



# Implementation of modified chaotic invasive weed optimization algorithm for optimizing the PID controller of the biped robot

RAVI KUMAR MANDAVA\* and PANDU R VUNDAVILLI

School of Mechanical Sciences, Indian Institute of Technology Bhubaneswar, Bhubaneswar 751013, India  
e-mail: rm19@iitbbs.ac.in; pandu@iitbbs.ac.in

MS received 5 June 2017; revised 24 January 2018; accepted 24 January 2018; published online 25 April 2018

**Abstract.** The present research work involves the implementation of Modified Chaotic Invasive Weed Optimization (MCIWO) algorithm for optimizing the gains of torque based proportional integral and derivative (PID) controller used to control the motors of the biped robot while walking on flat surface. While designing the controller, the dynamics of the biped robot has been derived using the well-known Lagrange-Euler (L-E) formulation. Subsequently, manual tuning procedure is employed to find the ranges of the gains of PID controller used in the developed algorithm. Once it is optimized, the effectiveness of the proposed algorithm is then compared with the Differential Evolution (DE) algorithm, in terms of variation of error, torque required, zero moment point (ZMP) and dynamic balance margin (DBM) of the biped robot. It has been observed that the MCIWO algorithm tuned PID controller is found to perform better than DE tuned controller. Further, the optimal gait obtained through the developed algorithm is validated by executing it on the real robot. It has been observed that the robot has successfully negotiated the flat terrain with the gaits obtained by the optimal PID controller.

**Keywords.** Modified Chaotic Invasive Weed Optimization (MCIWO); PID controller; dynamic balance margin; biped robot.

## 1. Introduction

Recently, the usage of biped robots are growing in various applications such as industry, houses, offices and hospitals. Based on the utility of the biped robots, most of the research is focussed on the development of human friendly features which are partially inspired by the growth in the technology. The major challenge in the area of biped robotics is to generate dynamically balanced gaits for the robot in an unknown environment. Further, for the past few decades, many researchers were working on the development of various types of linear and non-linear controllers that generate dynamically balanced walking gaits for the biped robot. Up to now, most of the research is focused on the generation of dynamically balanced gaits only in the sagittal plane. In the present research work, the gait of the biped robot is considered in both the sagittal and frontal planes and designed a torque based PID controllers while walking on the flat surface.

Some of the literature related to the generation of dynamically balanced gaits of a biped robot is presented below. A semi-inverse method was proposed by Juricic and Vukobratovic [1] to determine the trunk motion of a biped robot. In that work, initially the ZMP position was defined,

and then the trunk motion was calculated based on the defined position of ZMP. In [2], Furusho and Masubuchi introduced a new reduced order model based on the concept of local feedback of the system to develop a stable locomotion. The said reduced order model was derived after considering the effect of motion of the body and swing leg of the biped robot. Further, Seo and Yoon [3] conducted several simulations for the biped robot on various terrain conditions and designed a robust dynamic gait based on the concept of ZMP and DBM. Moreover, Lum *et al* [4] developed computed torque control scheme and variable structural control law to control a 5-DOF biped robot. The performance of the developed method was tested on a real robot and found that it could walk on a horizontal surface and climb a stair case. For achieving dynamic walking, Kim *et al* [5] used distributed control architecture to control all the joints of the robot in a smooth way. The architecture was reduced to limit the computation effect of the main controller on the system. Further, a servo motor was used as the sub-controller for the inertia sensor and a 3-axis force/torque sensor used in the sensory system. The results of simulation was tested on KHR-2, a 41-DOF humanoid robot. Pandu *et al* developed dynamically balanced gait generation algorithm for a 7-DOF biped robot on different terrains i.e., stair case [6] and slope surface [7] after using the concept of zero moment point. In that work, the authors

\*For correspondence

had not used any controller to control the robot, and the degrees of freedom of the robot is limited. In [8], Hernandez *et al* designed a PD control algorithm with gravity compensation on a 16- DOF biped robot for obtaining desired trajectories, and to determine the torque required at each joint. Later on, to generate smooth walking gaits of the biped robot in both single and double support phase, Al-Shuka *et al* [9] proposed an algorithm that induce smooth transition from SSP to DSP and vice versa by using three methods i.e., LIPM, LPM and COG. Further, they also implemented different foot trajectories in DSP to improve the stability of the biped robot. Katic and Vukobratovic [10] developed an intelligent control technique after using genetic algorithm (GA), neural network (NN) [11], fuzzy logic controller (FLC) and their hybrid versions, such as neuro genetic, neuro-fuzzy networks and fuzzy genetic algorithms in the area of biped robotics.

In addition to the above methods, Patrizio [12] developed an adaptive PD controller for controlling the robotic manipulators. The controller was shown to be globally convergent and the simulations were conducted on a 3-DOF robot. From the results of simulation, it was observed that the performance of the adaptive PD controller had hardly influenced by the initial error, although the performance of the PID control law deteriorated considerably as the error was increased. A tuning procedure was proposed by Rafael [13] to tune the controller of a robotic manipulator. They utilized Lyapunov function together with the LaSalle invariance principle to obtain the required tuning procedure. For tuning the PID controller by using these approaches, in depth knowledge over the inertia matrix and gravitational torque vectors was required. Further, Khoury *et al* [14] developed a fuzzy PID controller for a 5-DOF robotic arm and the performance of their controller was compared with the help of the computed torque control method and direct adaptive control method. Moreover, to tune the PID controller parameters of a 4-DOF spatial and planar manipulators, Ravi and Pandu [15] utilized manual tuning procedure. The performance of these controllers were tested in computer simulations. Further, to control the multiple inputs and outputs of a robotic arm, Hassan [16] developed a self-organising fuzzy PID controller. It was observed that the output trajectories of the SOF-PID controller path was closer and smoother than the output trajectories of the SOFC and PID controller. Later on, Aulia *et al* [17] used Reinforcement learning (RL) algorithm with Q-Learning approach to tune the PID control parameters of the soccer robot. Finally, they compared the results of the approach with Ziegler–Nichols tuning method. The response of RL algorithm was 1.5 times faster than the Ziegler–Nichols tuning method. Jeong *et al* [18] used a non-parametric estimation method based on particle swarm optimization (PSO) to search the parameters used in the central pattern generator of the two legged robot. Moreover, Helon *et al* [19] proposed a multi-objective genetic algorithm (that is, NSGA-II) to tune the gains of the PID

controller of the robotic manipulator. It was observed that the performance of the NSGA-II was good at reference tracking and provides robust solutions for the robotic manipulator. In [20], Mohd *et al* discussed the tuning of PID controller parameters using differential evolution (DE) and genetic algorithm (GA). The performance of these algorithms were compared with the standard Ziegler–Nichols method. Further, Ravi *et al* [21] developed a tuning procedure for the PID controller of a 3-DOF planar robotic manipulator after using GA and PSO algorithms. It was observed that the PSO algorithm was seen to be performed better than GA in terms of error in the joint angle and torque required at various joints. Moreover, Geetha *et al* [22] implemented a virtual feedback controller to control the state variables of the continuously stirred tank chemical reactor and used extended kalman filter in the feedback mechanism. They used GA, PSO [23] and ACO algorithms for determining the optimal PID parameters. It was found that the PSO algorithm was seen to perform better than the other methods. Joel *et al* [24] developed adaptive neural network controller for tracking the trajectory of a 2-DOF manipulator. The Lyapunov control function was used for analysing the stability of the tracking error, and the PID approach was used to determine the control law.

Further, Dusan *et al* [25] implemented some nature-inspired optimization algorithms, such as GA, PSO, Bat Algorithm (BA) [26], Hybrid Bat Algorithm (HBA), DE and Cuckoo Search (CS) algorithms to obtain the optimal parameters for the PID controller. They applied these algorithms for the online response of the PID controller by applying sudden loads, and observed a quick response. In [27], the authors had developed an optimal adaptive PID controller based on fuzzy rules and introduced a sliding mode controller to control the multi input and multi output chaotic nonlinear systems. A multi objective GA was used to optimize the parameters of the proposed controller. No work is reported on the usage of invasive weed optimization algorithm in the area of the proposed research. However, this algorithm was used to optimize the gains of the PID controller in other engineering applications. Navid and Mohsen [28] used invasive weed optimization (IWO) and Artificial Bee Colony (ABC) algorithms to tune the gains of the PID controller for hydro-turbine generator. The results of the simulation revealed that IWO algorithm had performed better than the ABC algorithm at different load conditions of the power system. Further, Abedinia *et al* [29] used a modified version of IWO algorithm to tune the parameters of fuzzy PID controller used in multi machine power system. A Bacterial Foraging Optimization algorithm (BFO) was also used by Latha and Rajinikanth [30] to tune the parameters of PID controller used in a 2-DOF manipulator. The robustness of the algorithm was tested by introducing uncertainty in the model parameters. In addition to the above algorithms, Nikhileshwar and Amith [31] used GA to tune the parameters of the PID controller that was in the position control of a DC motor. Recently, another

optimization algorithm called Antlion optimizer [32] was also used to tune the gains of the PID controller used in automobile cruise control system. Later on, Naidu and Ojha [33] developed a hybrid version of IWO with quadratic approximation, and compared with standard invasive weed optimization algorithm. In [34], [35] the authors discussed the applicability of invasive weed optimization algorithm in electromagnetic applications. Further, various new optimization algorithms such as chemical reaction optimization [36] and fireworks optimization [37] algorithms were developed by various authors to estimate the optimal parameters related to different engineering problems.

Based on the above literature, it has been observed that most of the researchers had used ZMP based PID controller to control the motors of the biped robot while executing the motion. It is important to mention that the ZMP based governing is an indirect method of controlling the motors in which the error signal between the obtained and reference ZMP trajectory is used to design the gaits for various links of the biped robot. During the said process, the controller may suggest some dynamically balanced postures which may not be kinematically feasible. Further, finding the most appropriate gaits of the biped robot that fulfils the sequence oriented criterion and repeatability conditions is challenging to achieve. Whereas torque-based PID controller considers the error signal of each joint of the biped robot, and try to design the gains of the controller by minimizing the error between two consecutive gait angles of each joint of the biped robot. With the best of the author's knowledge, no work is reported on the design of torque based PID controller for the biped robot. Moreover, none of the researcher had tried to implement the chaotic nature in IWO algorithm that helps in distributing the seeds in a wider spread that helps in enhancing the search space. In the present paper, an attempt is made to develop an appropriate torque-based PID controller for all the joints of the lower limbs of a biped robot to achieve stable walk. Further, the gains of the PID controllers are tuned with the help of a modified non-traditional optimization algorithm i.e., MCIWO developed in this research. Once the optimal PID controllers are designed, the effectiveness of the algorithm is compared with the existing DE tuned PID controller in terms of dynamic balance margin of the biped robot. Further, the gaits obtained using the best optimal PID controller are tested on a real robot and found that the robot has successfully performed its walk on flat terrain.

The rest of the paper is organised as follows. Section 2 explains the formulation of the problem that includes the kinematics, dynamics and design of PID controller. Section 3 discusses the formulation of the optimization problem. The description related to the non-traditional optimization algorithms are given in section 4. The results of simulation and experimentation are presented and discussed in section 5. Further, the conclusions of the said work and scope for future work are provided in section 6 and 7, respectively.

## 2. Formulation of the problem

The problem solved in the present research may be detailed as follows: A torque-based PID controller will be developed for the 18-DOF biped robot (figure 1) walking on the flat surface after maintaining the balance of the robot in both sagittal and frontal planes. The kinematics, dynamics and design of the torque-based PID controller for the said robot are explained below. The gains of the proposed PID controller are optimized with the help of a non-traditional optimization algorithm, MCIWO developed in this study.

### 2.1 Inverse kinematics of the biped robot

The concept of inverse kinematics is used to determine the included angles made between the lower and upper limbs of the swing leg of the biped robot. The mathematical expressions used for obtaining the joint angles of swing leg  $\theta_4$  and  $\theta_3$  in sagittal plane are given below.

$$\theta_4 = \sin^{-1} \left( \frac{H_1 l_3 \sin \psi + L_1 (l_4 + l_3 \cos \psi)}{(l_4 + l_3 \cos \psi)^2 + (l_3 \sin \psi)^2} \right) \quad (1)$$

where  $H_1 = l_4 \cos \theta_4 + l_3 \cos \theta_3$ ,  $L_1 = l_4 \sin \theta_4 + l_3 \sin \theta_3$ ,  $\psi = \theta_4 - \theta_3 = \arccos((H_1^2 + L_1^2 - l_4^2 - l_3^2)/2l_4l_3)$ . Thus,  $\theta_3$  can be obtained from the equation  $\theta_3 = \theta_4 - \psi$ . Similarly, the angles  $\theta_9$  and  $\theta_{10}$  are obtained by using the following mathematical expressions.

$$\theta_{10} = \sin^{-1} \left( \frac{H_2 l_9 \sin \psi + L_2 (l_{10} + l_9 \cos \psi)}{(l_{10} + l_9 \cos \psi)^2 + (l_9 \sin \psi)^2} \right) \quad (2)$$

where  $H_2 = l_9 \cos \theta_9 + l_{10} \cos \theta_{10}$ ,  $L_2 = l_9 \sin \theta_9 + l_{10} \sin \theta_{10}$ ,  $\psi = \theta_{10} - \theta_9 = \arccos((H_2^2 + L_2^2 - l_{10}^2 - l_9^2)/2l_{10}l_9)$ . Thus,  $\theta_9$  can be obtained from the equation  $\theta_9 = \theta_{10} - \psi$ . The angle at ankle with respect to vertical axis is assumed to be equal to zero i.e.,  $\theta_6 = \theta_{12} = 0$ . The mathematical relationships used for determining the joint angles of both the legs in frontal plane are given below.

$$\theta_2 = \theta_8 = \tan^{-1}(f_w/H_1) \quad (3)$$

$$\theta_5 = \theta_{11} = \tan^{-1}((0.5 * f_w)/H_2) \quad (4)$$

It is important to note that the joint angles in both the sagittal and frontal planes of the robot are needed to follow repeatability conditions for smooth transition from single support phase to double support phase and vice versa.

### 2.2 Dynamic balance margin of the biped robot

While walking, the gait generated by the biped robot should be dynamically balanced. For achieving dynamically balanced gait, the zero moment point should lie inside the foot support polygon. The foot-ground interaction model, in

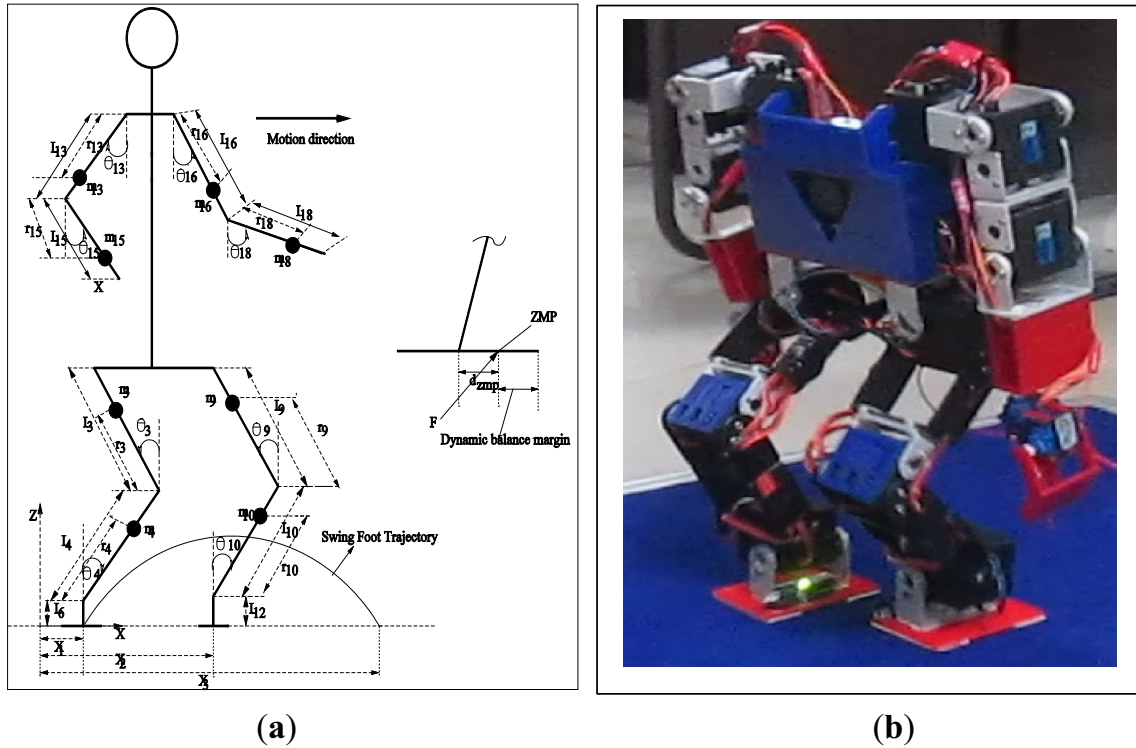


Figure 1. A schematic diagram showing the full body of biped robot: (a) stick diagram and (b) real robot.

which the position of ZMP and forces acting on the foot of the biped robot are shown in figure 2.

The location of ZMP in both the directions i.e.,  $X_{zmp}$  and  $Y_{zmp}$  is calculated by using the following equations:

$$x_{ZMP} = \frac{\sum_{i=1}^n (I_i \dot{\omega}_i - m_i \ddot{x}_i z_i + m_i x_i (g - \ddot{z}_i))}{\sum_{i=1}^n (m_i (\ddot{z}_i - g))} \quad (5)$$

$$y_{ZMP} = \frac{\sum_{i=1}^n (I_i \dot{\omega}_i - m_i \ddot{y}_i z_i + m_i y_i (g - \ddot{z}_i))}{\sum_{i=1}^n (m_i (\ddot{z}_i - g))} \quad (6)$$

where  $\dot{\omega}_i$  and  $I_i$  denotes the angular acceleration ( $\text{rad/s}^2$ ) and mass moment of inertia ( $\text{kg}\cdot\text{m}^2$ ) of the  $i^{\text{th}}$  link,  $g$  and  $m_i$  represents the acceleration due to gravity ( $\text{m/s}^2$ ) and mass ( $\text{kg}$ ) of the link  $i$ ,  $\ddot{z}_i$  and  $\ddot{x}_i$  indicates the acceleration of the  $i^{\text{th}}$  link moving in  $z$  and  $x$  directions ( $\text{m/s}^2$ ),  $(x_i, y_i, z_i)$  signifies the coordinates of the  $i^{\text{th}}$  lumped mass.

It is important to note that, the ZMP should always lie inside the foot support polygon to ensure dynamically balanced gaits of a biped robot. The DBM, which is used to measure the amount of balance of the robot is defined as the distance between the ends of the foot support polygon to the point, where the ZMP is acting on the foot at the foot ground interaction. The following expressions are used to calculate the quantities  $x_{DBM}$  and  $y_{DBM}$  of the biped robot.

$$x_{DBM} = \left( \frac{f_s}{2} - |x_{ZMP}| \right) \quad (7)$$

$$y_{DBM} = \left( \frac{f_w}{2} - |y_{ZMP}| \right) \quad (8)$$

where  $f_s$  and  $f_w$  represent the length and width of the foot support, respectively. It is important to note that in the present research the orientation of the swing foot is parallel to the ground.

### 2.3 Dynamics of the biped robot

The parameters, such as linear and angular velocity of the joints are very important for the locomotion of the biped robot. The Jacobian method is used to determine the linear ( $J_{vi}$ ) and angular ( $J_{\omega i}$ ) velocity of the joints of the biped robot.

$$J_i = \begin{bmatrix} J_{vi} \\ J_{\omega i} \end{bmatrix} = \begin{bmatrix} Z_{i-1} \times (O_n - O_{i-1}) \\ Z_{i-1} \end{bmatrix} \quad (9)$$

Once the angular velocity and angular acceleration are evaluated, the dynamics of the biped robot is determined with the help of Lagrange-Euler formulation, which is given below.

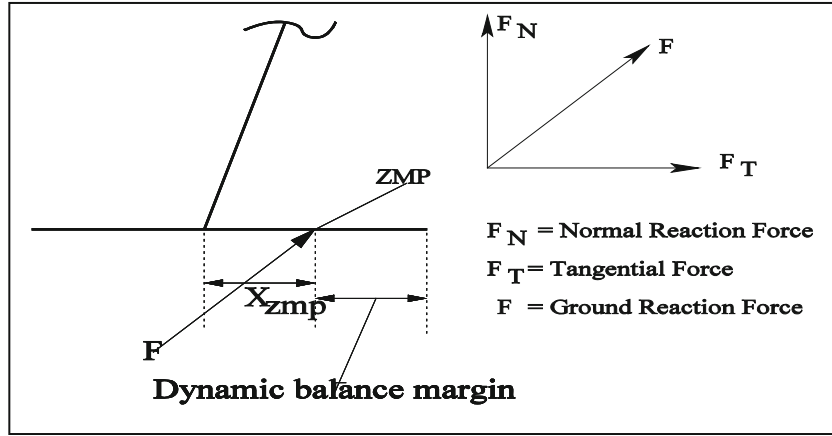


Figure 2. Schematic diagram showing the foot-ground interaction model.

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

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

(10)

where  $\tau_{ithe}$  indicates the torque required at joint  $i$  (N-m),  $q_j, \dot{q}_j$  and  $\ddot{q}_j$  represent the displacement, velocity (rad/sec) and acceleration (rad/sec<sup>2</sup>) of the joint, respectively. While controlling the joint, the acceleration of the link plays a major role. Therefore, the expression in terms of acceleration will be obtained by rearranging the terms of equation (10), and is given below.

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

$$+ \left( \sum_{j=1}^n M_{ij}(q)^{-1} * \tau_{i,the} \right)$$

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

(11)

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

(12)

the torque required can be obtained by using the equation (13), which is given below.

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

(13)

### 2.4 Design of PID controller for biped robot

After solving the dynamics of the biped robot, the PID controllers are employed to control the motors of all the joints of the biped robot. The main aim of the PID controller is to diminish the error between the actual process variable to the reference of the system. The equation for the joint based PID controller that is used in this study is given below as in eq. (14).

$$\tau_{act} = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int e(t) dt$$

(14)

where  $e(t)$  represent the error in the displacement of joint and  $K_p, K_d$  and  $K_i$  indicate proportional, derivative and integral gains for the controller, respectively. The expanded form of the above equation after including the meaning of  $e(t)$  is given below.

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

(15)

$$\tau_{i,act} = K_{pi}(\theta_{if} - \theta_{is}) - K_{di} \dot{\theta}_{is} + K_{ii} \int e(\theta_{is}) dt$$

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

(16)

where  $\tau_{i,act}$  represent the actual torque supplied by the controller to individual joints to move from an initial angular position ( $\theta_{is}$ ) to final angular position ( $\theta_{if}$ ). The final control equation that represents the control action for all joints is given below.

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

$$+ K_{pi}(\theta_{if} - \theta_{is}) - K_{di}\dot{\theta}_{is} + K_{ii} \int e(\theta_{is}) dt$$

(17)

Further, the integral terms in the above equation need to be substituted by its state variables, and is given below.

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

(18)

### 3. Formulation as an optimization problem

In the present study, the biped robot has to move on the flat terrain with the help of dynamically balanced gaits obtained from the torque-based controller, whose gains are optimized with an objective to minimize the error in angular

displacement of the swing and stand legs. There are several entities such as, links of the robot, orientation of the foot, structure of the body, nature of the surface, walking with single or double support phases and the nature of the gait in both the sagittal as well as frontal planes, etc. influence the balance of the biped robot in real time and unique when compared with the other mobile robots. The improper planning and lack of coordination among these entities lead to decrease in the balance of the robot and the gait may fail also. To maintain proper balance of the biped robot while walking on flat terrain it is necessary to control the motors mounted on the joints in an appropriate manner. Minimization of error in angular displacement of the motors mainly depend on various constraints related to the controller. The major constraints include the proportional, derivative and integral gains of the controllers that are used to control the motors mounted on swing and stand legs. The problem aims at evolving the gains of the controllers that minimizes the error in angular displacement of both the swing and stand leg. The various assumptions and notations considered while developing the model are explained below.

### 3.1 Assumptions

- (1) The double support phase is instantaneous and is used to exchange the leg support.
- (2) The friction between the foot and ground is sufficient enough for the robot to avoid slipping while walking on the floor.
- (3) Free movement for upper body of the biped robot is considered for proper balancing of the biped robot.

### 3.2 Notations

In the present research,  $i$  and  $j$  are used as two sets/indices that are used to represent number of joints in the swing and stand leg, respectively. The parameters and decision variables used in this study are given in tables 1 and 2, respectively.

### 3.3 Objective function

The overall objective of the present model is to minimize the total error which comprises of error in angular displacement of both the swing and stand legs and is formulated [38 and 39] as follows:

*Minimize total error = Error in angular displacement for swing leg + Error in angular displacement for stand leg*

The description of the objective functions is given by Equations (19) and (20), respectively.

**Table 1.** The parameters used in the problem.

$e_1$	Error in angular displacement of the swing
$e_2$	Error in angular displacement of the swing
$i_s$	Initial joint angle of the swing leg
$i_f$	Final joint angle of the swing leg
$j_s$	Initial joint angle of the stand leg
$j_f$	Final joint angle of the stand leg

**Table 2.** The decision variables used in the problem.

$K_{pi}$	Proportional gain for the $i^{\text{th}}$ joint of the swing leg
$K_{di}$	Derivative gain for the $i^{\text{th}}$ joint of the swing leg
$K_{ii}$	Integral gain for the $i^{\text{th}}$ joint of the swing leg
$K_{pj}$	Proportional gain for the $j^{\text{th}}$ joint of the stand leg
$K_{dj}$	Derivative gain for the $j^{\text{th}}$ joint of the stand leg
$K_{ij}$	Integral gain for the $j^{\text{th}}$ joint of the stand leg

$$\begin{aligned} & \text{Error in angular displacement for swing leg } (e_1) \\ & = \sum_{i=1}^6 (\theta_{if} - \theta_{is}) \end{aligned} \quad (19)$$

$$\begin{aligned} & \text{Error in angular displacement for stand leg } (e_2) \\ & = \sum_{j=7}^{12} (\theta_{jf} - \theta_{js}) \end{aligned} \quad (20)$$

**Subject to constraints:** The bounds on various design variables related to the problem are given below.

$$\begin{aligned} K_{pi,\min} &\leq K_{pi} \leq K_{pi,\max} \\ K_{di,\min} &\leq K_{di} \leq K_{di,\max} \quad i = 1, 2, \dots, 6 \\ K_{ii,\min} &\leq K_{ii} \leq K_{ii,\max} \end{aligned} \quad (21)$$

The constraints provided in Eq. (21) restricts the variation of proportional, derivative and integral gains, to minimize the error in the angular displacement of various joints of the swing leg at various instances of time.

$$\begin{aligned} K_{pj,\min} &\leq K_{pj} \leq K_{pj,\max} \\ K_{dj,\min} &\leq K_{dj} \leq K_{dj,\max} \quad j = 7, 8, \dots, 12 \\ K_{ij,\min} &\leq K_{ij} \leq K_{ij,\max} \end{aligned} \quad (22)$$

Eq. (22) restricts the gains of the PID controller to minimize the error in the angular displacement of various joints of the stand leg at various instances of time. The design of PID controller is a highly non-linear problem that involves modelling the dynamic behaviour of the biped robot. Therefore, the authors of the manuscript is decided to use evolutionary algorithms to tackle the problem of tuning the gains of the PID controller.

## 4. Proposed algorithms

To solve the aforementioned optimization problem related to the tuning of the torque based PID controller, two non-traditional optimization algorithms, namely modified chaotic invasive weed optimization and differential evolution algorithms are used. The explanation related to the

proposed algorithms are given in the subsequent sub-sections.

4.1 Modified chaotic invasive weed optimization algorithm

The Invasive weed optimization (IWO) algorithm is a numerical stochastic optimization algorithm inspired from

the colonizing behavior of weeds introduced by Mehrabain and Lucas [40] in 2006. The algorithm (figure 3) works based on the following principle. The weeds invade the unused resources in the cropping system (that is, field) by means of dispersal and occupy the opportunity spaces between the crops. The invaded weeds (that is, parents) grow to flowering weeds after using the unused resources in the crop and produce new seeds. These seeds are dispersed randomly in a field and grow to a new flowering weed (that

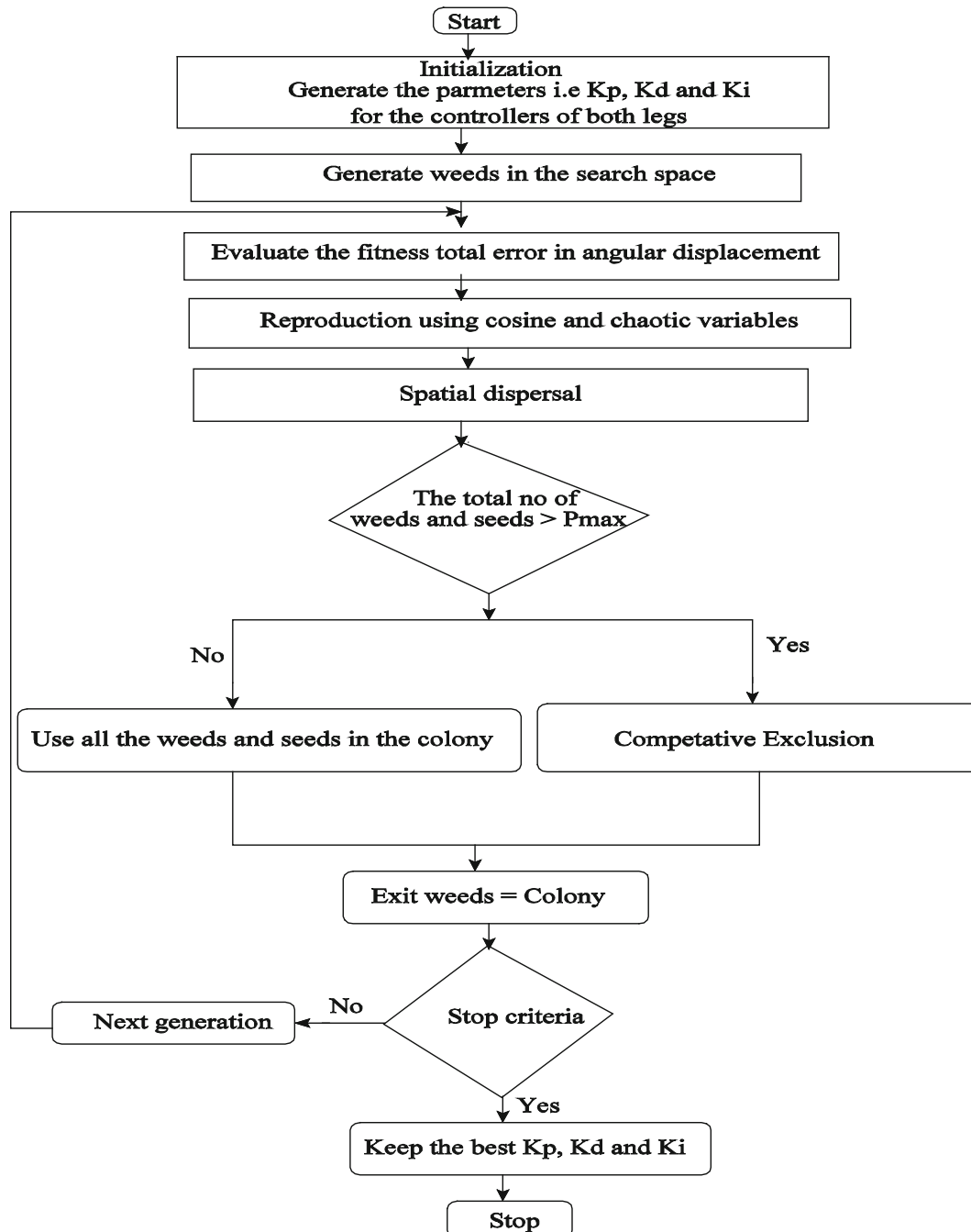
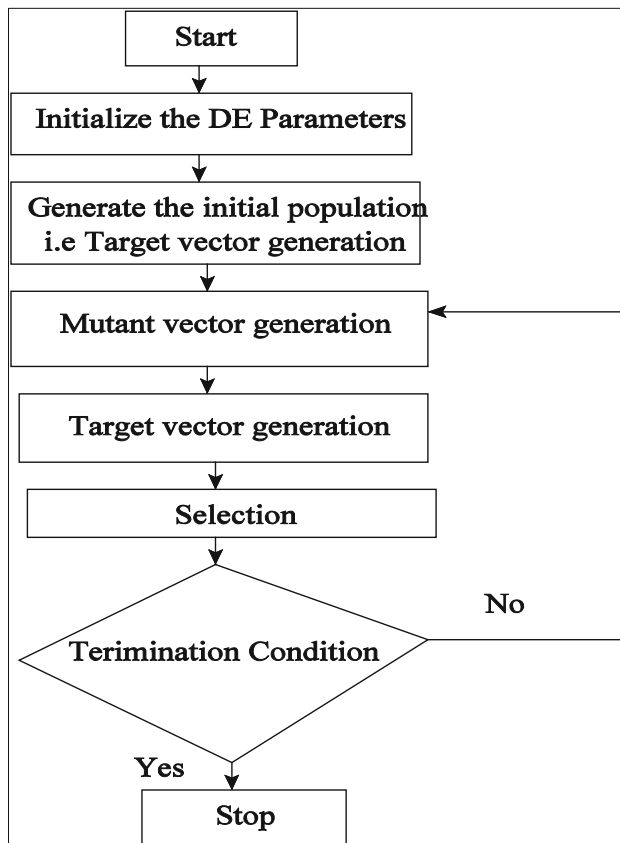


Figure 3. Flow chart showing the operation of MCIWO algorithm.



**Figure 4.** Flow chart showing the operation of DE algorithm.

is, children). The step-by-step procedure used in the implementation of MCIWO is given below.

- Choose the number of parameters of the problem and assign the minimum and maximum values of each parameter. A finite number of seeds are dispersed randomly in an N-dimensional solution space and each seed occupies a random position called initial solution.
- Based on the usage of unused resources in the field, each initial seed grows to a flowering plant and produce new seeds on the fitness value of each plant. The number of seeds ( $S$ ) produced by each plant in

reproduction process is calculated by using the equation (23). The goodness of this algorithm is that it allows all generated weeds to contribute towards reproduction process. This is advantageous under certain situations, that the weed with worst fitness value also gives some useful information to participate through the evolutionary process.

$$S = \text{Floor} \left[ S_{\min} + \frac{f - f_{\min}}{f_{\max} - f_{\min}} \times S_{\max} \right] \quad (23)$$

where  $S_{\min}$  and  $S_{\max}$  are the minimum and maximum production by each plant, respectively and  $f_{\max}$  and  $f_{\min}$  indicate the maximum and minimum fitness value in the colony, respectively.

- The newly produced seeds are scattered randomly after following normal distribution over the entire search space. The mean of normal distribution is equal to the position of the parent plants and the standard deviation (SD) is determined by following the below expression.

$$\sigma_{Gen} = \frac{(Gen_{\max} - Gen)^n}{(Gen_{\max})^n} \times (\sigma_{initial} - \sigma_{final}) + \sigma_{final} \quad (24)$$

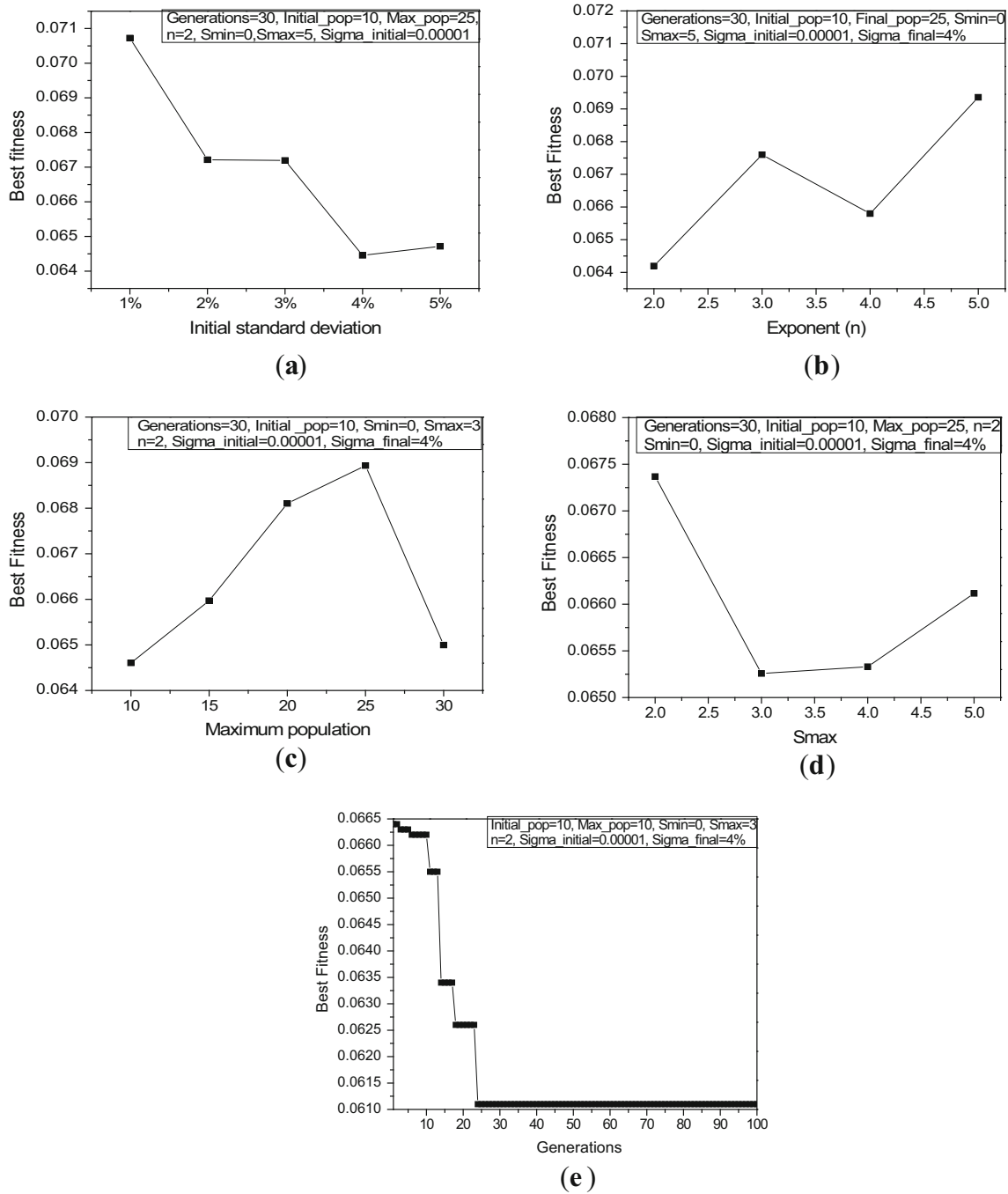
where  $Gen_{\max}$  is the maximum number of generations,  $n$  represents the modulation index i.e., real number and  $\sigma_{Initial}$  and  $\sigma_{Final}$  denote the initial and final value of the standard distribution, respectively (figure 3).

Normal IWO algorithm confirms that all possible candidate solutions would contribute in the reproduction process. But, in most of the meta-heuristic algorithms, less fit individuals are not allowed to form the new off spring. Moreover, IWO algorithm is straight forward and it takes less computational time compared to other methods. On the other hand, some problems require more number of seeds to explore outsized search space. This leads to a drastic increase in the computational time. In order to overcome the limitations of IWO algorithm, in the present research work the authors incorporated two new terms, namely cosine and chaotic variables [41–44] to improve the performance of the IWO algorithm. The main idea of adding cosine variable is to increase the search space for exploring the better solutions. Moreover, the introduction of chaotic

**Table 3.** 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





**Figure 5.** Results of parametric study to determine the parameters of MCIWO: (a) Initial standard deviation vs best fitness, (b) exponent vs best fitness, (c) maximum number of population vs best fitness, (d) maximum number of seeds vs best fitness and (e) generations vs best fitness.

variable helps in reducing the chances of the solution to trap in the local optimum. Due to the greater scanning and search capabilities of the modified algorithm, the maximum number of seeds that participate in the search process are more, and the computational cost of the overall algorithm is less. The chaotic variable that is used in this algorithm is

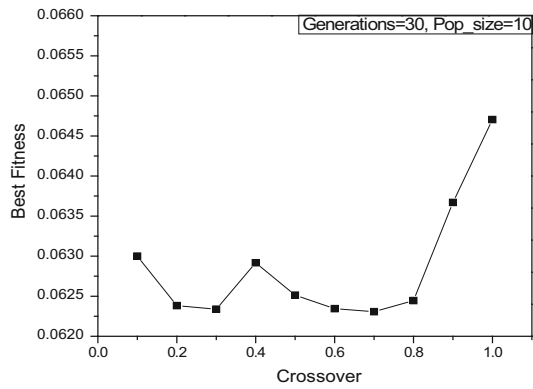
derived from the Chebyshev map, and the expression is given in the following equation.

$$X_{k+1} = \cos(k \cos^{-1}(X_k)) \tag{25}$$

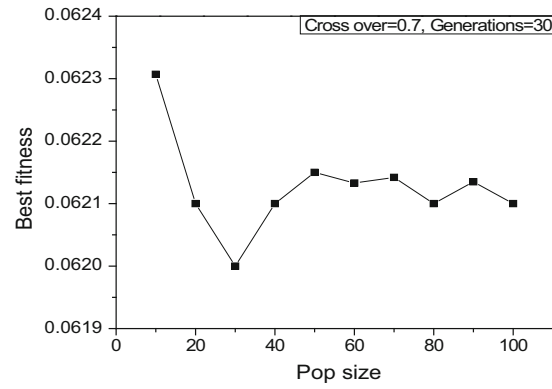
The main goal of the present modification is not only to get the global optimum solution, but also to obtain the best

**Table 4.** Optimal tuning parameters

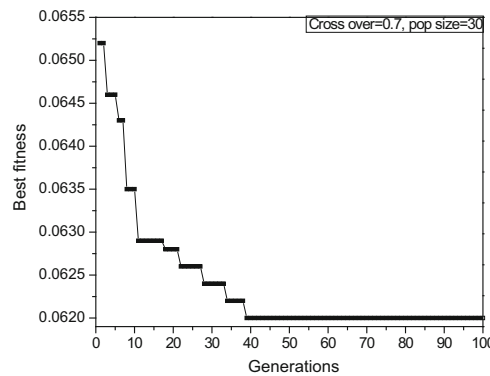
Joint	Manual tuning			MCIWO based tuning			DE based tuning		
	K <sub>p</sub>	K <sub>d</sub>	K <sub>i</sub>	K <sub>p</sub>	K <sub>d</sub>	K <sub>i</sub>	K <sub>p</sub>	K <sub>d</sub>	K <sub>i</sub>
1	800	300	2000	797.04	257.89	1927.42	755.86	325.96	1800.00
2	900	300	1200	816.80	282.93	1148.77	800.00	350.00	1197.67
3	850	400	700	835.76	350.00	797.14	950.00	350.02	800.00
4	800	400	600	899.99	350.00	632.45	900.00	350.00	699.68
5	1000	300	4000	982.23	350.00	3917.47	900.02	349.99	3800.03
6	600	300	700	500.01	326.69	637.05	507.60	349.65	603.38
7	1000	300	4000	924.64	328.31	3866.66	1077.64	278.24	3800.00
8	800	400	3000	723.84	417.74	3178.15	846.07	350.00	3164.22
9	1000	200	4000	1037.12	161.04	3863.30	900.92	150.02	3800.00
10	800	300	1200	803.78	342.27	1107.32	718.72	349.96	1249.43
11	800	300	3200	703.73	266.52	3252.65	700.90	350.00	3384.95
12	2000	600	3000	2120.14	613.74	2963.58	1849.20	550.07	3200.00



(a)



(b)

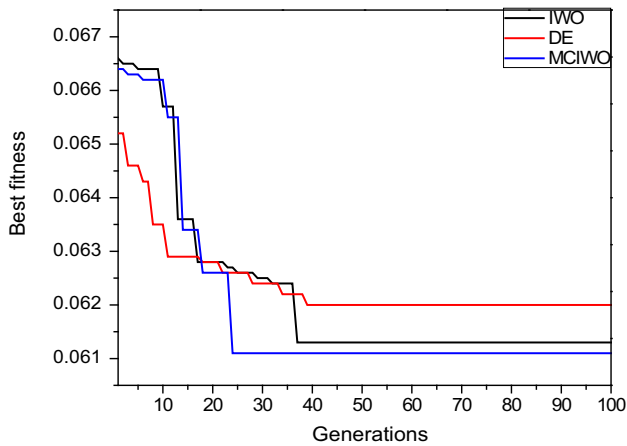


(c)

**Figure 6.** Results of parametric study to determine the parameters of DE: **(a)** Crossover vs best fitness, **(b)** pop size vs best fitness, **(c)** generations vs best fitness.

possible outcome after utilizing minimum resources. After introducing the cosine term, the equation (24) will be modified to the following form.

$$\sigma_{Gen} = \frac{(Gen_{max} - Gen)^n}{(Gen_{max})^n} \times |\cos(Gen)| \times (\sigma_{initial} - \sigma_{final}) + \sigma_{final} \tag{26}$$

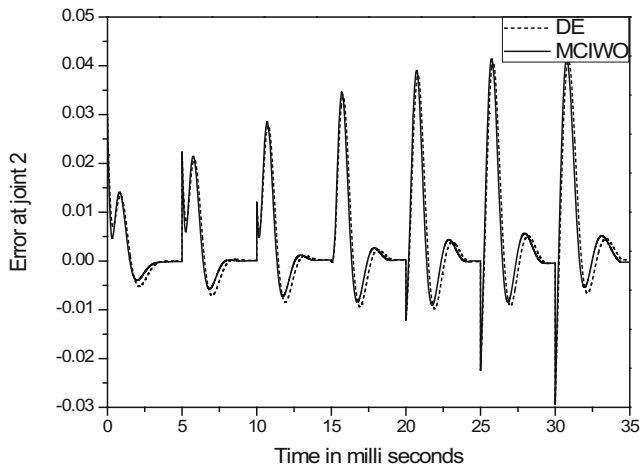


**Figure 7.** Results showing the convergence of the fitness comparison with different optimization algorithms.

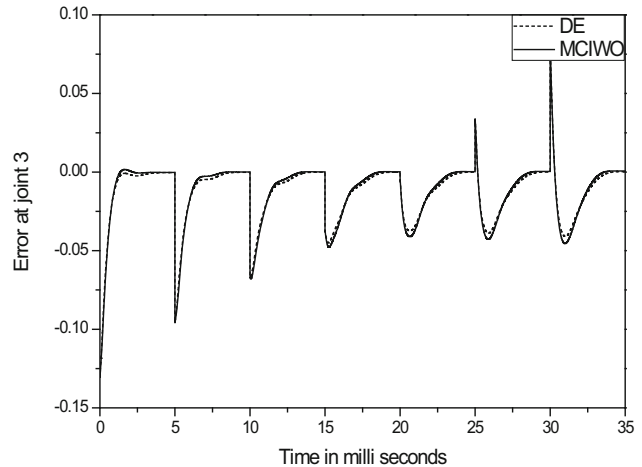
After the new seeds occupy their position in the search area, they grow and become flowering plants and they are also assigned with a rank along with their parent plants. Then these plants are arranged from the lower rank to higher rank based on the fitness value of each plant. The lower ranked plants are eliminated, once the colony reaches its maximum number of plants i.e.,  $P_{max}$ . This procedure continues until the maximum number of iterations are reached.

#### 4.2 Differential evaluation algorithm

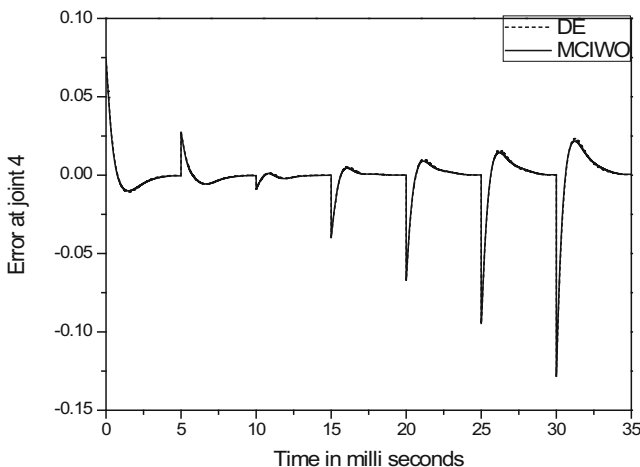
DE is a vector based stochastic optimization algorithm developed by Storn and Price in 1995 [45]. The flow chart shown in figure 4 explains the operation of DE algorithm. In DE, the populations (NP) are vectors in a D-



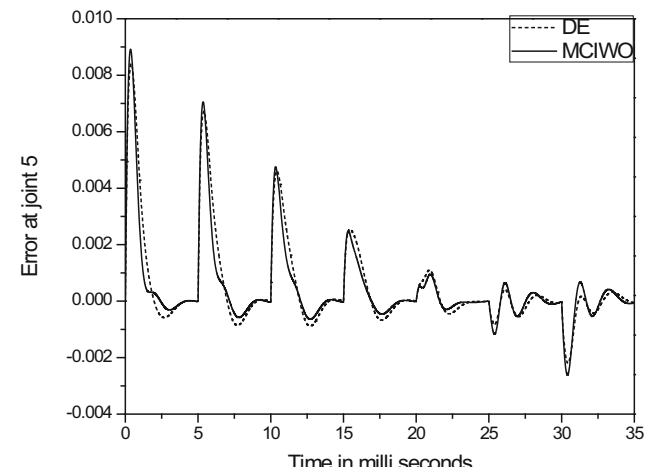
(a)



(b)

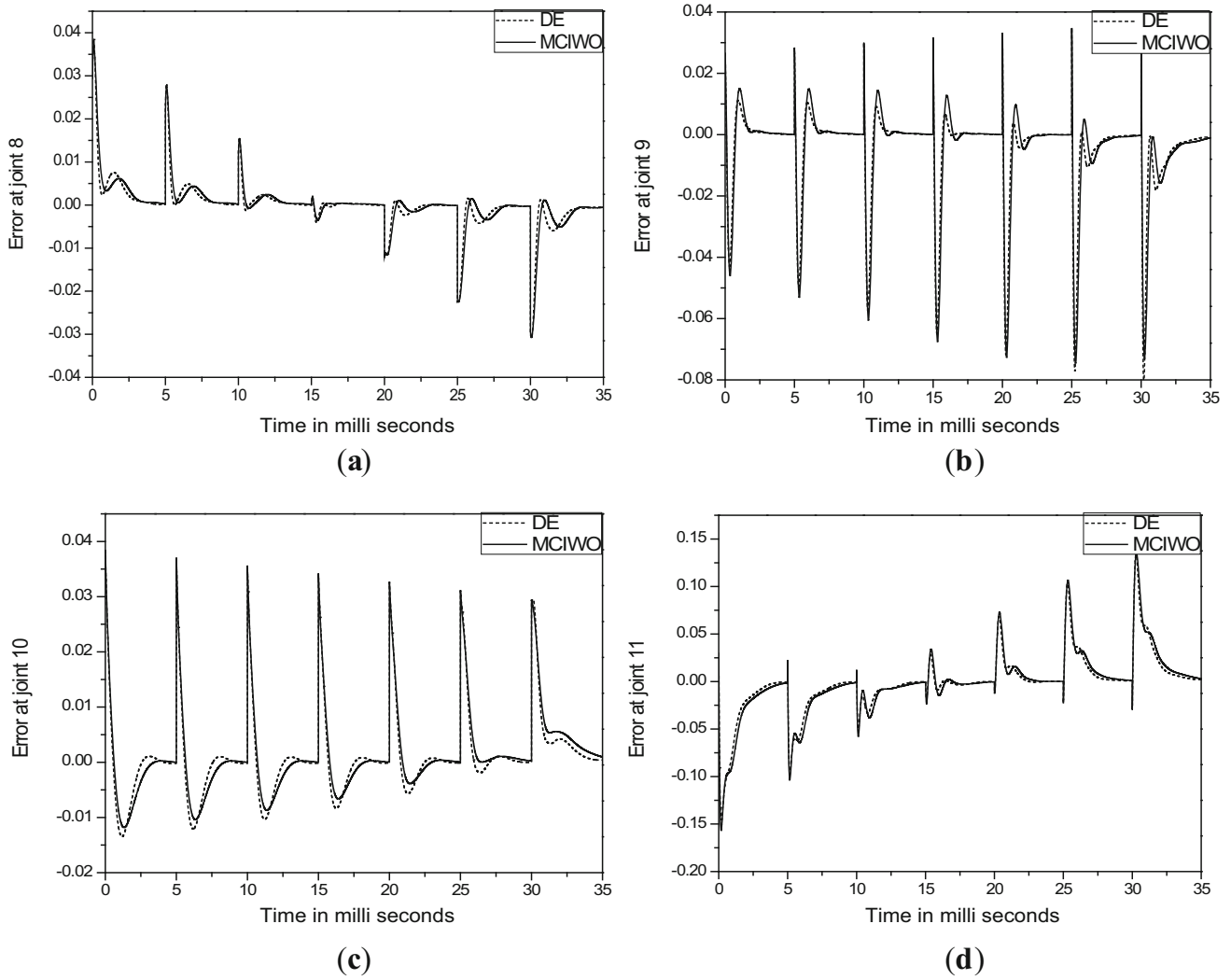


(c)



(d)

**Figure 8.** Results showing the variation of joint error of the swing leg: (a) Joint 2, (b) Joint 3, (c) Joint 4 and (d) Joint 5.



**Figure 9.** Results showing the variation of joint error of the swing leg: (a) Joint 8, (b) Joint 9, (c) Joint 10 and (d) Joint 11.

dimensional search space. Some crucial control parameters namely scaling factor ( $F$ ) and crossover rate ( $CR$ ) are used to improve the convergence performance of DE. The initial population in the search space is generated with the help of the following equation.

$$x_{j,i,0} = x_{j,\min} + rand_{ij}[0, 1] * (x_{j,\max} - x_{j,\min}) \quad (27)$$

Once the population is initialized, DE employs the mutation operator to bring changes in the vectors that represent parameter space. It is to be noted that the parent vector in the current generation ( $X_{i,G}$ ) and vector obtained after mutation operation ( $V_{i,G}$ ) are called target vector and mutant vector, respectively. In each generation, the target vector is associated with the mutant vector  $V_{i,G} = \{v_{1,i,G}, v_{2,i,G}, v_{3,i,G}, v_{4,i,G}, \dots, v_{D,i,G}\}$  and can be generated after using a certain mutation strategy. In the present paper, *DE/rand/1* strategy has been used to implement the mutation strategy.

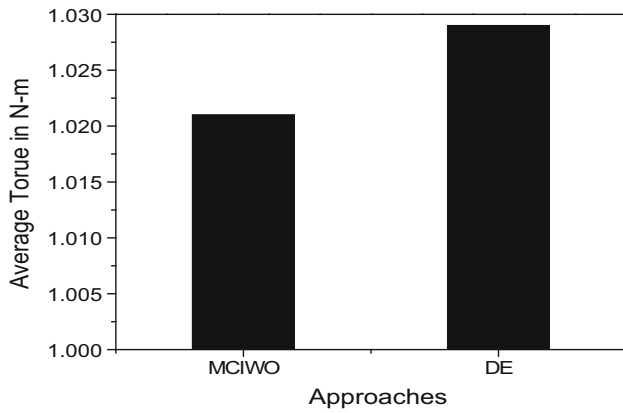
$$\text{“DE/rand/1”} : V_{i,G} = X_{r_1,G} + F * (X_{r_2,G} - X_{r_3,G}) \quad (28)$$

The parameters  $r_1, r_2$  and  $r_3$  indicates three integers generated randomly once for each mutant vector in the range of  $[1, NP]$  and the term  $F$  represent the scaling factor. Once the mutation is over, crossover operation is implemented on the mutant vector to generate the trail vector:  $U_{i,G} = [u_{i,G}^1, u_{i,G}^2, u_{i,G}^3 \dots u_{i,G}^D]$ . In the present problem, binomial crossover is used to perform the operation and is given below:

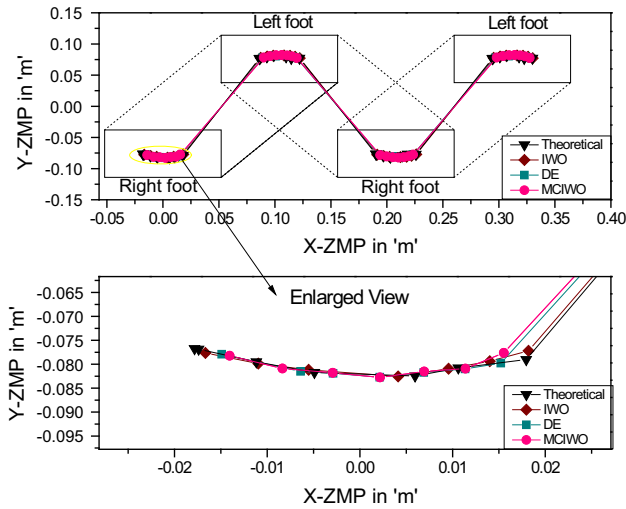
$$u_{i,G}^j = \begin{cases} v_{i,G}^j, & \text{if } rand(0, 1) \leq CR \text{ or } (j = j_{rand}) \\ x_{i,G}^j, & \text{if } rand(0, 1) > CR \end{cases} \quad (29)$$

$j = 1, 2, 3, \dots, D$

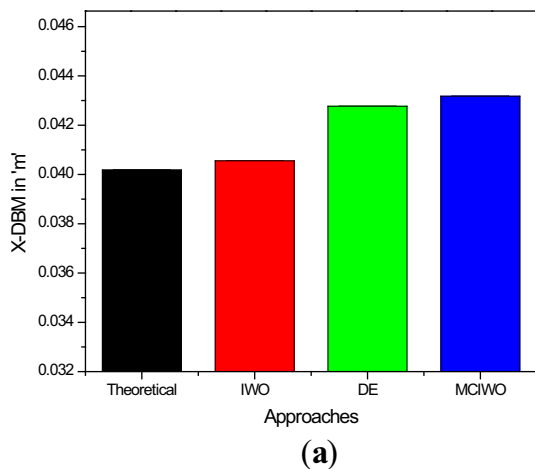
where  $CR$  represents the crossover rate and is a constant within the range of  $[0, 1]$ , which controls the parameter values chosen from the mutant vector. The term  $j_{rand}$  is



**Figure 10.** Comparison of average torque required to complete one cycle: (a) MCIWO, (b) DE tuned based PID controllers.



**Figure 11.** Schematic diagram showing the variation of ZMP in X and Y-directions.



integer randomly chosen in the range [1, D]. The binomial crossover operator selects the  $j^{\text{th}}$  parameter of the mutant vector ( $V_{i,G}$ ) to the corresponding element in the trail vector  $U_{i,G}$  if  $\text{rand}(0, 1) \leq CR$  or ( $j = j_{\text{rand}}$ ). Otherwise, it is copied from the corresponding target vector  $X_{i,G}$ .

If the parameters generated from the new trail vector is not within the corresponding lower and upper limits of the parameters, re-initialization has to be done within the specified range of the parameters and their objective function is to be evaluated. The functional value of each trail vector  $f(U_{i,G})$  and the corresponding functional value of the target vector  $f(X_{i,G})$  of the current population are compared. If the functional value of the trail vector is less than or equal to the objective function value of the corresponding target vector, the trail vector will replace the target vector and enter in to the next generation. Otherwise the same target vector will be copied to the next generation.

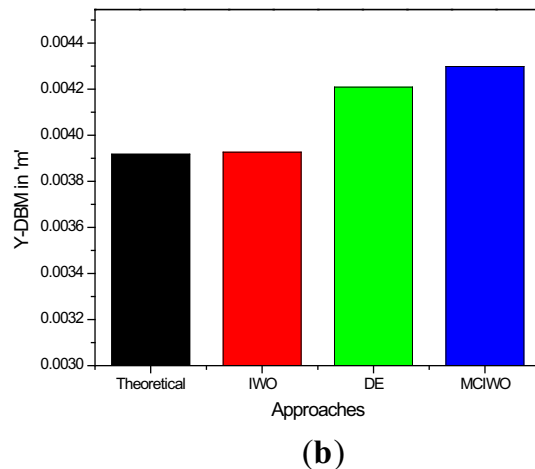
$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \quad (30)$$

### 5. Results and discussions

In the present study, two non-traditional optimization algorithms, namely MCIWO and DE are used to tune the gains of the torque based PID controller. Table 3 shows the parameters of the biped robot used in the study to carry out the simulations.

#### 5.1 MCIWO based PID controller

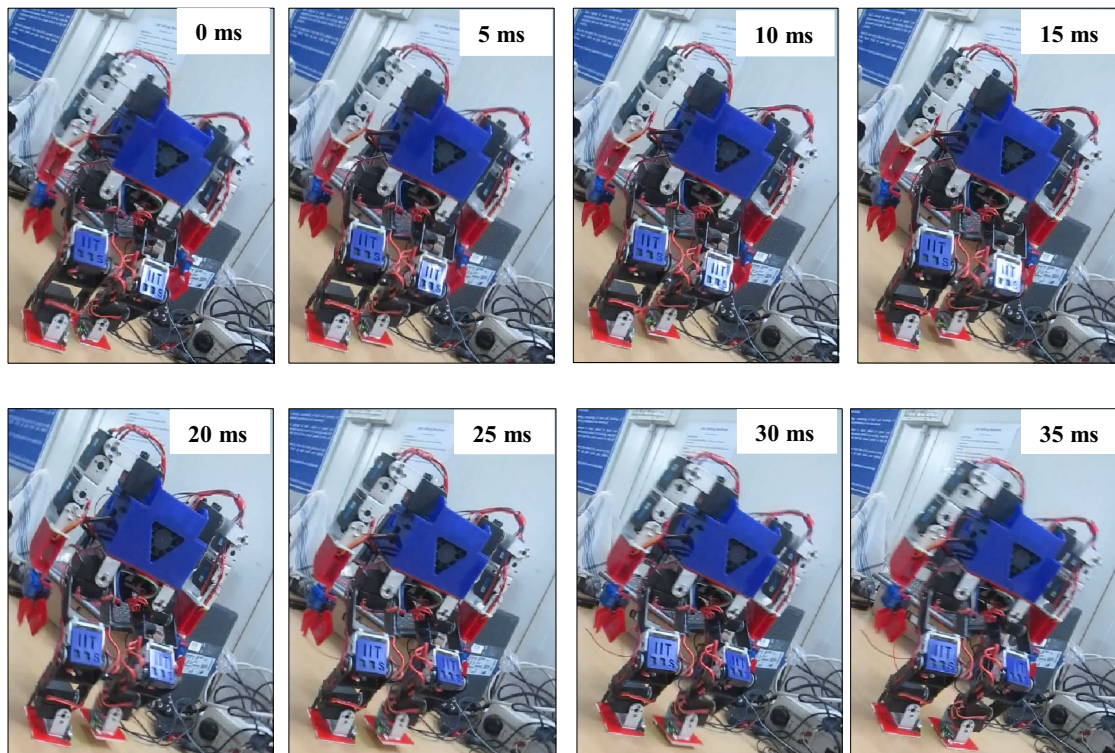
As the performance of the MCIWO algorithm depends on its parameters, namely initial standard deviation, exponent



**Figure 12.** Schematic diagram showing the variation of DBM: (a) X-DBM and (b) Y-DBM.

**Table 5.** Results of robustness test for the developed optimal control algorithms

Percentage Change in inputs	Percentage change in DBM			
	MCIWO Algorithm		DE Algorithm	
	X-DBM	Y-DBM	X-DBM	Y-DBM
- 4	0.0013	- 0.0063	0.0014	- 0.0054
- 3	0.0009	- 0.0047	0.0010	- 0.0040
- 2	0.0006	- 0.0032	0.0007	- 0.0027
- 1	0.0004	- 0.0016	0.0002	- 0.0014
1	- 0.0002	0.0018	- 0.0005	0.0018
2	- 0.0005	0.0039	- 0.0008	0.0034
3	- 0.01	0.0076	- 0.0012	0.0052
4	- 0.0097	0.0097	- 0.0015	0.0067

**Figure 13.** Results showing the swing moment of the left leg.

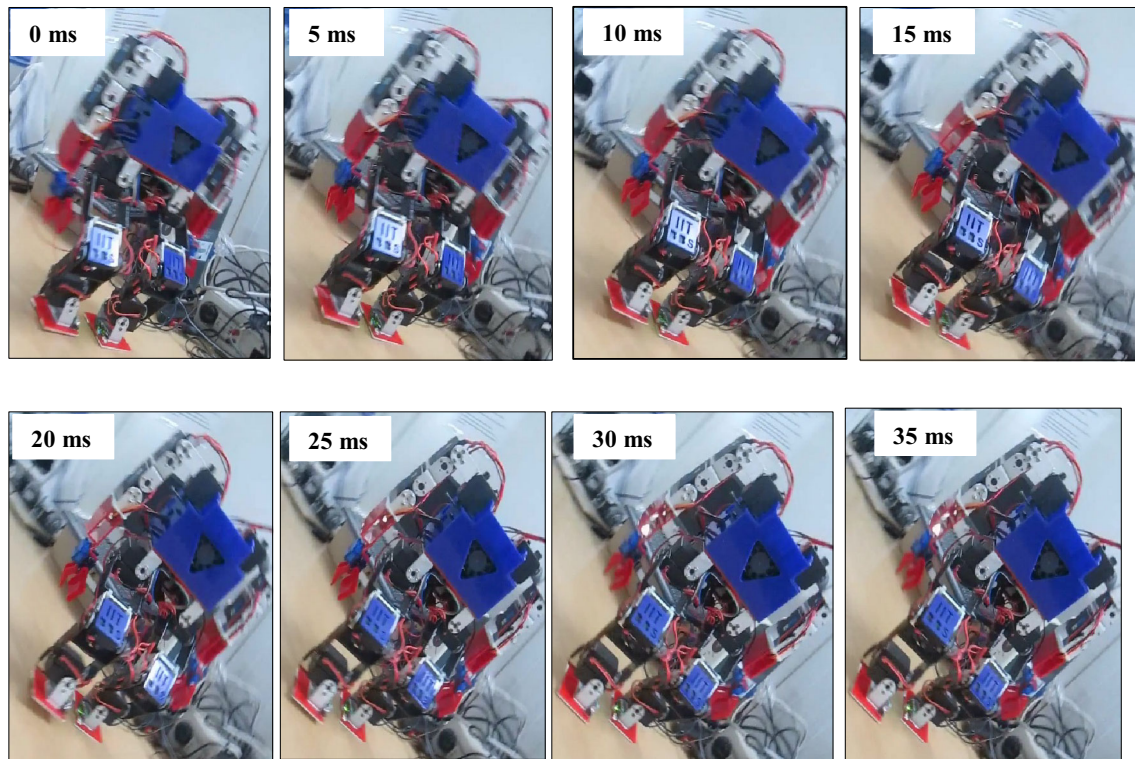
value, maximum population size, maximum number of seeds and number of generations, a detailed parametric study has been conducted to determine the optimal parameters. It is to be noted that the final standard deviation, minimum number of seeds and initial population size are kept equal to 0.00001, 0 and 10, respectively. The results related to the parametric study are shown in figure 5.

The optimal values obtained for the initial standard deviation, exponent value, maximum population size, maximum number of seeds and number of generations are found to be equal to 4%, 2, 10, 3 and 100, respectively.

These optimal parameters of the MCIWO are responsible for deriving the optimal gains of the PID controller i.e.,  $K_p$ ,  $K_d$  and  $K_i$  are given in table 4.

### 5.2 DE based PID controller

Here also, a parametric study has been conducted to determine the optimal parameters, namely cross over rate, population size and generations of the DE algorithm. It is to be noted that the minimum and maximum scaling factor are kept equal to 0.2 and 0.8, respectively. The results of the



**Figure 14.** Results showing the swing moment of the right leg.

DE-parametric study are shown in figure 6. The optimal values of DE parameters are found to be as follows:

- (i) Cross over rate = 0.7
- (ii) Population size = 30
- (iii) Maximum no. of generations = 100

The above optimal values of the DE algorithm are useful to tune the gains of the PID controller. The optimal gains obtained by using DE are given in table 4.

Further, figure 7 shows the final converge graph for IWO, DE and MCIWO algorithms. It can be observed that MCIWO has shown a better performance in convergence when compared with the IWO algorithm. Therefore, only MCIWO along with DE are used in the further study to compare the performances of the torque based PID controllers used by the biped robot.

### 5.3 Comparative study

Once the optimal PID controllers are evolved, a comparative study has been conducted between MCIWO and DE-tuned controllers in terms of variation of joint error, torque required, variation of ZMP and DBM of the biped robot. Table 4 shows the comparison of optimal gains (that is,  $K_p$ ,  $K_d$  and  $K_i$ ) obtained using the manual, MCIWO and DE based tuning of the PID controller.

Further, the performances of the optimal MCIWO and DE-based PID controllers are tested in computer simulations. The graphs 8 (a), (b), (c) and (d) show the variation of error in angular position for the swing leg joints. Initially, the magnitude of the error is very high and it slowly reached to the study state. Moreover, it has also been observed that both the MCIWO and DE-based PID controllers are found to reach the steady state error by exhibiting a similar trend. Further, it has also been observed that the MCIWO tuned PID controller converges quickly when compared with DE tuned PID controller. It may be due to the enhanced search space created in MCIWO algorithm due to the introduction of cosine and chaotic terms, when compared with the DE algorithm. Similarly, figures 9 (a), (b), (c) and (d) show the error in angular position of the stand leg of the biped robot, and both the controllers have shown the similar trend as explained for the joints of the swing leg.

In addition to the above result, a comparative study has also been conducted for the aforementioned torque based PID controllers in terms of average torque required by the biped robot to perform the walk on the flat surface (ref. figure 10). It can be observed that the MCIWO tuned PID controller required less torque to execute the walk on the flat surface by the biped robot when compared with the DE tuned PID controller.

Further, a study has also been conducted to see the variation of actual ZMP (that is, after controller action) in X and

Y-directions with that of the theoretical ZMP (that is, obtained by gait generation algorithm) in X and Y-directions. It can be observed that the ZMP trajectories (refer to enlarged view of figure 11) obtained by MCIWO-based PID controller is seen to be more close to the center of the foot when compared with the trajectory obtained by IWO and DE based PID controllers. It may be due to the reason that in the case of controllers, they might have introduced a correction for the angular position of the link at the end of each interval that pushes the position of ZMP towards the end of the foot support polygon. Finally, the gait obtained is found to be balanced in nature as the ZMP is lying inside the foot support polygon.

Along with ZMP trajectory, a comparative study has also been conducted on DBM of the biped robot in both X- and Y-directions (ref. to figure 12). It can be observed that MCIWO based PID controller has produced more dynamically balanced gaits when compared with the theoretical, IWO and DE based PID controllers.

#### 5.4 Robustness test

A robustness test has been conducted to study the performance of the controller after providing some variation in the controller parameter. Table 5 shows the results related to the robustness test conducted by using the aforementioned algorithms. It has been observed that a maximum of -0.01% and 0.0097% variation in the values of DBM is obtained for MCIWO and DE-based PID controllers, respectively for a  $\pm 4\%$  variation in the values of the gains of the controller. From this test, it can be observed that both the controllers are robust enough to accommodate small change in the gains of the controller for generating the dynamically balanced gaits of the biped robot.

#### 5.5 Experiments with real robot

From the simulations, it has been observed that the MCIWO tuned PID controller is found to perform better than DE tuned PID controller. Then the gait angles obtained using MCIWO tuned torque based PID controller is fed to the real robot. Figures 13 and 14 show the results of swing moment of left and right legs, respectively of the biped robot while moving on the flat surface. From the images it can be observed that the biped robot has successfully walked on the flat surface.

## 6. Conclusions

An attempt has been made to develop the torque based PID controller for the biped robot while moving on the flat surface. Two non-traditional optimization algorithms, namely MCIWO and DE algorithms are used to tune the gains of the controller. The simulation results show that

both the developed controllers are able to generate the dynamically balance gaits for the biped robot on the flat terrain. It has been observed that the MCIWO-based PID controller converged quickly and average torque required is less when compared with the DE-based controller. It is interesting to note that the MCIWO-based algorithm has developed more dynamically balanced gaits than the DE and manual tuning methods. It may be due to the reason that in MCIWO algorithm, the search space is more enhanced in the error domain and the chances of trapping the solution in the local minima is less, when compared with DE algorithm. Further, the algorithms are tested in computer simulations and found to perform in a satisfactory manner. Moreover, the gait angles obtained using the MCIWO tuned PID controller are fed to the real robot and found that the robot has successfully negotiated the terrain.

## 7. Scope for future work

In this manuscript, the performances of the developed controllers are tested only on flat surface. However, it will be more interesting to test the aforementioned controllers on stair case and slope (ascending and descending) surfaces. Further, an adaptive algorithm i.e., MCIWO-NN, PSO-NN algorithms and recently developed chemical reaction and firework optimization algorithms may also be used in future to solve the said problems. At present the authors are working on these issues.

## References

- [1] Juricic D and Vukobratovic M 1972 Mathematical modeling of biped walking systems. *ASME Publication*. 72
- [2] Furusho J and Masubuchi M 1987 A theoretically motivated reduced order model for the control of dynamic biped locomotion, *Trans. ASME J. Dyn. Syst. Meas. Control*. 109: 155–163
- [3] Seo Y J and Yoon Y S 1995 Design of a robust dynamic gait of the biped using the concept of dynamic stability margin. *Robotica*. 13: 461–468
- [4] Lum H K, Zribi M and Soh Y C 1999 Planning and control of a biped robot. *Int. J. Eng. Sci.* 37: 1319–1349
- [5] Kim J Y, Park I W, Lee J, Kim M S, Cho B K and Oh J H 2005 System design and dynamically walking of humanoid robot KHR-2. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1443–1448
- [6] Pandu R V, Sahu S K and Pratihar D K 2007 Dynamically balanced ascending and descending gaits of a two-legged robot. *Int. J. Humanoid Robot.* 4: 717–751
- [7] Pandu R V and Pratihar D K 2011 Balanced gait generations of a two legged robot on sloping surface, *Sadhana- Acad. Proc. Eng. Sci.* 36: 525–550
- [8] Hernández-Santos C, Rodríguez-Leal E, Soto R and Gordillo J L 2012 Kinematics and dynamics of a new 16 DOF



- humanoid biped robot with active toe joint. *Int. J. Adv. Robot. Syst.* 9: 1–12
- [9] Hayder F N Al S, Burkhard J C, Bram V and Wen H Z 2015 zero-moment point-based biped robot with different walking patterns. *Int. J. Intell. Syst. Appl.* 7: 31–41
- [10] Dusko K and Miomir V C 2003 Survey of intelligent control techniques for humanoid robots. *J. Intell. Robot. Syst.* 37: 117–141
- [11] Sujoy B, Manoj K T and Felix T S C 2017 Predicting the consumer's purchase intention of durable goods: An attribute-level analysis. *J. Bus. Res.* (In press)
- [12] Patrizio T 1991 Adaptive PD controller for ROBOT manipulators. *IEEE Trans. Robot. Autom.* 7: 565–570
- [13] Rafael K 1995 A tuning procedure for stable PID control of robot manipulators. *Robotica.* 13: 141–148.
- [14] Khoury G M, Saaad M, Kanaan H Y and Asmar C 2004 Fuzzy PID control of a five DOF robot arm. *J. Intell. Robot. Syst.* 40: 299–320
- [15] Ravi K M and Pandu R V 2015 Design of PID Controllers for 4-DOF Planar and Spatial Manipulators. *IEEE Conference on Robotics, Automation, Control and Embedded Systems*, pp. 1–6
- [16] Hassan B K 2002 The SOF-PID controller for the control of a MIMO robot arm. *IEEE Trans. Fuzzy Syst.* 10: 523–532
- [17] Aulia el H, Hilwadi H and Estiko R 2013 Application of reinforcement learning on self-tuning PID controller for soccer robot multi-agent system. In: *IEEE Conference on Rural Information & Communication Technology and Electric-Vehicle Technology*, pp. 1–6
- [18] Jeong J K, Jun W L and Ju J L 2009 Central pattern generator parameter search for a biped walking robot using nonparametric estimation based particle swarm optimization. *Int. J. Control, Autom. Syst.* 7: 447–457
- [19] Helon V H A and Leandro dos S C 2012 Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator. *Expert Syst. Appl.* 39: 8968–8974
- [20] MOHD S S, HISHAMUDDIN J, INTAN Z M D 2012 Implementation of PID controller tuning using differential evolution and genetic algorithms. *Int. J. Innov. Comput. Inf. Control.* 8: 7761–7779
- [21] Ravi K M, Sai M K and Pandu R V 2015 *Optimization of PID Controller Parameters for 3-DOF Planar Manipulator Using GA and PSO*. *New Developments in Expert Systems Research*, Nova Science publications, Chapter 3, 67–88
- [22] Geetha M, Manikandhan P and Jovitha Jerome 2014 Soft computing techniques based optimal tuning of virtual feedback PID controller for chemical tank reactor. In: *IEEE Congress on Evolutionary Computation*, pp. 1922–1928
- [23] Belkadi A, Oulhadj H, Touati Y, Safdar A K and Daachi B 2017 On the robust PID adaptive controller for exoskeletons: A particle swarm optimization based approach. *Appl. Soft Comput.* 60: 87–100
- [24] Joel P P, Jose P P, Rogelio S, Angel F, Francisco R and Jose L M 2012 Trajectory tracking error using PID control law for two-link robot manipulator via adaptive neural networks. In: *The 2012 Iberoamerican Conference on Electrical Engineering and Computer Science*, Elsevier Procedia Technology, pp. 139–146
- [25] Dusan F, Iztok F Jr, Iztok F and Riko S 2016 Parameter tuning of PID controller with reactive nature inspired algorithms. *Robot. Auton. Syst.* 84: 64–7
- [26] Mehran R, Ahmad G and Mir M E 2016 A novel adaptive neural network integral sliding-mode control of a biped robot using bat algorithm. *J. Vib. Control.* 1–16
- [27] Mahmoodabadi J M, Abedzadeh M R and Taherkhorsandi M 2017 An optimal adaptive robust PID controller subject to fuzzy rules and sliding modes for MIMO uncertain chaotic systems. *Appl. Soft Comput.* 52:1191–1199
- [28] Navid R and Mohsen K 2015 A new design for PID controller by considering the operating points changes in hydro-turbine connected to the equivalent network by using Invasive Weed Optimization (IWO) Algorithm. *Int. J. Inf. Secur. Syst. Manag.* 4: 468–475
- [29] Abedinia O, Foroud A A, Amjady N and Shayanfar H A 2012 Modified invasive weed optimization based on fuzzy PSS in multi-machine power system. In: *International Conference on Artificial Intelligence*, pp. 73–79
- [30] Latha K and Rajinikanth V 2012 2DOF PID Controller Tuning for Unstable Systems Using Bacterial Foraging Algorithm. *SEMCCO, LNCS, Springer*, 7677: 519–527
- [31] Nikhileshwar P A and Amit G 2013 PID Controller Tuning Using Soft Computing Techniques. In: *Proceedings of All India Seminar on Biomedical Engineering 2012*, pp. 277–284
- [32] Rosy P, Santosh K M, Jatin K P and Bibhuti B P 2018 Antlion optimizer tuned PID controller based on Bode Ideal Transfer Function for Automobile Cruise Control System. *J. Ind. Inf. Integr.* 9: 45–52
- [33] Naidu Y R and Ojha A K 2015 A hybrid version of invasive weed optimization with quadratic approximation. *Soft Comput.*, 19: 3581–3598
- [34] Shaya K and Ahmed A K 2010 Invasive weed optimization and its features in electromagnetics. *IEEE Trans. Antennas Propag.* 58: 1269–1278
- [35] Pappula L and Ghosh D 2013 Large array synthesis using invasive weed optimization. In: *Proceedings of the IEEE International conference on microwave and photonics*, pp. 1–6
- [36] Mogale D G, Kumar S K, Fausto P G and Manoj K T 2017 Bulk wheat transportation and storage problem of public distribution system. *Comput. Ind. Eng.* 104: 80–97
- [37] Vijay K, Jitender K C and Dinesh K 2015 Optimal choice of parameters for fireworks algorithm. *Procedia Comput. Sci.* 70: 334–340
- [38] Mogale D G, Kumar S K and Manoj K T 2018 An MINLP model to support the movement and storage decisions of the Indian food grain supply chain. *Control Eng. Pract.* 70: 98–113
- [39] Lohithaksha M M and Jitesh J T 2017 A combined tactical and operational deterministic food grain transportation model: Particle swarm based optimization approach. *Comput. Ind. Eng.* 110: 30–42
- [40] Mehrabian A R and Lucas C 2006 A novel numerical optimization algorithm inspired from weed colonization. *Ecol. Inform.* 1: 355–366
- [41] Aniruddha B, Siddharth P, Swagatam D and Ajith A 2010 A modified Invasive Weed Optimization algorithm for time-modulated linear antenna array synthesis. *IEEE Congress on Evolutionary Computation*, pp. 1–8
- [42] Roy G G, Das S, Chakraborty P and Suganthan P N 2011 Design of non-uniform circular antenna arrays using a modified invasive weed optimization algorithm. *IEEE Trans. Antennas Propag.* 59: 110–118

- [43] Mohamadreza A and Hamed M 2012 Chaotic invasive weed optimization algorithm with application to parameter estimation of chaotic systems. *Chaos, Solitons Fractals*. 45: 1108–1120
- [44] Mojtaba G, Sahand G, Jamshid A, Mohsen G and Hasan F 2014 Application of chaos-based chaotic invasive weed optimization techniques for environmental OPF problems in the power system. *Elsevier Chaos, Solitons Fractals*. 69: 271–284
- [45] Storn, R. and Price K V 1995 *Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces*. ICSI, Tech. Report: 1–12