_i \right] + \left( \sum_{j=1}^n M_{ij}(q)^{-1} * \tau_{i,the} \right) \tag{6}$$

$i, j, k = 1, 2, \dots, \dots, n$

The term  $\hat{\tau} = \sum_{j=1}^n M_{ij}(q)^{-1} * \tau_{i,the}$  \tag{7}

After substituting Eq. (7) in Eq. (6), it can be written as

$$\ddot{q}_j = \sum_{j=1}^n M_{ij}(q)^{-1} \left[ - \sum_{j=1}^n \sum_{k=1}^n C_{ijk}\dot{q}_j\dot{q}_k - G_i \right] + \hat{\tau} \tag{8}$$

$i, j, k = 1, 2, \dots, \dots, n$

where the term  $\tau_{i,the}$  represents the theoretical torque required at each joint (N m) and  $q_j$ ,  $\dot{q}_j$  and  $\ddot{q}_j$  indicate the displacement in (rad), velocity in (rad/s) and acceleration in ( $rad/s^2$ ) of the joint, respectively. The actual torque required at various joints of the biped robot is calculated using the following expression.

$$\tau_{act} = K_p e(\theta_i) + K_d e(\dot{\theta}_i) + K_i \int e(\theta_i) dt \tag{9}$$

$$e(\theta_i) = \theta_{if} - \theta_{is} \tag{10}$$

where  $\tau_{act}$  represents the actual torque required for each joint,  $K_p$ ,  $K_d$  and  $K_i$  denote the proportional, derivative and integral gains of the controllers, respectively. Further,  $\theta_{is}$  and  $\theta_{if}$  are the initial and final angular positions. Further, the integral term in Eq. (9) needs to be substituted by its state variables, namely  $\dot{x}_i$  and its meaning is given as follows.

$$x_i = \int e(\theta_{is}) dt \Rightarrow \dot{x}_i = \theta_{if} - \theta_{is} \quad i = 1, 2, \dots, \dots, n \tag{11}$$

Now, substituting Eqs. (10 and 11) in Eq. (9), it can be written as

$$\tau_{i,act} = K_{pi}(\theta_{if} - \theta_{is}) - K_{di}\dot{\theta}_{is} + K_{ii}\dot{x}_i \quad i = 1, 2, \dots, \dots, n \tag{12}$$

The above equation represents the torque-based PID controller equation used to reduce the error between the initial and final angular positions of each joint. The equation that represents the acceleration of the link, i.e. after substituting the actual torque expression, is given as follows.

$$\ddot{q}_j = \sum_{j=1}^n M_{ij}(q)^{-1} \left[ - \sum_{j=1}^n \sum_{k=1}^n C_{ijk}\dot{q}_j\dot{q}_k - G_i \right] + K_{pi}(\theta_{if} - \theta_{is}) - K_{di}\dot{\theta}_{is} + K_{ii}\dot{x}_i \tag{13}$$

### 3 Neural network-based PID controller

The working principle of neural network is similar to that of the human brain in the following two ways: firstly, the network attains the knowledge through learning, and secondly, the data are stored within the synaptic weights. The network consists of junctions which are connected with simple processing units to process the data. The structure of the network (that is, shown in Fig. 2) consists of three

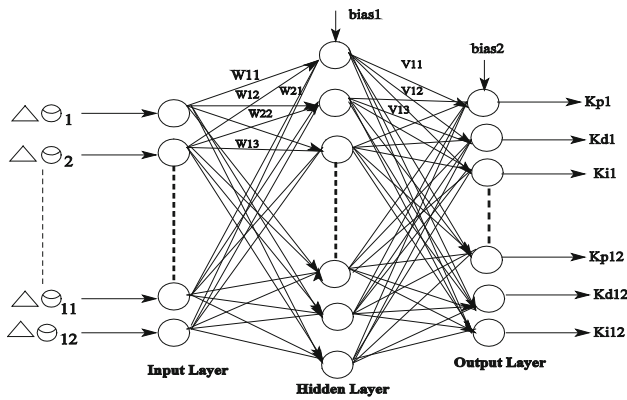


Fig. 2 Structure of the neural network

layers, that is, input, hidden and output layers, and each layer consists of number of neurons. The weighted sum of the inputs and the bias values are added to each neuron, and the values are passed through transfer functions [30]. The transfer functions, namely linear, log-sigmoid and tan-sigmoid functions, are assigned to the input, hidden and output layers, respectively.

In the present research work, the structure of the network consists of 12 neurons in the input layer that indicates the angular error at various joints of the biped robot at various instances of time. The output layer consists of 36 neurons that represent the proportional, integral and derivative gains of the PID controller for the 12 joints of the lower links of the biped robot. These gains are useful to control each joint of the biped robot. Further, the number of neurons in the hidden layer is decided with the help of a parametric study. It is important to note that the performance of neural network depends on the connecting weights between the input–hidden and hidden–output layers. During this study, the connecting weights, bias value of the network and the coefficients of transfer function values of individual layers are optimized with the help of the two nature-inspired metaheuristic algorithms MCIWO and PSO

(that is, shown in Fig. 3). Moreover, the performance of the network highly depends on the number of neurons in its hidden layer. The NN-parameters, such as weights between input to hidden [ $v$ ], hidden to output [ $w$ ] and bias values [ $b$ ], are generated as one string of the MCIWO and PSO algorithms. The length of the MCIWO/PSO string always depends on the topology of the NN. The MCIWO/PSO string can be represented as follows:

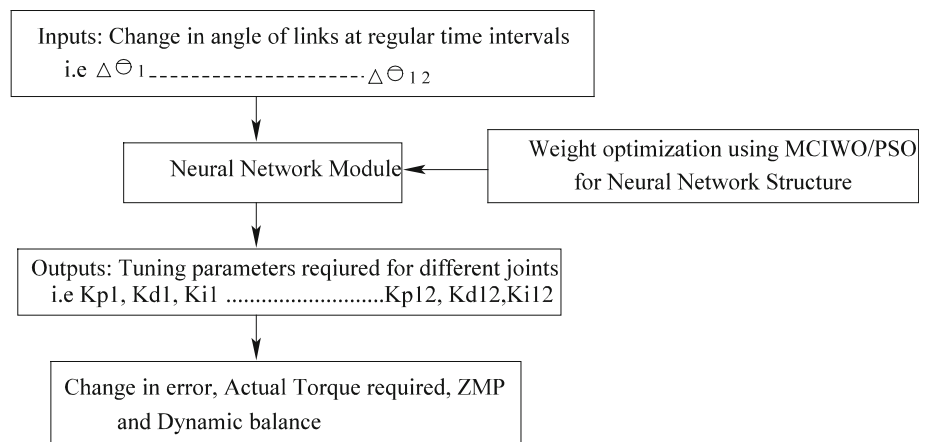
$$\underbrace{0 \dots 1}_{v_{11}} \underbrace{1 \dots 1}_{v_{12}} \underbrace{1 \dots 0}_{v_{13}} \underbrace{0 \dots 1}_{v_{21}} \underbrace{1 \dots 0}_{v_{22}} \underbrace{0 \dots 0}_{v_{23}} \\
 \dots \dots \dots \underbrace{0 \dots 1}_{w_{11}} \underbrace{1 \dots 1}_{w_{12}} \underbrace{0 \dots 1}_{b_1} \underbrace{0 \dots 1}_{b_2}$$

The number of candidate NNs are considered as population of MCIWO/PSO strings, and the MCIWO/PSO will try to find the best one through search. To train the neural network, batch-mode training has been implemented. Initially, a set of data has been used for training the network with the help of evolutionary algorithms. The difference ( $\Delta\theta_k = \theta_j - \theta_i$ ) between the initial and final joint angles of all the 12 joints,  $\Delta\theta_1, \Delta\theta_2, \Delta\theta_3, \Delta\theta_4, \Delta\theta_5, \Delta\theta_6, \Delta\theta_7, \Delta\theta_8, \Delta\theta_9, \Delta\theta_{10}, \Delta\theta_{11}, \Delta\theta_{12}$ , is considered as inputs to the neural network. The adaptive gains of the PID controllers are predicted at the output of the network. It is called adaptive because the gains are varied for every instant of time based on the amount of error in angular displacement received at the input nodes.

Once the gains of the PID controllers are obtained, they are used to tune the PID controller. The variation in the RMS deviation of the angular displacement at the end of each interval ( $\infty_{ijkf}$ ) and the beginning of the interval ( $\infty_{ijki}$ ) is considered as the fitness ( $f$ ) of each population and is given as follows.

$$f = \frac{1}{d} \sum_{i=1}^d \left[ \frac{1}{h} \sum_{j=1}^b \sqrt{\frac{1}{2} \sum_{k=1}^p (\alpha_{ijkf} - \alpha_{ijks})^2} \right]$$

Fig. 3 Flow chart related to the MCIWO-NN and PSO-NN approaches



where  $d$ ,  $h$  and  $p$  are the number of training scenarios, number of intervals considered in one step and number of joints, respectively, for which the controllers are designed.

### 4 Optimization algorithms used

In the present research work, the authors have used two nature-inspired optimization algorithms, namely a newly proposed MCIWO and a well-established PSO to train the structure of the neural network. The block diagram showing the proposed adaptive PID control algorithm is given in Fig. 4. The explanation related to the optimization algorithms is presented in the subsequent sub-sections.

#### 4.1 MCIWO algorithm

The MCIWO algorithm is developed after making a modification to the standard invasive weed optimization (IWO) algorithm, which is a nature-inspired stochastic optimization algorithm proposed by Mehrabain and Lucas [31] in 2006.

The standard procedure used by the IWO algorithm is explained as follows:

1. Initially, a number of seeds are distributed randomly over the search space,  $S_i = (x_1, x_2, \dots, x_n)$ , where  $n$  is the number of selected variables, over the search space. Subsequently, each seed occupies random values for each variable in  $N$ -dimensional space.
2. The fitness of each individual seed is calculated after incorporating them into the optimization problem. The fittest seeds grow to flowering weeds and are able to produce new seeds.
3. Each individual weed is ranked based on its fitness value with respect to the subsequent weeds. Consequently, each weed produces new seeds based on its rank in the colony. The weeds which have attained more resources in the field have a better chance of producing more seeds. The number of seeds to be produced by each weed varies linearly from  $N_{\min}$  to

$N_{\max}$  which can be calculated using the following equation.

$$\text{Number of seeds} = \frac{F_i - F_{\text{worst}}}{F_{\text{best}} - F_{\text{worst}} + N_{\min}} (N_{\max} - N_{\min})$$

where  $F_i$  denotes the fitness of the  $i$ th weed.  $F_{\text{best}}$  and  $F_{\text{worst}}$  represent the best and worst fitness in the weed colony. This mechanism ensures that every weed considered in the generation takes part in the reproduction process.

4. Once the seeds are generated, they are distributed normally over the search space with mean equal to zero and varying standard deviation  $\sigma_{\text{iter}}$ , which is described by

$$\sigma_{\text{iter}} = \left( \frac{\text{iter}_{\max} - \text{iter}}{\text{iter}_{\max} - 1} \right)^m (\sigma_0 - \sigma_f) + \sigma_f$$

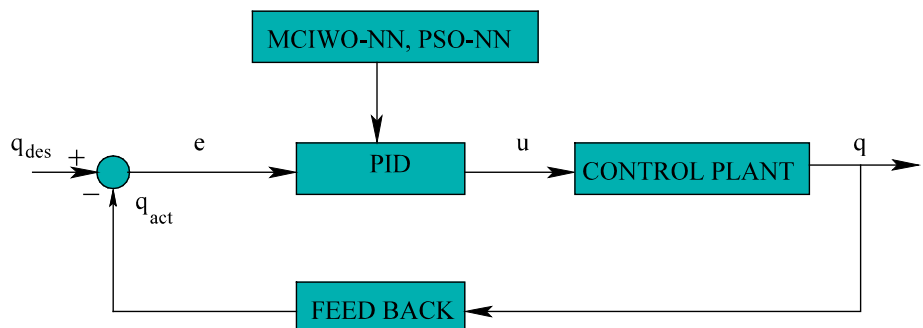
where  $\text{iter}$  and  $\text{iter}_{\max}$  represent the number of iterations and maximum number of iterations, respectively,  $\sigma_0$  and  $\sigma_f$  indicate the initial and final standard deviations, respectively, and  $m$  denotes the nonlinear modulation index. The modulation index is a critical parameter which can influence the convergence performance of the IWO algorithm.

5. Further, the fitness of each seed is calculated along with their parent weed and the whole population is ranked. Those weeds with less fitness are eliminated from the colony through competition to make the

**Table 1** Parameters related to the biped robot

Link	Mass (kg)	Inertia (kg m <sup>2</sup> )	Length (m)
Lower limb of the leg	0.1190	0.00007440	0.093
Upper limb of the leg	0.0700	0.00012600	0.093
Ankle to foot	0.2460	0.00003300	0.033
Upper arm	0.1930	0.00008569	0.060
Lower arm	0.0592	0.00012000	0.060
Trunk	0.0975	0.00017700	0.122
Pelvis	0.1940	0.00671000	0.037

**Fig. 4** Block diagram showing the proposed adaptive PID controller architecture



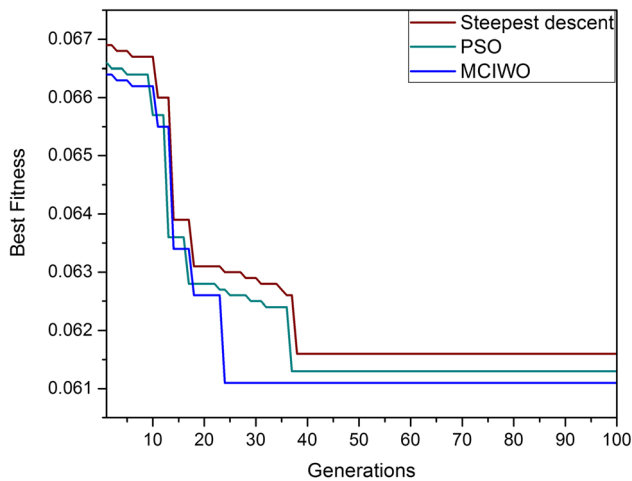


Fig. 5 Graph showing the convergence of the fitness with different optimization algorithms

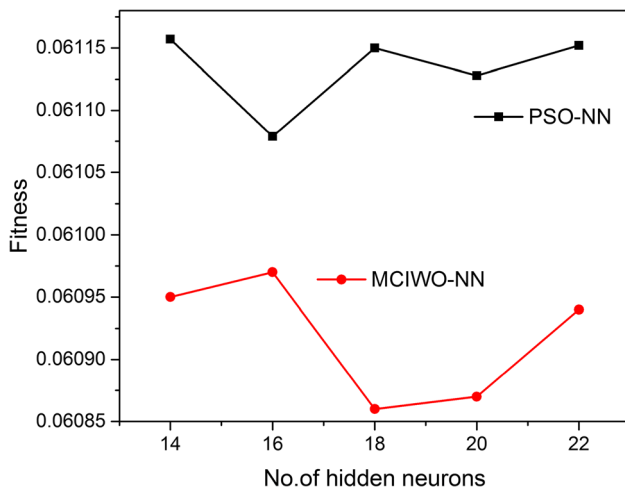


Fig. 6 Graph showing the fitness value by varying the hidden neurons with different adaptive algorithms

number of weeds equal to the maximum weeds allowed in the field.

Table 2 Parametric settings of the ascending and descending the slope algorithm

MCIWO			PSO		
Parameter	Setting value		Parameter	Setting value	
	Ascending	Descending		Ascending	Descending
$\sigma_{initial}$	4%	3%	$C_1$	1.5	1.5
$\sigma_{final}$	0.000001	0.000001	$C_2$	2.0	2.0
$n$	3	4	$w$	1.0	1.0
$S_{min}$	1	1	popsize	50	70
$S_{max}$	6	8			
$n_{pop_{initial}}$	5	5			
$n_{pop_{final}}$	8	10			

6. The above process is repeated from step 2 until the maximum number of iterations allowed by the user is reached.

To enhance certain capabilities of IWO, such as improving the pre-mature convergence and exploring the better solution quickly by preventing the spread of the new solution in the search space, cosine and chaotic variables are introduced during the dispersal stage. In the present study, Chebyshev map is used to generate the random number used in the chaotic term and is given as follows.

$$X_{k+1} = \cos(k \cos^{-1}(X_k))$$

Further, the cosine term-added spatial dispersal scheme of the present study is given as follows.

$$\sigma_{iter} = \left( \frac{iter_{max} - iter}{iter_{max} - 1} \right)^m * |\cos(iter)| * (\sigma_0 - \sigma_f) + \sigma_f$$

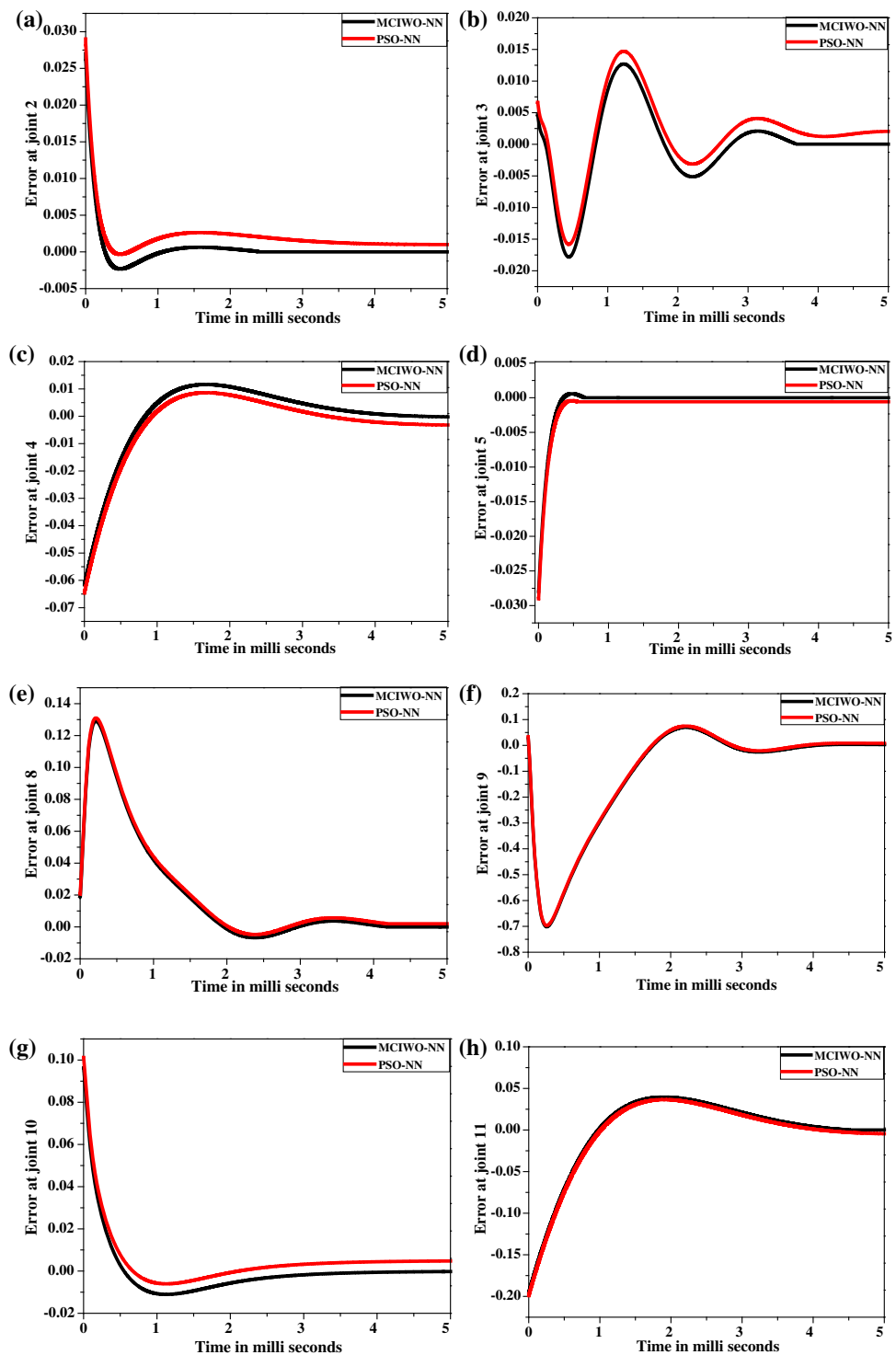
Algorithm 1: MCIWO algorithm

- 1: Set the maximum and minimum value for each variable
- 2: Generate the random populations in N-dimensional space
- 3: For iter=1: maximum number of generations
- 4: Update the standard deviation after adding the ‘cos’ term
- 5: Calculate the maximum and minimum fitness in the colony
- 6: Generate the new pioneer seeds and disperse them using chaotic variable
- 7: Calculate the fitness of new seeds
- 8: Once the population is reached maximum ( $p_{max}$ )
- 9: Sort the population in descending order with their fitness, lower fitness weeds are eliminated
- 10: Go to next iteration

4.2 PSO algorithm

PSO is a swarm-based optimization algorithm developed by Kennedy and Eberhart [32]. In PSO, each particle represents a candidate solution to the problem and the implementation of the algorithm is relatively easy. Initially, the position of the particle  $i$  is represented by  $X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD})$ , which denotes a potential solution to the

**Fig. 7** Error at various joints of the biped robot while ascending the slope surface. **a** Joint 2, **b** Joint 3, **c** Joint 4, **d** Joint 5, **e** Joint 8, **f** Joint 9, **g** Joint 10 and **h** Joint 11



problem in  $D$ -dimensional space. Each particle is having a memory to store its previous best position  $P_{\text{best}}$ , and it is calculated based on its velocity along each dimension, denoted as  $V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD})$ .

Once the initial swarm is generated, the fitness value of the problem is denoted in terms of the average error in angular position for all the joints of the biped robot, and

will be calculated for all the swarms. The particle with best fitness is designated as  $P_{\text{best}}$ . Further, the  $G_{\text{best}}$  is calculated from the available local best swarm. After calculating the two best positions, the velocity and the new position of the particles in the swarm can be calculated by using the following two equations.



$$V_{i,D}^{t+1} = W \times V_{i,D}^t + C_1 \times \text{rand}_1 \times (P_{i,D}^t - X_{i,D}^t) + C_2 \times \text{rand}_2 \times (G_{i,D}^t - X_{i,D}^t)$$

$$X_{i,D}^{t+1} = X_{i,D}^t + V_{i,D}^{t+1} \quad i = 1, 2, \dots, n$$

where  $W$  is the inertia weight,  $C_1$  and  $C_2$  determine the relative influence of the social and cognition components (learning factors), while  $\text{rand}_1$  and  $\text{rand}_2$  denote the two random numbers uniformly distributed in the interval  $[0, 1]$ .

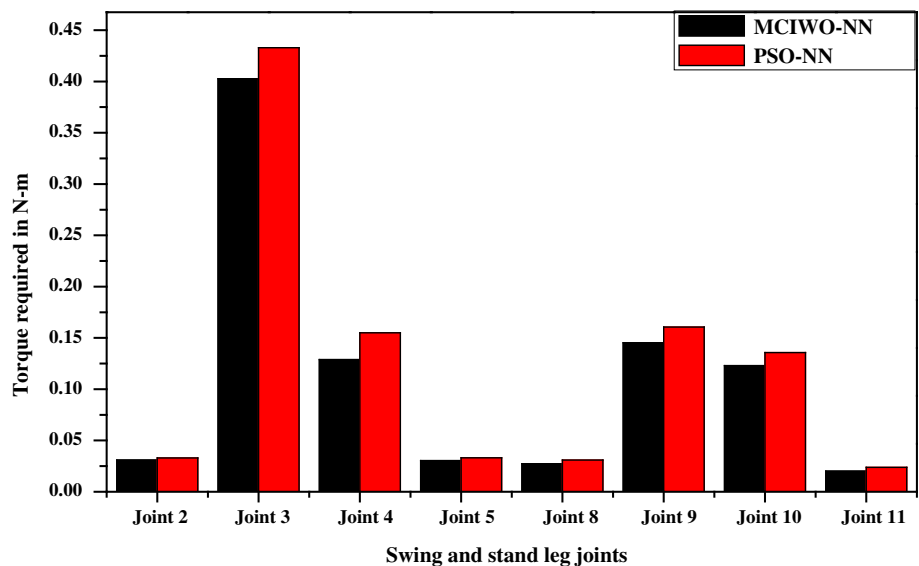
Algorithm 2: PSO algorithm

- 1: Set the maximum and minimum value for each variable
- 2: Generate the swarm randomly in D-dimensional space
- 3: For  $i=1$ :population size  
 Calculate the fitness value  
 If the fitness is better than the previous  $P_{best}$   
 Set current  $P_{best}$ .  
 end
- 4: Choose the best fitness among all the particles that is called  $G_{best}$ .
- 5: For  $i=1$ :population size  
 Calculate the new velocity using the velocity equation  
 Update the new position of the particle using the position equation.  
 End
- 6: Repeat the above process until it reaches the termination criterion.

### 5 Results and discussion

The purpose of the research work is twofold: firstly to illustrate the good quality results attained by the adaptive PID controller tuned by the nature-inspired optimization algorithm to control the biped robot walking on a slope surface, and secondly, to carry out analysis on various performance measures of the controller and the behaviour of the biped robot, such as

Fig. 8 Torque required at various joints of the swing and stand leg while walking on ascending the sloping surface



- An error study has been carried out for the selected nature-inspired algorithms on their capability to tune the gains of the PID controller.
- Analysis is carried out in terms of the position of ZMP and torque required at various joints of the biped robot for both the optimization algorithms.
- A comparative study has been carried out on the basis of the dynamic balance margin values obtained for ascending and descending the sloping surfaces using the said algorithms.
- Finally, the gait angles obtained by the best nature-inspired optimization algorithm are fed to the real biped robot for testing.

The parameters related to the robot, that is, length, mass and inertia of the link, are given in Table 1. The results of convergence for the three approaches, namely MCIWO-NN,

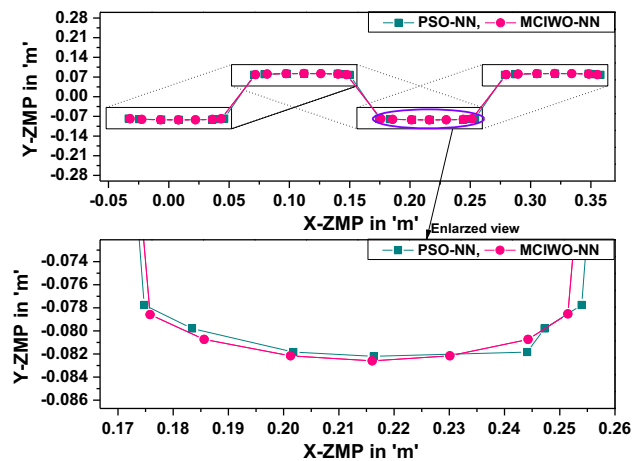
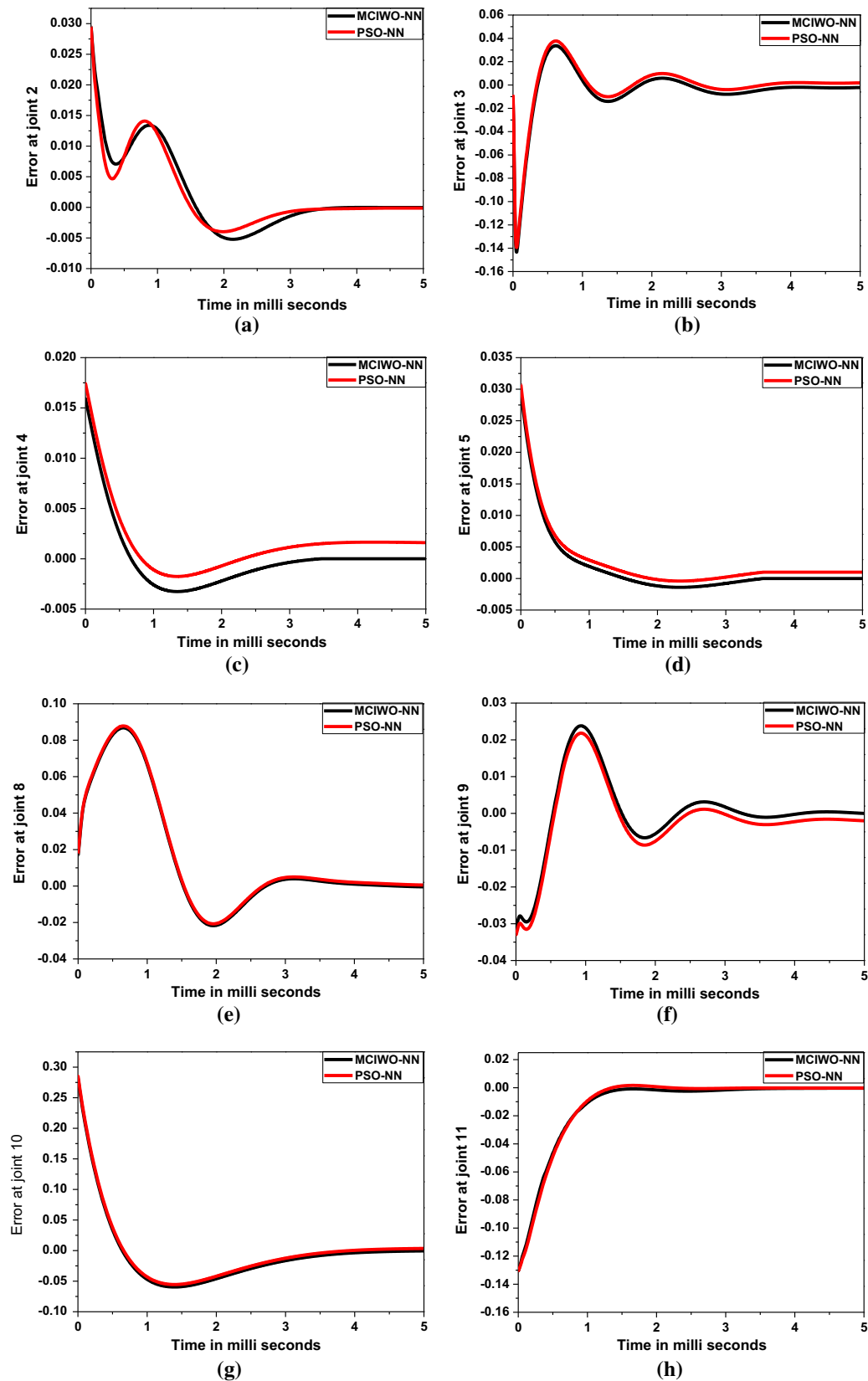


Fig. 9 Variation of zero moment point while ascending the slope surface



**Fig. 10** Error at various joints of the biped robot while descending the slope surface **a** Joint 2, **b** Joint 3, **c** Joint 4, **d** Joint 5, **e** Joint 8, **f** Joint 9, **g** Joint 10 and **h** Joint 11

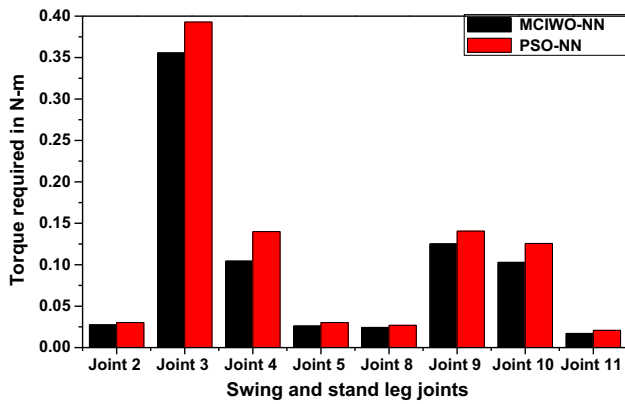


Fig. 11 Torque required at various joints of the swing and stand leg joints while descending the slope surface

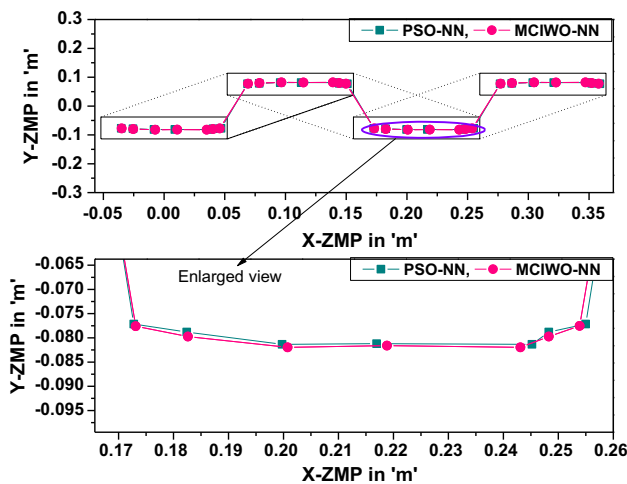


Fig. 12 Variation of zero moment point while moving on descending the sloping surface

PSO-NN and BPNN, are given in Fig. 5. From this test, it can be observed that the MCIWO has shown better performance in terms of convergence of the solution when compared with the PSO and steepest descent method. However, both the MCIWO and PSO are seen to perform better than the steepest descent algorithm. It may be due to the fact that the solutions obtained from the back-propagation algorithm are prone to be stuck in the local minimum.

In the present study, an attempt is made to tune the architecture of the NNs that provide the adaptive gains for the PID controller with the help of two nature-inspired algorithms MCIWO and PSO. The graph showing the effect of variation of neurons in the hidden layer on the performance of the controllers is shown in Fig. 6. The study is conducted by varying the number of hidden neurons between 14 and 22. From Fig. 6, it has been observed that the number of hidden neurons that are responsible for the better performance of MCIWO-NN and PSO-NN approach is seen to be equal to 18 and 16, respectively. The number of input and

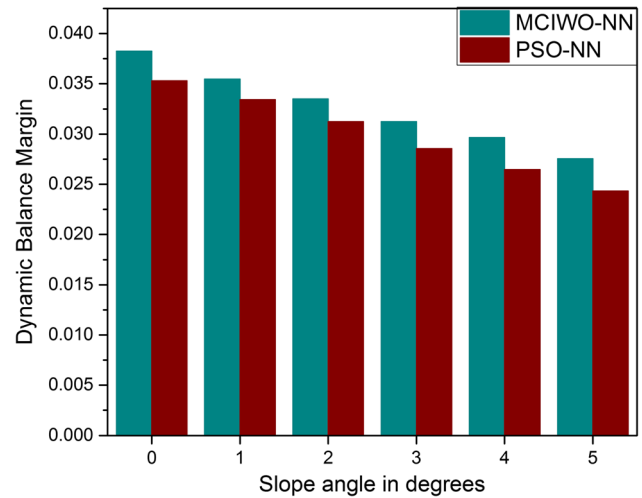


Fig. 13 Graph showing the dynamic balance margin by varying the slope angle

output neurons that correspond to the input and output layers is observed to be equal to 12 and 36, respectively. Therefore, the connecting weights for the MCIWO-NN and PSO-NN are found to be equal to 864 ( $12 \times 18 + 18 \times 36$ ) and 768 ( $12 \times 16 + 16 \times 36$ ). Alongside, two bias values are introduced at hidden and output layers of the network. Therefore, the total number of variables required to optimize the structure of NN with MCIWO and PSO is found to be equal to 866 and 770, respectively.

The parameter setting of these algorithms that are used to train the NN, which produce the gains of the controller for the robot while walking on ascending and descending the slope, is illustrated in Table 2. As nature-inspired algorithms are used in this study, the program is run for a minimum no. of generations (Gen) equal to 30, to set the other parameters of the algorithm. Further to obtain the final convergence, the two algorithms are terminated when the maximum number of fitness function evaluations is equal to 100. Further, each algorithm is run for ten times to determine its stochastic nature.

All the algorithms are implemented in the MATLAB 2013, and the simulations are carried out on a Dell Precision T1700 computer, having the following specifications:

- Processor: Intel(R) Xeon(R) CPU E3-1226 v3 @ 3.30 GHz,
- RAM: 8 GB,
- Operating System: Windows 7 Professional, 64-bit.

The aim of the present study is to illustrate the behaviour of two nature-inspired metaheuristic-based optimization algorithms, i.e. MCIWO and PSO, on tuning the weights of the neural network. Further, the neural network controller is used to provide adaptive gains for the simple PID controller based on the error signal received at the

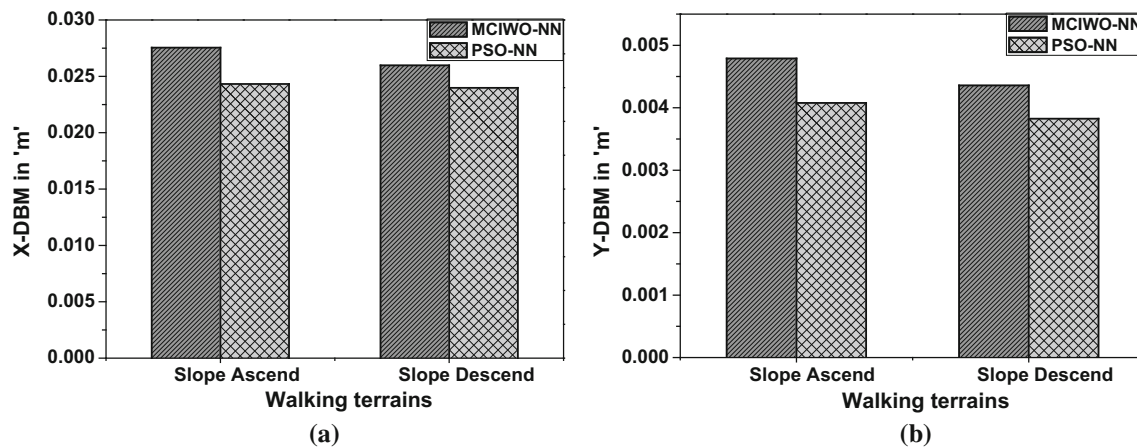


Fig. 14 Dynamic balance margin in  $X$ - and  $Y$ -directions **a**  $X$ -DBM and **b**  $Y$ -DBM

input of NN. The performance of the MCIWO-NN- and PSO-NN-based controllers is compared in terms of error in angular position, torque required at each joint of the biped robot, variation of ZMP and DBM of the biped robot while ascending and descending the sloping surface.

### 5.1 Ascending the slope surface

Figure 7 shows the variation of angular error at various joints of the biped robot after using the MCIWO-NN and PSO-NN adaptive PID controllers. From the error plots, it can be observed that the error for all the joints of the biped robot has shown a certain amount of overshoot initially for both the said cases. Then the overshoot slowly decreases and coincides with zero. Further, it has also been observed that MCIWO-NN is seen to converge quickly when compared with the PSO-NN-based PID controller.

Later on, a comparative study has been made on the torque required by the two-legged robot to move each motor from one position to the next and is shown in Fig. 8. It can be observed that the motor mounted on joint 3 (that is, hip joint of the swing leg) has consumed more torque when compared with the other joint motors of the biped robot. It may be due to the fact that while shifting the leg, the hip joint carries other links and joint motors of the biped robot. Moreover, it has also been observed that PSO-NN-based PID controller is found to consume more torque when compared with the MCIWO-NN PID controller. It might have happened due to the large angular displacement obtained at that particular joint. This was also concluded from error plots in which the convergence took more time when compared with the MCIWO-NN approach (ref. Fig. 5(b)).

Further, a comparative study has also been carried out on the position of ZMP (that is, both in  $X$ - and  $Y$ -directions) obtained for the controlled gait determined by the MCIWO-NN- and PSO-NN-adaptive torque-based PID

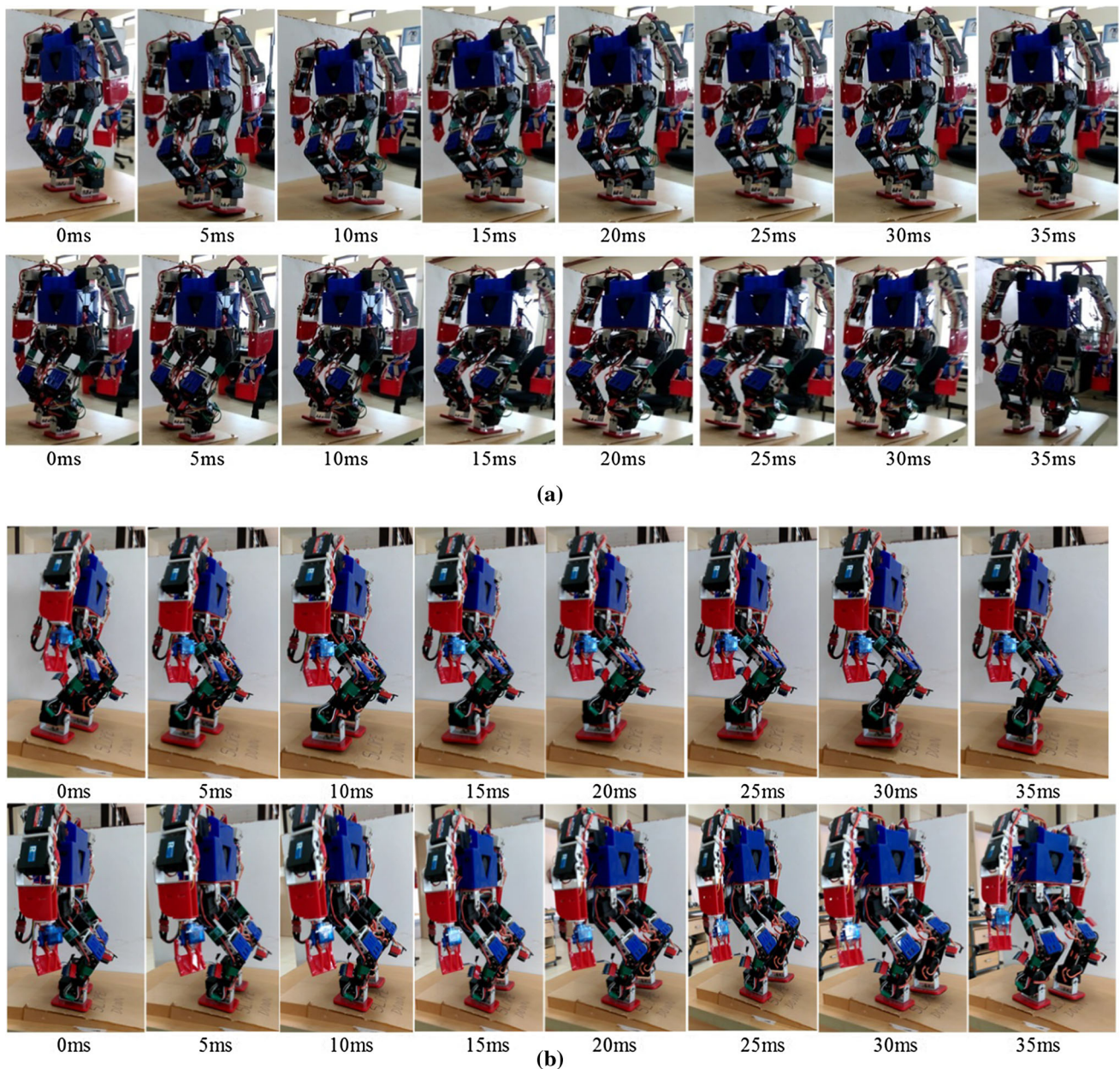
controllers (ref. Fig. 9). It can be observed that the ZMP is found to lie inside the foot support polygon for both the said controllers when the two-legged robot is ascending the slope surface. But, when compared to PSO-NN-based PID controller, the ZMP point of MCIWO-NN-based PID controller is seen to lie close to the centre of the foot. It is important to mention that when the ZMP is close to the centre of the foot, the robot is found to be more dynamically balanced. From this, it can be concluded that MCIWO-NN-based PID controller is found to derive more dynamically balanced gaits when compared with the PSO-NN-based PID controller.

### 5.2 Descending the slope surface

Like ascending the slope surface, analysis is also carried out on descending the slope surface. The variation of angular error at various joints of the biped robot is shown in Fig. 10. Similar to the ascending case, here also initially the error is seen to overshoot and then slowly settles down and coincides with zero. In descending case also the error is found to settle down and coincides quickly with zero in the MCIWO-NN-based PID controller when compared with PSO-NN-based PID controller.

Figure 11 shows the torque required to move the motors at various joints of the biped robot while descending the slope. Similar observation as that of ascending the slope, regarding the torque required hip joint, is conceived during descending the slope. It is also observed that the torque required for all the joints is less when the robot is descending the slope when compared with ascending the slope. This might have happened due to the smaller angular displacements of the links when compared with the ascending case.

The variation of ZMP for the biped robot while descending the slope surface is given in Fig. 12. In this case also MCIWO-NN torque-based PID controller is



**Fig. 15** Biped robot walking on different surfaces **a** ascending of the slope surface; and **b** descending of the slope surface

observed to produce dynamically more balanced gaits when compared with the PSO-NN-based controller. The reason for this is also same as the one explained for the ascending case. Moreover, the DBM values obtained by the MCIWO-NN adaptive PID controller are seen to be high when compared with the PSO-NN adaptive PID controller in both X- and Y-directions. This may be due to the global optimal nature induced in the MCIWO by allowing all the weeds to participate in the reproduction process. Based on the above process, fitter plants will produce more seeds than less-fit plants, which is the reason to improve the

convergence of the algorithm. Another important feature of MCIWO is that the weeds reproduce new seeds independently without mating and contribute towards the generation of fit plants. Further, additional experiments are conducted to study the advantage of MCIWO-NN over PSO-NN by varying the angle of slope from  $0^\circ$  to  $5^\circ$  (refer to Fig. 13). It can be observed that the MCIWO-NN-based PID controller has consistently outperformed PSO-NN-based PID controller in terms of dynamic balance margin of the biped robot while negotiating the slope surface.

### 5.3 Comparative study

A comparative study has been conducted in terms of DBM in X- and Y-directions while ascending and descending the slope surfaces. From Fig. 14a, b, it can be observed that the DBM is high when the robot is ascending the slope surface when compared with descending the slope surface.

From the above analysis, it has been concluded that MCIWO-NN-based PID controller is found to perform better than the PSO-NN-based PID controller. The controlled gait obtained using the best controller (that is, MCIWO-NN-based PID controller) is fed to the real robot to test the performance of the controller. From Fig. 15a, b, it can be observed that the two-legged robot has successfully performed walk while ascending and descending the slope, respectively. Further, a qualitative comparison has been performed with other approaches provided in [1, 2, 7, 33–36]. They considered a 7-DOF biped robot and used analytical approach to generate the gaits of the robot. In the present approach, the authors have not only considered the gait generation issues of the biped robot, but also designed a PID controller to execute the generated gait in a smooth manner. Moreover, in [18, 26–29], some researchers had developed the ZMP-based PID controllers, which are an indirect way of controlling the motors mounted on the joints of the robot, whereas in the present study a torque-based PID controller directly deals with the error in angular displacements of the motors. Further, in [20–25], the researchers had used evolutionary algorithms that provide only one set of gain values for the PID controller to tune the gains of the same. However, in the present research, an evolutionary algorithm trained NN is used to predict the gains of the PID controller in an adaptive manner to suit the magnitude of error signals. The said algorithm is tested on a real biped robot.

## 6 Conclusion

In general, tuning of the PID controller is a process of determining the optimal values of its gains before the start of the run. Compared to other controllers, the simplified PID controllers are more popular and powerful control systems used in industrial and non-industrial applications. In the present research paper, an attempt is made to develop an adaptive torque-based PID controller with the help of MCIWO-NN and PSO-NN algorithms. The results of these adaptive algorithms are analysed in terms of error, torque required for each joint, variation of ZMP and DBM of the biped robot in order to obtain the best suitable algorithm for this purpose. In this analysis, the online response of the MCIWO-NN-based control algorithm is seen to be better when compared with the PSO-NN-based

control algorithm. This might have happened due to the introduction of cosine and chaotic variables in the IWO algorithm. Further, the gait angles obtained by the best adaptive PID controller (i.e. MCIWO-NN) are fed to the real robot. It has been observed that the real biped robot has successfully performed walk on ascending and descending the slope surface.

### Compliance with ethical standards

**Conflict of interest** All authors declare that they have no conflict of interest.

## References

- Shih CL (1999) Ascending and descending stairs for a biped robot. *IEEE Trans Systems Man Cybern A: Syst Hum* 29:255–268
- Huang Q, Yokoi K, Kajita S, Kaneko K, Arai H, Koyachi N, Tanie K (2001) Planning walking patterns for a biped robot. *IEEE Trans Robot Autom* 17:280–289
- Sugihara T, Nakamura Y, Inoue H (2002) Real time humanoid motion generation through ZMP manipulation based on inverted pendulum control. In: *Proceeding of the international conference on robotics and automation, Washington, USA*, pp 1404–1409
- Kajita S, Kanehiro F, Yokoi K, Hirukawa H (2001) The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In: *Proceeding of the international conference on intelligent robots and systems, Maui, USA*, pp 239–246
- Park IW, Kim JY, Lee J, Oh JH (2006) Online free walking trajectory generation for biped humanoid robot KHR-3 (HUBO). *Proceeding of the international conference on robotics and automation, Orlando, Florida, USA*, pp 1231–1236
- Seven U, Akbas T, Fidan KC, Erbatur K (2012) Bipedal robot walking control on inclined planes by fuzzy reference trajectory modification. *Soft Comput* 16:1959–1976
- Chiang MH, Chiang FR (2013) Anthropomorphic design of the human-like walking robot. *J Bionic Eng* 10:186–193
- Ren L, Qian ZH, Ren LQ (2014) Biomechanics of musculoskeletal system and its biomimetic implications: a review. *J Bionic Eng* 11:159–175
- Vukobratovic M, Stepanenko J (1972) On the stability of anthropomorphic systems. *Math Biosci* 15(1/2):1–37
- Goswami A (1999) Foot rotation indicator (FRI) point: a new gait planning tool to evaluate postural stability of biped robots. In: *Proceeding of the IEEE international conference on robotics and automation*, pp 47–52
- Kajita S, Kanehiro F, Kaneko K, Fujiwara K, Harada K, Yokoi K, Hirukawa H (2003) Biped walking pattern generation by using preview control of zero-moment point. In: *Proceeding of the IEEE international conference on robotics and automation*, pp 1620–1626
- Shibuya M, Suzuki T, Ohnishi K (2006) Trajectory planning of biped robot using linear pendulum mode for double support phase. In: *Proceeding of the 32nd annual conference of the IEEE industrial electronics society*, pp 4094–4099
- Erbatur K, Kurt O (2009) Natural ZMP trajectories for biped robot reference generation. *IEEE Trans Ind Electron* 56(3):835–845

14. Shih C-L, Chiou C-J (1998) The motion control of a statically stable biped robot on an uneven floor. *IEEE Trans Syst Man Cybern B: Cybern* 28(2):244–249
15. Hwang SW, Yeon JS, Park JH (2013) Trajectory generation method for biped robots to climb up an inclined surface. In: 44th international symposium on robotics (ISR), pp 1–5
16. Cristiano J, Puig D, Garcia MA (2015) Efficient locomotion control of biped robots on unknown sloped surfaces with central pattern generators. *Electron Lett* 51(3):220–222
17. Choi T-Y, Park S-Y, Lee J-J (2006) A hybrid SOF-PID controller for a MIMO biped robot. *Artif Life Robot* 10:69–72
18. Bouhajar S, Maherzi E, Khraief N, Besbes M, Belghith S (2015) Trajectory generation using predictive PID control for stable walking humanoid robot. *Procedia Comput Sci* 73:86–93
19. Anh HPH, Huan TT, Nam NT (2014) Novel robust walking for biped robot using adaptive neural PID controller. In: International conference on automatic control theory and application, pp 86–89
20. Engelbrecht AP (2007) *Computational intelligence: an introduction*. Wiley, Hoboken
21. Shahbazi H, Jamshidi K, Monadjemi AH, Eslami H (2014) Biologically inspired layered learning in humanoid robots. *Knowl Syst* 57:8–27
22. Yau TJ, Yun TC, Chih PH (2008) Design of fuzzy PID controllers using modified triangular membership functions. *J Inf Sci* 178(12):1325–1333
23. Hang CC, Astrom KJ, Wang QG (2002) Relay feedback auto-tuning of process controllers: a tutorial review. *J Process Control* 12(1):143–162
24. Fister D, Fister Jr I, Fister I, Safaric R (2016) Parameter tuning of PID controller with reactive nature-inspired algorithms. *Robot Auton Syst* 84:64–75
25. Varol HA, Bingul Z (2004) A new PID tuning technique using ant algorithm. In: Proceeding of the 2004 American Control Conference Boston, Massachusetts
26. Alcaraz-Jimenez JJ, Herrero-Perez D, Martinez-Barbera H, JJ (2013) A simple feedback controller to reduce angular momentum in ZMP-based gaits. *Int J Adv Robot Syst* 10:1–7
27. Aghbali B, Ousefi-Koma AY (2013) ZMP Trajectory control of a humanoid robot using different controllers based on an offline trajectory generation. In: Proceeding of the 2013 RSIISM international conference on robotics and mechatronics, Tehran, Iran
28. Kajita S, Morisawa M, Miura K, Nakaoka S, Harada K, Kaneko K, Kanehiro F, Yokoi K (2010) Biped walking stabilization based on linear inverted pendulum tracking. In: The 2010 IEEE/RSJ international conference on intelligent robots and systems, Taipei, Taiwan
29. Park S, Han Y, Hahn H (2009) Balance control of a biped robot using camera image of reference object. *Int J Control Autom Syst* 7(1):75–84
30. Amin AE (2013) A novel classification model for cotton yarn quality based on trained neural network using genetic algorithm. *Knowl-Based Syst* 39:124–132
31. Mehrabian AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. *Ecol Inf* 1(4):355–366
32. Kennedy J, Eberhart R (1995) Particle swarm optimization. *Proceedings of IEEE international conference on neural networks, Perth, vol 27*, pp 1942–1948
33. Vundavilli PR, Pratihari DK (2011) Balanced gait generations of a two-legged robot on sloping surface. *Sadhana* 36(4):525–550
34. Mousavi PN, Bagheri A (2007) Mathematical simulation of a seven link biped robot on various surfaces and ZMP considerations. *Appl Math Model* 31:18–37
35. Ranjbar M, Mayorga RV (2017) A seven link biped robot walking pattern generation on various surfaces. *WSEAS Trans Syst* 16:299–312
36. Bagheri A, Miripour-Fard B, Mousavi PN (2010) Mathematical modelling and simulation of combined trajectory paths of a seven link biped robot. *Climbing and Walking Robots*, Published by Intech Open

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.