Hindawi Publishing Corporation
ISRN Robotics
Volume 2014, Article ID 241767, 16 pages
http://dx.doi.org/10.1155/2014/241767

Research Article

An Improved ZMP-Based CPG Model of Bipedal Robot Walking Searched by SaDE

H. F. Yu, E. H. K. Fung, and X. J. Jing

Department of Mechanical Engineering, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Correspondence should be addressed to H. F. Yu; fai5121988@yahoo.com.hk

Received 31 October 2013; Accepted 23 December 2013; Published 4 March 2014

Academic Editors: D. K. Pratihar and K. Terashima

This paper proposed a method to improve the walking behavior of bipedal robot with adjustable step length. Objectives of this paper are threefold. (1) Genetic Algorithm Optimized Fourier Series Formulation (GAOFSF) is modified to improve its performance. (2) Self-adaptive Differential Evolutionary Algorithm (SaDE) is applied to search feasible walking gait. (3) An efficient method is proposed for adjusting step length based on the modified central pattern generator (CPG) model. The GAOFSF is modified to ensure that trajectories generated are continuous in angular position, velocity, and acceleration. After formulation of the modified CPG model, SaDE is chosen to optimize walking gait (CPG model) due to its superior performance. Through simulation results, dynamic balance of the robot with modified CPG model is better than the original one. In this paper, four adjustable factors ($R_{\text{hs,support}}$, $R_{\text{hs,swing}}$, $R_{\text{ks,support}}$, and $R_{\text{ks,swing}}$) are added to the joint trajectories. Through adjusting these four factors, joint trajectories are changed and hence the step length achieved by the robot. Finally, the relationship between (1) the desired step length and (2) an appropriate set of $R_{\text{hs,support}}$, $R_{\text{hs,swing}}$, $R_{\text{ks,support}}$, and $R_{\text{ks,swing}}$ searched by SaDE is learnt by Fuzzy Inference System (FIS). Desired joint angles can be found without the aid of inverse kinematic model.

1. Introduction

Recently, many approaches have been adopted for generation of bipedal walking gait. Some researches [1–3] adopted a simplified dynamic model to generate walking gait calculated through inverse kinematic model which is complex and hence the computation load is high.

Inspired by neural science, some researchers investigated central pattern generator (CPG). The prime reason for arousing their interest is that CPG models provide several parameters for modulation of locomotion, such as step stride and rhythm, and are suitable to integrate feedback sensors. Hence, a good interaction between the robot and the environment can be achieved [4]. According to Ijspeert [4], CPG becomes more and more popular in robot community. Taga et al. [5] integrated feedbacks with neural oscillators for unpredicted environment. Yang et al. [6], Shafii et al. [7], and Yazdi et al. [8] utilized TFS to formulate ZMP-based CPG model as the basic walking pattern of bipedal robot. Or [9] presented a hybrid CPG-ZMP control system for flexible spine humanoid robot. Aoi and Tsuchiya [10] proposed a locomotion control system based on CPG model for straight and curved walking. Farzenah et al. [11] noted that many researches on CPG model are designed for specific motion only and thus cannot generate arbitrary walking gait, such as changing step length and proposed 31 TS-Fuzzy systems for adjusting speed and step length.

Ijspeert [4] and Gong et al. [12] stated that stochastic population-based optimization algorithms have been chosen to optimize parameters of CPG model in many studies. Genetic Algorithm (GA) [6, 13], Genetic Programming (GP) [14], Particle Swarm Optimization (PSO) [7], and Bee Algorithm [8] are adopted in searching the parameters of CPG model. Besides the above-mentioned techniques, there are still other gradient-free optimization techniques. Storn and Price [15] proposed Differential Evolution (DE) and conducted comparisons with some prominent algorithms, such as Adaptive Simulated Annealing (ASA) and the Breeder Genetic Algorithm (BGA). DE outperforms the above-mentioned prominent algorithms in terms of least number of

generations for finding global minimum [15]. Similar results are reported in the following studies [16–18]. Hegery et al. [16] carried out comparisons between DE and GA on N-Queen and travelling salesman problem and concluded that the performance of DE is better. Tušar and Filipič [17] carried out comparisons between DE-based variants DEMO and basic GA on multiobjective optimization problem and their result showed that DEMO outperforms basic GA. Vesterstrøm and Thomsen [18] noted that DE outperforms PSO and Evolutionary Algorithms (EAs) on majority of numerical benchmark problems. DE consists of population size (NP), scaling factor (F), and crossover rate (CR) which significantly affect the performance of DE [19–22]. Different problems require different parameters and strategies for effective optimization. Even in the same problem, different regions of search space may require different strategies and parameters for better performance [20]. It is time consuming to search the most appropriate strategy and parameters by trial and error. Hence, Omran et al. [21] and Brest et al. [22] have proposed different methods to adjust CR and F. However, appropriate mechanism for choosing suitable strategies is not considered in [21, 22]. Qin et al. [20] proposed Self-adaptive Differential Evolution Algorithm (SaDE) which can adjust CR, F and choose strategy automatically during optimization. SaDE outperforms conventional DE variants and the other adaptive DE, such as SDE [21] and jDE [22], in terms of higher successful rate.

Based on the above-mentioned findings, this paper focuses on (1) CPG model for trajectory generation and (2) providing an efficient method to adjust step length. In this paper, original CPG model proposed by Yang et al. [6] is adopted since it provides a good foundation for the goal stated in this paper. The angular velocity of trajectories generated by GAOFSF [6] is usually discontinuous which has an adverse effect on ZMP. As a result, GAOFSF is modified to ensure that the trajectory generated is continuous in angular position and its first and second derivatives. After formulation of modified CPG model, parameters of CPG model are searched based on kinematic and dynamic constraints. It shows that the problem can be formulated as a multiobjectives and multiconstraints optimization. Gradient-free optimization technique is chosen since a set of parameters is searched in a highly irregular and multidimensional space which cannot be handled by standard gradient-based search method [12]. SaDE is chosen as the method for optimizing the walking gait of robot in this paper because (1) its performance is superior and (2) appropriate strategies and parameters are not chosen manually. Based on [6, 23], step length can be varied by simply changing several adjustable factors of GAOFSF. Look-up table proposed by Yang et al. [6] is not adopted in this paper because (1) a lot of memory is occupied if tremendous data is stored and (2) arbitrary step length within specific range cannot be commanded to the robot. To deal with this problem, four parameters ($R_{\text{hs,support}}$, $R_{\text{hs,swing}}$, $R_{\text{ks,support}}$, and $R_{\text{ks,swing}}$) are added to the modified CPG model and searched by SaD. Four FIS systems are used to learn the relationship between (1) desired step lengths and $R_{\text{hs,support}}$, $R_{\text{hs,swing}}$, $R_{\text{ks,support}}$, and $R_{\text{ks,swing}}$. Then, desired joint angles can be found without the aid of inverse kinematics.

Table 1: Joint number of support leg and swing leg corresponds to different joints.

Joints of leg	Support leg	Swing leg
Ankle joints	1-2	5-6
Knee joint	3	4
Hip joints	4–6	1–3

Table 2: Height of each link in support leg and swing leg.

Link number	Support leg	Swing leg
0	0.026 m	N/A
1	0.04 m	0.028 m
2	0.0645 m	0.0385 m
3	0.0645 m	0.0645 m
4	0.0385 m	0.064 m
5	0.028 m	0.04 m
6	0.096 m	0.026 m

2. Kinematic Model and Dynamic Model of Bipedal Robot

In Figure 1, it shows that the bipedal robot consists of 12 DoF. Each leg has 6 DoF. RL and LL represent right and left legs respectively, while $1, 2, 3, \ldots, 6$ represent joint number. Figure 1 assumes that right leg is the support leg while the left leg is the swing leg. Joint number n of support leg and swing leg corresponds to different joint shown in Table 1. Also, z-axis of local coordinates attached on different joints acts as rotation axis and the direction of rotation is determined by right-hand rule.

Information of physical dimension of bipedal robot is measured and simplified based on a modified Kondo-3HV. The total mass of the bipedal robot is about 1.4 kg. The mass of upper trunk and lower body is about 0.4 kg and 1 kg, respectively. Since the mass of each link in lower body is almost the same, then their masses are simply obtained by $1\,\text{kg}/12 = 0.083\,\text{kg}$. The height of each link is shown in Table 2.

Since Denavit-Hartenberg notation [24, 25] and iterative Newton-Euler dynamic algorithm [26, 27] are maturely developed and commonly used in many studies of bipedal robot, these two methods are adopted to formulate forward kinematic model and inverse dynamic model, respectively, in this paper. Since this paper focuses on bipedal walking on horizontal flat plane in sagittal plane (parallel to the $Y_G - Z_G$ plane of global coordinate), only ZMP_y shown in (1) is considered and acts as an indicator to evaluate the dynamic equilibrium of the bipedal robot. In Figure 2, a local coordinate is attached to the center of support foot to observe the variation of ZMP_y with time.

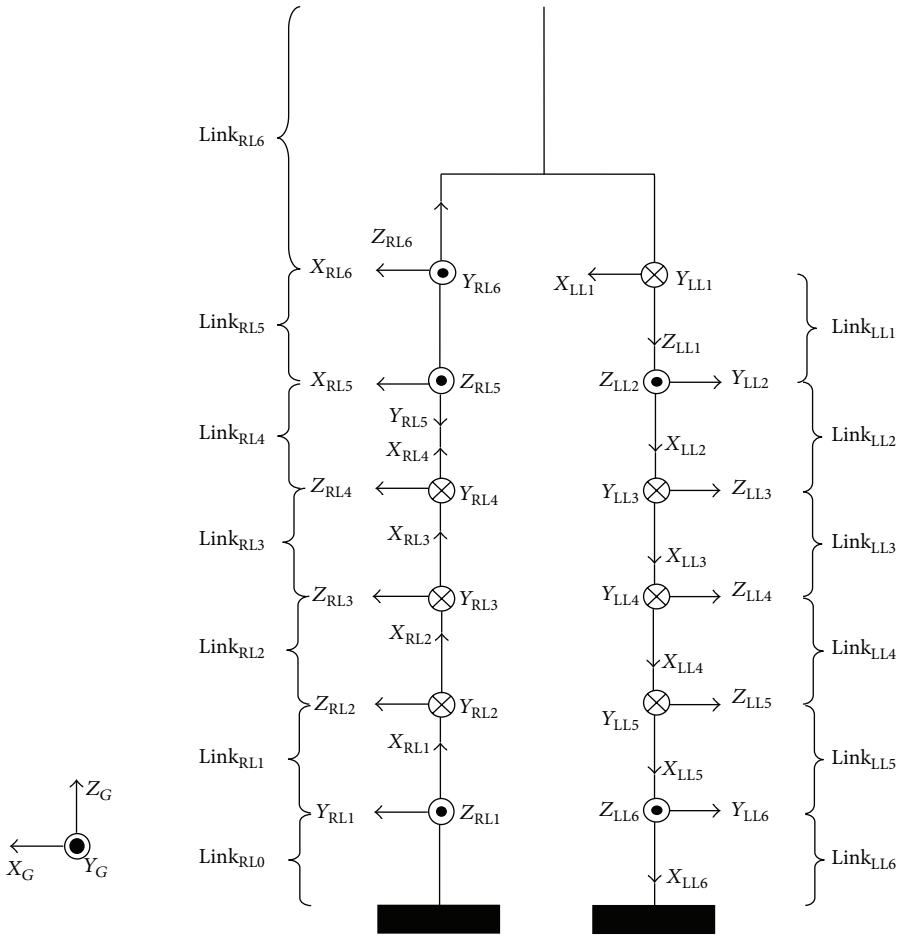

Figure 1: Schematic diagram of bipedal robot.

Consider

$$\mathrm{ZMP}_y = \left(\sum_{i=1}^{6} m_{i,RL} \left(\ddot{z}_{i,RL} + g \right) y_{i,RL} + \sum_{i=1}^{6} m_{i,LL} \left(\ddot{z}_{i,LL} + g \right) y_{i,LL} - \sum_{i=1}^{6} m_{i,RL} \ddot{y}z_{i,RL} - \sum_{i=1}^{6} m_{i,LL} \ddot{y}z_{i,LL} \right) \times \left(\sum_{i=1}^{6} \left(\ddot{z}_{i,RL} + g \right) m_{i,RL} + \sum_{i=1}^{6} \left(\ddot{z}_{i,LL} + g \right) m_{i,LL} \right)^{-1}. \quad (1)$$

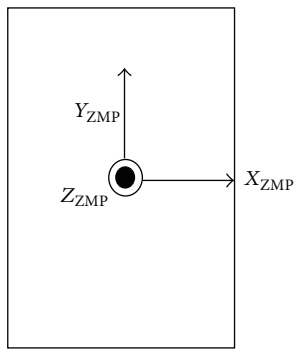

Figure 2: Local coordinate system attached on the foot sole (top view).

3. Formulation of Modified ZMP-Based CPG Model

Hip and knee trajectories of GAOFSF [6] consist of two different sections. For hip joint, two different Truncated Fourier Series (TFS) are used to formulate the upper portion (θ_h^+) and lower portion (θ_h^-) of trajectory. For knee joint, TFS and lock phase are joined together to formulate the whole trajectory. Based on Figure 3, the angular position is observed to be continuous. However, angular velocity of searched trajectory generated by GAOFSF is usually discontinuous (1) at the transition between θ_h^+ and θ_h^- (t_3 and t_6 or t_0) and (2) at the beginning (t_1 or t_4) and the end of lock phase (t_2 or t_5). Abrupt change in angular velocity has an adverse effect on ZMP and hence the dynamic equilibrium of bipedal robot. In this paper, the GAOFSF is modified to ensure that the trajectories generated are continuous in angular position and its first derivative and second derivative. $[t_5, t_2]$ and $[t_2, t_5]$ are regarded as the period of left support phase and right support phase, respectively. Then, t_2 and t_5 are the

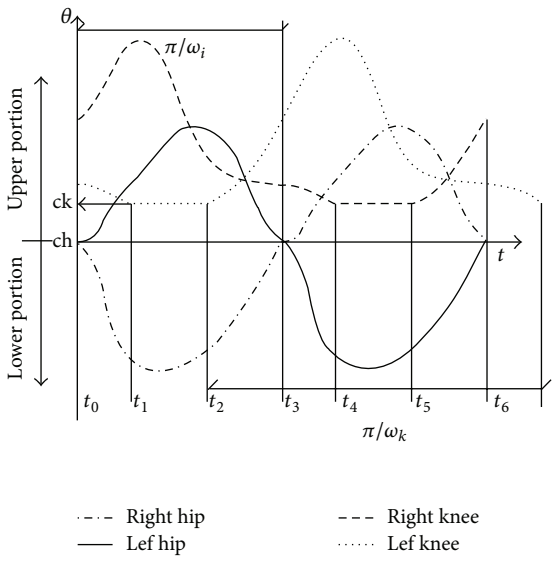

Figure 3: General shape of GAOFSF [6].

landing time of left support phase and right support phase, respectively [6]. The time duration $(t_5 - t_2)$ of one step is set as 1 s to ensure a reasonable walking speed. t_2 and t_5 are set as 0.7 s and 1.7 s while t_3 and t_6 are set as 1 s and 2 s. Four different parameters ($R_{\text{hs,support}}$, $R_{\text{hs,swing}}$, $R_{\text{ks,support}}$, and $R_{\text{ks,swing}}$) are added for adjustment of step length. Details of finding the values of these four parameters are discussed in Section 5. In this section, investigation is mainly focused on the formulation of modified CPG model.

3.1. Hip Trajectory in Sagittal Plane. Since the peaks of ${\theta_h}^+$ and ${\theta_h}^-$ are different, two different TFS are required to formulate the hip trajectory. Yang et al. [6] have proposed 5th order TFS and showed that the amplitudes of 4th and 5th orders are too small which can be neglected. 3rd order TFS is adopted. In (2)-(3), angular velocity of $\dot{\theta}_h^-$ and $\dot{\theta}_h^+$ at t_3 and t_6 is set as equal to ensure angular velocity is continuous. Thus, two more orders are added to ${\theta_h}^-$ to satisfy this constraint and are calculated by (4)-(5).

Consider

$$\dot{\theta}_h^- (t_6) = \dot{\theta}_h^+ (t_6), \tag{2}$$

$$\dot{\theta}_h^- (t_3) = \dot{\theta}_h^+ (t_3), \tag{3}$$

$$B_5 = \frac{-(2A_1 + 6A_3 + 2B_1 + 6B_3)}{10}, \tag{4}$$

$$B_4 = \frac{(-A_1 + 2A_2 - 3A_3 - B_1 - 2B_2 - 3B_3 - 5B_5)}{4}, \tag{5}$$

$$\begin{aligned} &t \in [t_0, t_3] \quad \theta_{\text{rhs}} = {\theta_h}^- (t) \quad \theta_{\text{rhs}} = {\theta_h}^+ (t), \\ &{\theta_h}^- = \sum_{n=1}^{5} R_{\text{hs,swing/support}} B_n \sin(n\omega_h t) + ch, \end{aligned} \tag{6}$$

$$\begin{aligned} &{\theta_h}^+ = \sum_{n=1}^{3} R_{\text{hs,swing/support}} A_n \sin(n\omega_h t) + ch, \\ &t \in [t_3, t_6] \quad \theta_{\text{rhs}} = {\theta_h}^+ (t - t_3) \quad \theta_{\text{lhs}} = {\theta_h}^- (t - t_3), \end{aligned} \tag{7}$$

$${\theta_h}^- = \sum_{n=1}^{5} R_{\text{hs,swing/support}} B_n \sin(n\omega_h (t - t_3)) + ch, \tag{8}$$

$${\theta_h}^+ = \sum_{n=1}^{3} R_{\text{hs,swing/support}} A_n \sin(n\omega_h (t - t_3)) + ch, \tag{9}$$

$$\omega_h = \frac{\pi}{t_3 - t_0}. \tag{10}$$

3.2. Knee Trajectory in Sagittal Plane. In the knee trajectory, Yang et al. [6] proposed a lock phase ($[t_1, t_2]$ and $[t_4, t_5]$) in knee trajectory which assumes constant joint angle and zero angular velocity and acceleration. This introduces abrupt change in angular velocity. The main goals to be achieved in the modified knee trajectory are (1) continuity in angular position, velocity, and acceleration and (2) the advantage proposed in GAOFSF that can be maintained. Then, a new formulation is proposed in the following equation:

$$\begin{aligned} &t \in [t_0, t_2], \\ &\theta_{\text{lks}} = \sum_{n=1}^{3} R_{\text{ks,swing/support}} C_n \sin(n\omega_k (t + (t_6 - t_5))) \\ &\qquad + \sum_{n=1}^{N=3} R_{\text{ks,swing/support}} C_n + ck \\ &t \in [t_2, t_6], \\ &\theta_{\text{lks}} = \sum_{n=1}^{3} R_{\text{ks,swing/support}} C_n \sin(n\omega_k (t - t_2)) \\ &\qquad + \sum_{n=1}^{N=3} R_{\text{ks,swing/support}} C_n + ck \\ &t \in [t_0, t_5], \\ &\theta_{\text{rks}} = \sum_{n=1}^{3} R_{\text{ks,swing/support}} C_n \sin(n\omega_k (t + (t_6 - t_5))) \\ &\qquad + \sum_{n=1}^{N=3} R_{\text{ks,swing/support}} C_n + ck \\ &t \in [t_5, t_6], \\ &\theta_{\text{rks}} = \sum_{n=1}^{3} R_{\text{ks,swing/support}} C_n \sin(n\omega_k (t - t_5)) \\ &\qquad + \sum_{n=1}^{N=3} R_{\text{ks,swing/support}} C_n + ck \\ &\omega_k = \frac{2\pi}{t_6 - t_0}. \end{aligned} \tag{11}$$

In (11), lock phase is canceled to ensure continuous angular velocity. By adjusting *ck* [6], GAOFSF can achieve energy efficient and stable "bent knee" walking gait. This advantage is still remained in the proposed modified model. The general shape of the hip and knee trajectories generated by modified CPG model is shown in Section 6.

3.3. Ankle Trajectory in Sagittal Plane. Right and left ankle joint trajectories are simply formulated as (12) to ensure that the trunk is upright and swing foot is parallel to the horizontal flat plane.

Consider

$$\theta_{\text{as}} = -\theta_{\text{hs}} - \theta_{\text{ks}}. \tag{12}$$

4. Optimization of Basic Walking Pattern by SaDE

This section focuses on how to search the basic walking pattern of bipedal robot by SaDE [19, 20]. To simplify the whole process in this section, $R_{\text{hs,support}}$, $R_{\text{hs,swing}}$, $R_{\text{ks,support}}$, and $R_{\text{ks,swing}}$ are set as 1. The following are the main objectives of basic walking pattern to be achieved through using SaDE.

(1) The desired step length is set as 0.05 m.

(2) Upper bound and lower bound of swing height are set as 0.02 m and 0.01 m, respectively.

(3) Premature landing should not occur throughout the walking cycle.

(4) ZMP is within the area of support polygon throughout the walking cycle.

The procedure of SaDE takes the following steps. Also, a flow chart of SaDE is shown in Figure 30 of the Appendix to facilitate the understanding of readers. Details of the procedure are discussed in Sections 4.1–4.7.

(1) Select fitness functions (f_i) and constraints (S_i).

(2) Code the parameters to be searched to form target vector ($X_{i,G}$).

(3) Initialize the first generation (G_1) of population (i).

(4) Initialize crossover rate ($\text{CR}_{u_{i,j}}$), scaling factor ($F_{u_{i,j}}$), and probability (P_j) for each mutation strategies (j) based on $\overline{\text{CR}}_j$ and $\overline{F}_j$.

(5) Select mutation strategies (j) based on probability (P_j) through MATLAB function rand() and perform mutation operation to generate mutant vectors (v_i).

(6) Perform crossover operation to generate trial vectors (u_i).

(7) Perform selection operation to select next generation of target vectors ($X_{i,G+1}$).

(8) During learning period (LP), record the successful time (ST_j) and failure time (FT_j) of each strategy (j).

(9) During learning period (LP), record values of $\text{CR}_{u_{i,j}}$ and $F_{u_{i,j}}$ for each strategy (j) that successfully help the trial vectors (u_i) enter the next generation G_{i+1}.

(10) Upon completion of LP, probability (P_j) for choosing suitable strategies (j) is adjusted based on successful rate of strategy j (SR_j) calculated through ST_j and FT_j.

(11) Upon completion of LP, $\overline{\text{CR}}_j$ and $\overline{F}_j$ are adjusted based on the mean values of stored $\text{CR}_{u_{i,j}}$ and $F_{u_{i,j}}$, respectively.

(12) Repeat procedures (4)–(11) until maximum generation is completed

4.1. Fitness Functions and Constraints (Step 1). Fitness functions (f_n) and constraints (S_n) are designed or modified based on [6]. Since trunk occupies a large proportion of the whole body mass, abrupt change in trunk velocity can lead to abrupt change in ZMP_y. f_1 is formulated as follows:

$$\overline{V}_{\text{trunk}} = \sum_{n=1}^{N} \frac{V_{\text{trunk},n}}{N},$$
$$f_1 = \sqrt{\sum_{n=1}^{N} \frac{\left(V_{\text{trunk},n} - \overline{V}_{\text{trunk}}\right)^2}{N}}, \tag{13}$$

where N is number of data.

Strike velocity during landing should be as small as possible since impact between landing foot and the ground can cause mechanical wear of parts and unstable walking. f_2 is stated as follows [6]:

$$f_2 = \sqrt{{V_{\text{strike},x}}^2 + {V_{\text{strike},y}}^2 + {V_{\text{strike},z}}^2}. \tag{14}$$

If ZMP_y is within the area of support polygon $[0.061, -0.061]$ throughout the walking cycle, dynamic equilibrium of the robot is satisfactory. To deal with the discrepancies between simulation model and physical test bed, a safety factor (0.01 m) is added to ensure good performance during experiment. S_1 is formulated as follows:

$$S_1 = \sum_{n=1}^{N} \max\left(\left|\text{ZMP}_{y,n}\right| - \left(L_y - \text{safty factor}\right), 0\right), \tag{15}$$

where L_y is length (0.121 m) of foot sole/2.

The robot is assigned to walk forward along positive y-axis. To ensure correct direction, S_2 and S_3 are designed as follows:

$$S_2 = \max\left(-\overline{V}_{\text{trunk},y}, 0\right),$$
$$S_3 = \max\left(-\overline{V}_{\text{swing foot},y}, 0\right), \tag{16}$$

where $\overline{V}_{\text{trunk},y}$ is mean trunk velocity in y-direction and $\overline{V}_{\text{swing foot},y}$ is mean swing foot velocity in y-direction.

To achieve natural human-like walking gait, S_4 and S_5 are designed based on the modified CPG model.

$$\begin{aligned}&\text{If } t \geq t_0 \;\&\&\; t \leq t_3,\\ &\quad S_4 = S_4 + \max\left(\theta_{\text{rhs},n} - ch, 0\right)\\ &\qquad + \max\left(-\left(\theta_{\text{lhs},n} - ch\right), 0\right).\\ &\text{Else if } t \geq t_3 \;\&\&\; t \leq t_6,\\ &\quad S_4 = S_4 + \max\left(\theta_{\text{lhs},n} - ch, 0\right)\\ &\qquad + \max\left(-\left(\theta_{\text{rhs},n} - ch\right), 0\right).\\ &\text{End}\end{aligned} \tag{17}$$

$$S_5 = \sum_{n=1}^{N} \max\left(-\theta_{\text{lks},n}, 0\right). \tag{18}$$

S_6 is designed to ensure that the robot can achieve desired step length:

$$S_6 = \left|\text{desired step length} - \text{step length}\right|. \tag{19}$$

To ensure reasonable swing height, S_7 is designed to ensure that the swing height is within the [0.01, 0.02] m.

$$\begin{aligned}&\text{If } \max\left(H_{\text{swing foot}}\right)\\ &\quad \geq H_{\text{lower}} \;\&\&\; \max\left(H_{\text{swing foot}}\right) \leq H_{\text{upper}},\\ &\quad S_7 = 0.\end{aligned} \tag{20}$$

$$\begin{aligned}&\text{Else}\\ &\quad S_7 = \sum_{n=1}^{N} \left|H_{\text{desired}} - H_{\text{swing Foot},n}\right|.\\ &\text{End.}\end{aligned} \tag{21}$$

S_8 is designed to prevent the swing foot from penetrating through the ground and hence reduce the chance of premature landing while S_9 is designed to ensure swing foot lands on the ground ($H_{\text{ground}} = 0$ m) at landing time which is t_2 and t_5:

$$\begin{aligned}S_8 &= \sum_{n=1}^{N} \max\left(H_{\text{support foot},n} - H_{\text{swing foot},n}, 0\right),\\ S_9 &= \left|H_{\text{swing foot},t_2}\right| + \left|H_{\text{swing foot},t_5}\right|.\end{aligned} \tag{22}$$

The total scores used for evaluating each target vector ($X_{i,G}$) are formulated as follows:

$$\text{scores} = -1200 + \sum_{i=1}^{N} \omega_{o,n} f_n + \sum_{i=1}^{N} \omega_{p,n} S_n. \tag{23}$$

These weightings are set by trial and error to ensure their importance is almost the same and walking gait with satisfactory performance can be obtained:

$$\begin{aligned}\omega_o &= [1000, 500],\\ \omega_p &= [2500, 285, 265, 100, 2200, 12000, 300, 960, 25000],\end{aligned} \tag{24}$$

where $\omega_{o,n}$ is weighting of fitness functions and $\omega_{p,n}$ is weighting of constraints.

4.2. Initialization of Parameters of Target Vectors ($X_{i,G}$) (Steps 2 and 3). The target vector ($X_{i,G}$) is formulated as follows:

$$X_{i,G} = \left[A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2, C_3, ch, ck\right]. \tag{25}$$

Then, the parameters of population (i) are initialized by MATLAB function rand() which generates [0, 1] randomly.

4.3. Initialization of Parameters of SaDE (Step 4). Since the range $[-1, 1]$ of parameters to be searched is small, then population (i) of one generation (G) is set as 30 to balance between good diversity of population and low computation load. Four mutation strategies ($j = 1, 2, 3, 4$) are adopted. For strategy j, MATLAB function *normrnd* (mean value, σ) is used to generate $\text{CR}_{u_{i,j}}$ and $F_{u_{i,j}}$ for each trial vector (u_i). In the initial phase, the crossover rate ($\text{CR}_{u_{i,j}}$) should not be too high to prevent premature convergence while the scaling factor ($F_{u_{i,j}}$) should not be too small to affect the exploration ability. Hence, a moderate value (0.5) is assigned to $\overline{\text{CR}}_j$ and $\overline{F}_j$ (mean value) and σ is set as 0.1. This function can generate random numbers from the normal distribution through mean value ($\overline{\text{CR}}_j$, $\overline{F}_j$) and standard deviation (σ). P_j is set as 0.25 for strategy j so that each strategy has the equal chance to be chosen during the first learning period:

$$\begin{aligned}\text{CR}_{u_{i,j}} &= normrnd\left(\overline{\text{CR}}_j, \sigma\right),\\ F_{u_{i,j}} &= normrnd\left(\overline{F}_j, \sigma, \right).\end{aligned} \tag{26}$$

4.4. Mutation Operation of SaDE (Step 5). DE/rand/1, DE/rand/2 and DE/current-to-rand/2 provide good exploration ability while DE/current-to-best/2 demonstrates good convergence speed. To balance between exploration ability and convergence speed, these four strategies are adopted. These four mutation strategies are stated in (27)–(30) as follows:

(1) DE/rand/1,

$$v_{i,G} = X_{r1,G} + F_{i,j}\left(X_{r2,G} - X_{r3,G}\right), \tag{27}$$

(2) DE/rand/2,

$$v_{i,G} = X_{r1,G} + F_{i,j}\left(X_{r2,G} - X_{r3,G}\right) + F_{i,j}\left(X_{r4,G} - X_{r5,G}\right), \tag{28}$$

(3) DE/rand-to-best/2,

$$v_{i,G} = X_{i,G} + F_{i,j}\left(X_{\text{best},G} - X_{i,G}\right) + F_{i,j}\left(X_{r1,G} - x_{r2,G}\right) + F_{i,j}\left(X_{r3,G} - X_{r4,G}\right), \quad (29)$$

(4) DE/current-to-rand/2,

$$v_{i,G} = X_{i,G} + F_{i,j}\left(X_{r1,G} - X_{i,G}\right) + F_{i,j}\left(X_{r2,G} - X_{r3,G}\right). \quad (30)$$

4.5. Crossover Operation of SaDE (Step 6). Because of its popularity, binomial crossover operator [20] is utilized in all mutation strategies.

$$\begin{aligned} &\text{If } \left(\text{rand}() \le \text{CR}_{i,j}\right) || \left(q == q_{\text{rand}}\right), \\ &\qquad u_{q,i,G} = v_{q,i,G}. \\ &\text{Else if } \left(\text{rand}() > \text{CR}_{i,j}\right), \\ &\qquad u_{q,i,G} = x_{q,i,G}. \\ &\text{End}, \end{aligned} \quad (31)$$

where q is the parameter number $(1, 2, 3, 4, \ldots, 11)$ in the target $(X_{i,G})$ and mutant $(v_{i,G})$ vector. Arbitrary parameter number is assigned to q_{rand} to ensure that trial vector $(u_{i,G})$ is different from target vector $(X_{i,G})$. If values of $u_{q,i,G}$ exceed the desired range $[-1, 1]$ of parameters to be searched, $u_{q,i,G}$ is reset by MATLAB function rand ().

4.6. Selection Operation of SaDE (Step 7). If score of trial vector $(u_{i,G})$ is lower than that of target vector $(x_{i,G})$, then trial vector enters the next generation. Otherwise, target vector enters the next generation.

$$\begin{aligned} &\text{If scores}_{u_{i,G}} \le \text{scores}_{x_{i,G}}, \\ &\qquad x_{i,G+1} = u_{i,G}. \\ &\text{Else} \\ &\qquad x_{i,G+1} = x_{i,G}. \\ &\text{End}. \end{aligned} \quad (32)$$

4.7. Adjustment of P_j, $\overline{CR}_j$ and $\overline{F}_j$ (Steps 8–11). Initially, a learning period (LP = 5 *generations*) is assigned to balance between good sample size of data and update frequency. During learning period, successful times (ST_j) and failure times (FT_j) of each mutation strategy $(j = 1, 2, 3, 4)$ in each generation are recorded. For example, if strategy j is chosen and helps one trial vector $(u_{i,G})$ to enter next generation, then ST_j is added by 1. Otherwise, FT_j is added by 1. Once learning period is completed, P_j is adjusted by the successful rate (SR_j). Then, ST_j and FT_j are reset to zero for the next learning period to eliminate effect of the past data:

$$\begin{aligned} \text{SR}_j &= \frac{\text{ST}_j}{\text{FT}_{j,G} + \text{ST}_{j,G}} \\ P_j &= \frac{\text{SR}_j}{\sum_{j=1}^{N=4} \text{SR}_j}. \end{aligned} \quad (33)$$

Similar approach is applied to adjust $\overline{\text{CR}}_j$ and $\overline{F}_j$. During learning period (LP = 5 generations), for strategy j, $\text{CR}_{u_{i,j}}$ and $F_{u_{i,j}}$ corresponding to the trial vectors $(u_{i,G})$ that successfully enters next generation are stored in database. Once learning period is completed, mean value $(\overline{\text{CR}}_j$ and $\overline{F}_j)$ of the stored value corresponding to strategy j is calculated and is used to generate a new set of $\text{CR}_{u_i,j}$ and $F_{u_{i,j}}$. Then, storage data of $\text{CR}_{u_{i,j}}$ and $F_{u_{i,j}}$ are removed for the next learning period to eliminate the effects of past data.

5. Optimization of Factors ($R_{\text{hs,support}}$, $R_{\text{hs,swing}}$, $R_{\text{ks,support}}$, and $R_{\text{ks,swing}}$) for Step Length Adjustment

5.1. Objective Functions and Constraints of SaDE. The objective functions and constraints mentioned in Section 4 are adopted to search appropriate value of $R_{\text{hs,support}}$, $R_{\text{hs,swing}}$, $R_{\text{ks,support}}$, and $R_{\text{ks,swing}}$final corresponding to different desired step length (0.04 m, 0.03 m, 0.02 m, and 0.01 m).

5.2. Lower Bound of Maximum Desired Swing Height. It is natural that the maximum swing height becomes lower to consume less energy if step length is smaller. Lower bound of swing height is reset as lower value for different step lengths (Table 3) while the upper bound (0.02 m) remains the same.

Table 3: Lower bound of desired swing height corresponding to different step lengths.

Step length (m)	Lower bound of maximum desired swing height (m)
0.05 m	0.01 m
0.04 m	0.009 m
0.03 m	0.008 m
0.02 m	0.007 m
0.01 m	0.006 m

5.3. Smooth Transition of $R_{joint,support/swing}$. Values of $R_{\text{joint,support/swing}}$ (joint = rhs, lhs, rks, and lks) of support leg and swing leg are different. At the moment of landing time, $R_{\text{joint,support}}$ is changed to $R_{\text{joint,swing}}$ since the support leg is changed to swing leg in the next step. In order to have a smooth transition, fifth order polynomial shown in equation (34) is utilized. The period of transition time $(t_f - t_0)$ is set as 0.4 s which is 40% of time duration (1 s) of one step to balance between rapid transition time and good dynamic equilibrium.

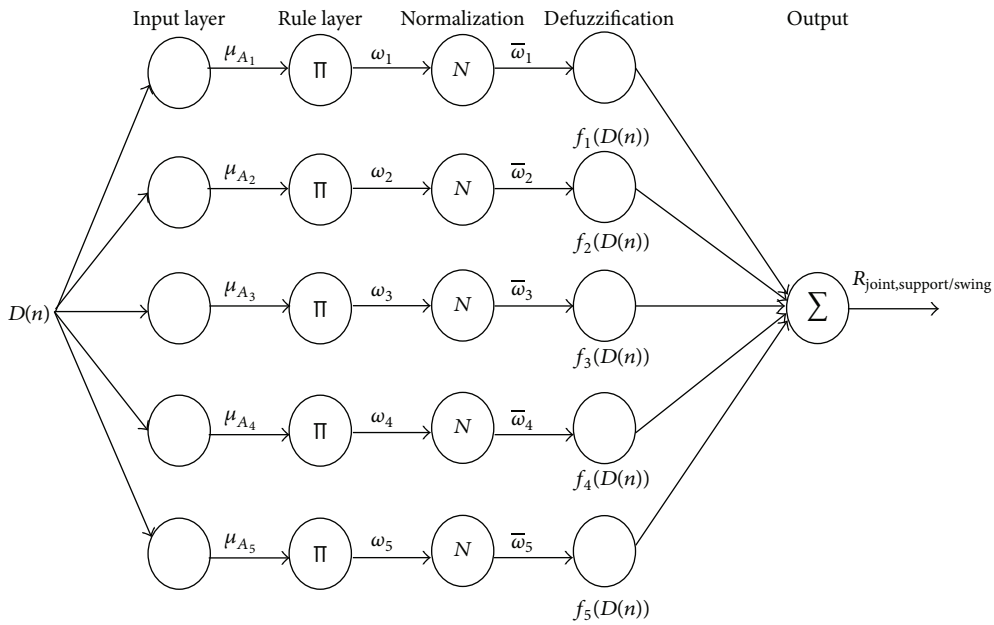

Figure 4: Architecture of Fuzzy Inference System (FIS).

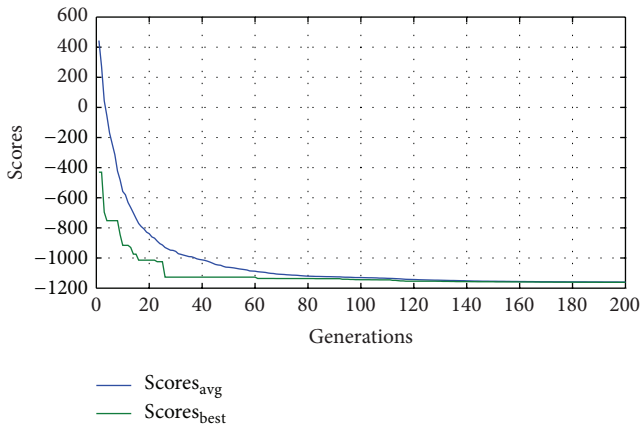

Figure 5: Average scores and scores of best target vector of modified CPG model.

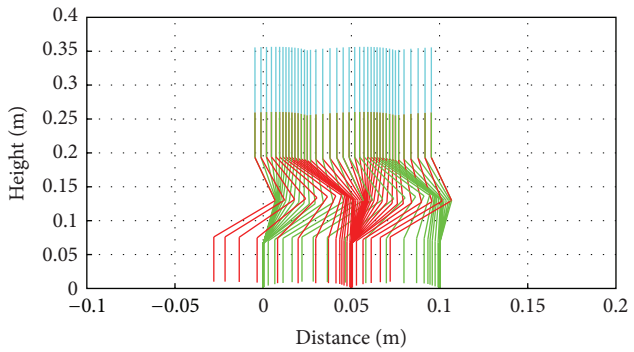

Figure 6: Walking gait of modified CPG model searched by SaDE.

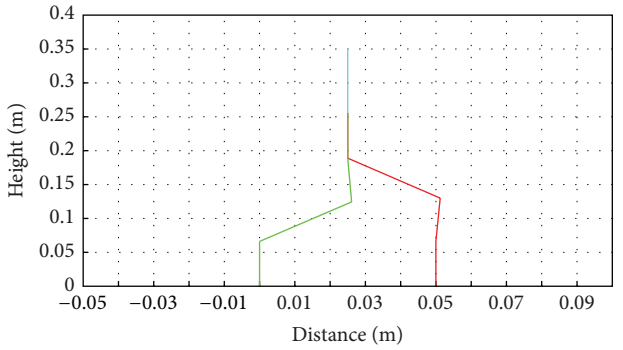

Figure 7: The Landing instant of right swing leg (modified CPG Model).

Table 4: $R_{\text{hs,support}}$, $R_{\text{ks,support}}$, $R_{\text{hs,swing}}$, and $R_{\text{ks,swing}}$ searched by SaDE.

Desired step length (cm)	$R_{\text{hs,support}}$	$R_{\text{ks,support}}$	$R_{\text{hs,swing}}$	$R_{\text{ks,swing}}$
5	1	1	1	1
4	0.8391	0.9419	0.7923	0.9714
3	0.7915	0.8469	0.5576	0.9672
2	0.6090	0.7861	0.3165	0.8709
1	0.4885	0.7157	0.0494	0.7779
0	0	0	0	0

Consider

$$R_{\text{joint,support/swing}} = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5. \quad (34)$$

Table 5: Parameters of FLS tuned by LSE.

	$f_1(D)$	$f_2(D)$	$f_3(D)$	$f_4(D)$	$f_5(D)$
$\text{FLS}_{\text{hs,support}}$	$k_{11} = 6.6435$ $k_{12} = 25.6113$	$k_{21} = -15.4254$ $k_{22} = -17.4962$	$k_{31} = 7.3103$ $k_{32} = 22.3738$	$k_{41} = 7.3103$ $k_{42} = 7.3103$	$k_{51} = -8.2392$ $k_{52} = 13.8564$
$\text{FLS}_{\text{ks,support}}$	$k_{11} = 4.3824$ $k_{12} = 24.6557$	$k_{21} = -9.6047$ $k_{22} = -14.8921$	$k_{31} = 2.6114$ $k_{32} = 15.1129$	$k_{41} = 3.7569$ $k_{42} = -9.1219$	$k_{51} = -5.5025$ $k_{52} = 7.3744$
$\text{FLS}_{\text{hs,swing}}$	$k_{11} = -2.7274$ $k_{12} = -9.6848$	$k_{21} = 5.7349$ $k_{22} = 7.1272$	$k_{31} = -1.2563$ $k_{32} = -7.2583$	$k_{41} = -0.6222$ $k_{42} = 5.9490$	$k_{51} = 3.6331$ $k_{52} = -3.4074$
$\text{FLS}_{\text{ks,swing}}$	$k_{11} = 7.1051$ $k_{12} = 32.6818$	$k_{21} = -16.0968$ $k_{22} = -21.0042$	$k_{31} = 6.2666$ $k_{32} = 24.1201$	$k_{41} = 4.0344$ $k_{42} = -17.0796$	$k_{51} = -9.0802$ $k_{52} = 13.2950$

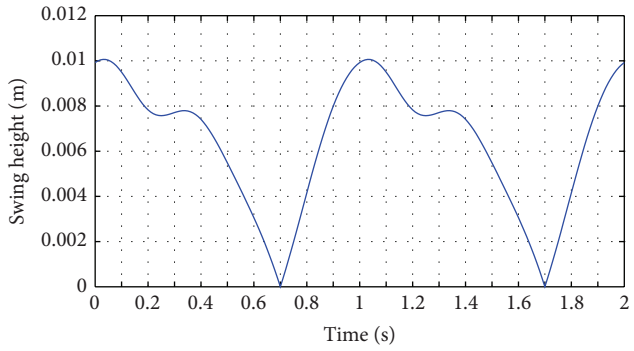

Figure 8: Variation of height of swing foot with time (modified CPG model).

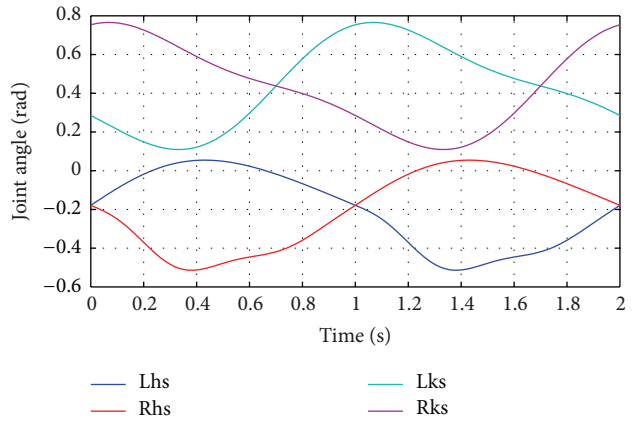

Figure 9: Joint trajectories of modified CPG model searched by SaDE.

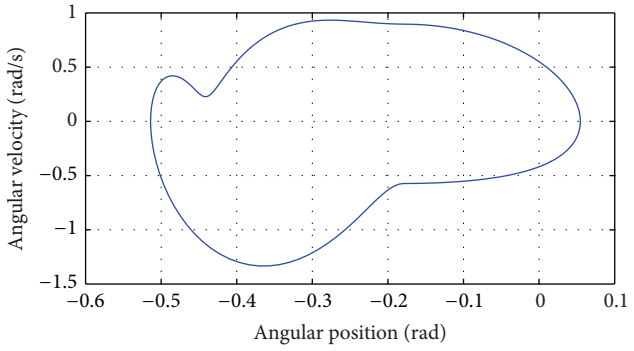

Figure 10: $\dot{\theta}_{\text{hs}}$ versus θ_{hs} of modified CPG model.

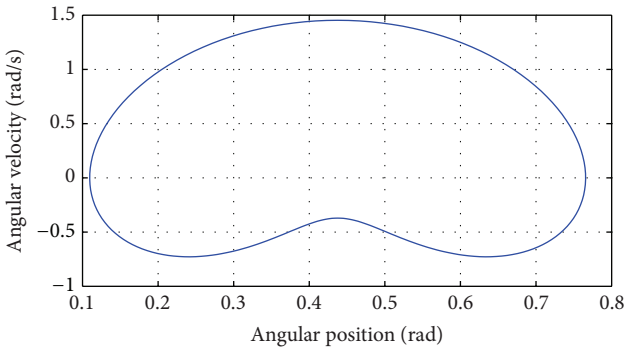

Figure 11: $\dot{\theta}_{\text{ks}}$ versus θ_{ks} of modified CPG model.

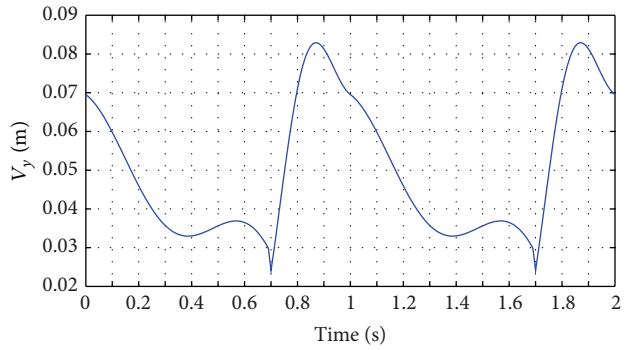

Figure 12: Variation of $V_{\text{trunk},y}$ with time (modified CPG model).

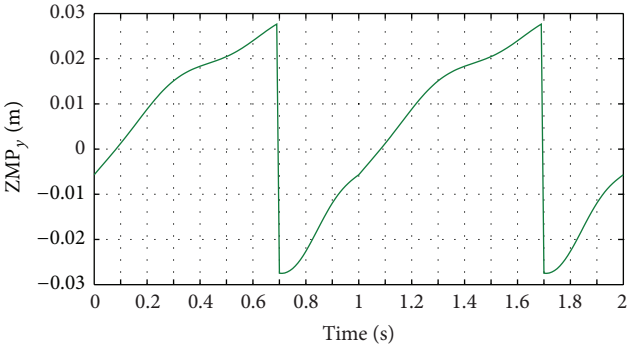

Figure 13: Variation of ZMP_y with time (modified CPG model).

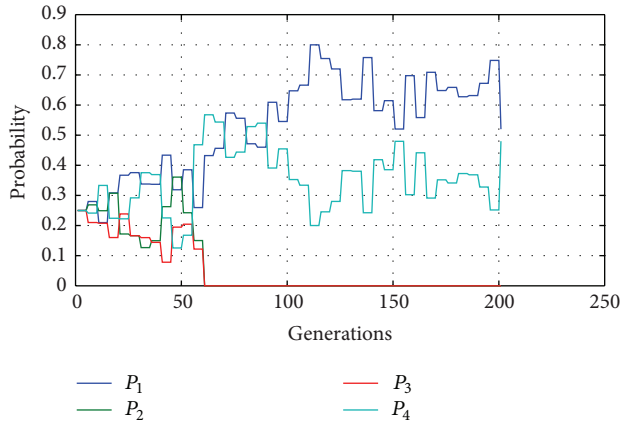

FIGURE 14: Probability of mutation strategy j to be chosen (modified CPG model).

The coefficients of $R_{\text{joint,support/swing}}$ are solved by the following constraints:

$$\begin{aligned} R_{\text{joint,support/swing}}(t_0) &= R_{\text{joint,support/swing,initial}}, \\ R_{\text{joint,support/swing}}(t_f) &= R_{\text{joint,support/swing,final}}, \\ \dot{R}_{\text{joint,support/swing}}(t_0) &= 0, \\ \dot{R}_{\text{joint,support/swing}}(t_f) &= 0, \\ \ddot{R}_{\text{joint,support/swing}}(t_0) &= 0, \\ \ddot{R}_{\text{joint,support/swing}}(t_f) &= 0. \end{aligned} \tag{35}$$

5.4. Fuzzy Inference System (FIS). Four FISs are adopted to learn the relationship between (1) $R_{\text{hs,support}}$, $R_{\text{hs,swing}}$, $R_{\text{ks,support}}$, and $R_{\text{ks,swing}}$ and (2) desired step length (5 cm, 4 cm, 3 cm, 2 cm, 1 cm, and 0 cm). The architecture of FIS proposed by [28] is shown in Figure 4. The desired step length ($D(n)$) is the input of FIS while $R_{\text{joint,support/swing}}$ is the output of FIS which is obtained through the summation of $f_m(D(n))$. Based on [24], $f_m(D(n))$ is a function of desired step and can be represented as a first order polynomial stated in (40).

In this section, the membership function (M_m) is set as Gaussian function as follows:

$$\mu_{A_m} = e^{-0.5((x-c_m)/\sigma_m)^2}, \tag{36}$$

$$\omega_m = \mu_{A_m}, \tag{37}$$

$$\overline{\omega}_m = \frac{\omega_m}{\sum_{m=1}^{5} \omega_m}, \tag{38}$$

$$f_m(D(n)) = k_{m,1} + k_{m,2} D(n), \tag{39}$$

$$R_{\text{joint,support/swing}} = \sum_{m=1}^{5} \overline{\omega}_m f_m(D(n)), \tag{40}$$

where μ_{A_m} is degree of membership function of M_m, σ_m is parameter that affects the width of Gaussian function of M_m, c_m is parameter that affects the center of Gaussian function of M_m, ω_m is firing strength of rule m, and $\overline{\omega}_m$ is normalized firing strength.

This section focuses on adjusting the consequent parameters $(k_{m,1}, k_{m,2})$ of $f_m(X)$ by least square estimation (LSE) [28] stated in (41). Gradient descent algorithm is not adopted to adjust premise parameters since the size of training data ($N = 6$) is relatively small. Hence, premise parameters are fixed during the training process. In this paper, c_{1-5} are set as 0.2, 0.4, 0.6, 0.8, and 1, respectively, while σ_{1-5} are set as 0.3. After 50 training cycles, an appropriate set of consequent parameters is found. Details of result in this section are stated in Section 6.

Consider

$$K^* = \left(U^T U\right)^{-1} U^T Y, \tag{41}$$

$$K^* = [k_{1,1};\ k_{1,2};\ k_{2,1};\ k_{2,2};\ \ldots\ k_{5,1};\ k_{5,2}], \tag{42}$$

$$U = \begin{bmatrix} \overline{\omega}_1(1) & \overline{\omega}_1(1) D(1) & \overline{\omega}_2(1) & \overline{\omega}_2(1) D(1) & \ldots & \overline{\omega}_5(1) & \overline{\omega}_5(1) D(1) \\ & & & \vdots & & & \\ \overline{\omega}_1(N) & \overline{\omega}_1(N) D(N) & \overline{\omega}_2(N) & \overline{\omega}_2(N) D(N) & \ldots & \overline{\omega}_5(N) & \overline{\omega}_5(N) D(N) \end{bmatrix}, \tag{43}$$

where D is vector of desired step length and Y is vector of desired output.

6. Results and Discussions

6.1. Scores and Searched Parameters of Modified CPG Model. According to Figure 5, the maximum generation is set as 200. The scores of the best target vector is −1163.5. The values of fitness functions and constraints are [0.018 ms^{-1}, 0.0369 ms^{-1}] and [0 m, 0 ms^{-1},0 ms^{-1}, 0 rad, 0 rad, 0 m, 0 m, 0 m, and 0 m], respectively. The searched parameters are [0.2290 0.0257 0.0016 −0.3161 −0.0199 −0.0001 0.2892 0.0861 0.0004 −0.1786 0.0619]. The results show that the modified CPG model can achieve a satisfactory performance.

6.2. Kinematic Aspects of Modified CPG Model. The walking gait searched by SaDE is visualized in Figure 6. Figure 7

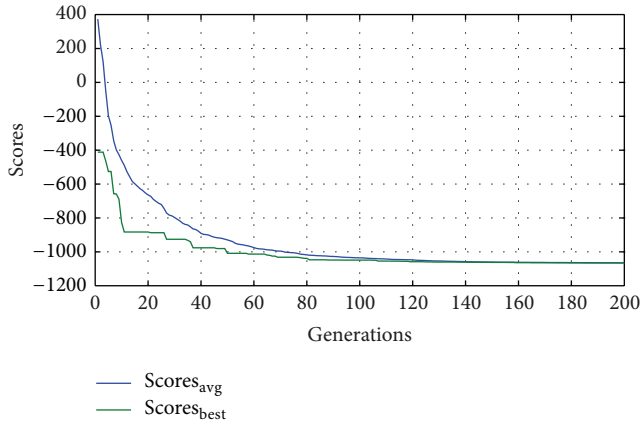

Figure 15: Average scores and scores of best target vector of original CPG model.

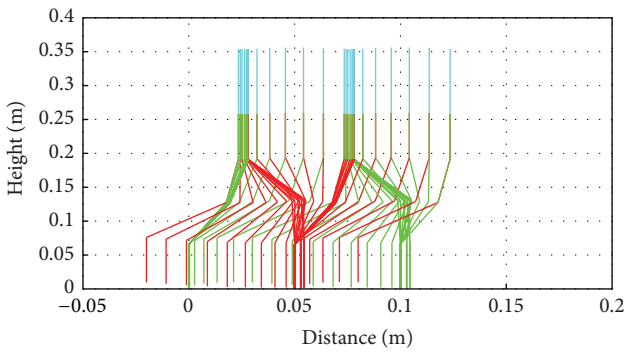

Figure 16: Walking gait of original CPG model searched by SaDE.

shows that step achieved by bipedal robot is the same as the desired step length (0.05 m). In Figure 8, maximum height of swing foot is satisfactory based on Table 3. Swing foot lands on the ground at desired landing time ($t_2 = 0.7$ s and $t_5 = 1.7$ s). The joint trajectories of modified CPG model are shown in Figure 9. Based on Figures 10 and 11, angular position and velocity of each joint trajectory is observed to be continuous and smooth. Because of the heavy trunk mass, abrupt change in $V_{\text{trunk},y}$ should be prevented. From Figure 12, it shows that $V_{\text{trunk},y}$ is continuous and smooth except the landing time.

6.3. Dynamic Aspects of Modified CPG Model. Figure 13 shows that dynamic equilibrium of the robot is satisfactory since ZMP$_y$ is within the area of support polygon $[-0.061, 0.061]$ m. The length of foot sole is 0.122 m while ZMP$_y$ margin is defined to be (length of foot sole/2) − max(|ZMP$_y$|). Based on Figure 13, ZMP$_y$ margin is observed to be at least 0.03 m which is large enough to allow larger step. ZMP$_y$ is smooth and the only abrupt change happens at the landing time ($t_2 = 0.7$ s and $t_5 = 1.7$ s) since ZMP$_y$ shifts to the support foot of the next step at these two moments.

6.4. Observations on Mutation Strategies of SaDE. During searching parameters of modified CPG model, it shows

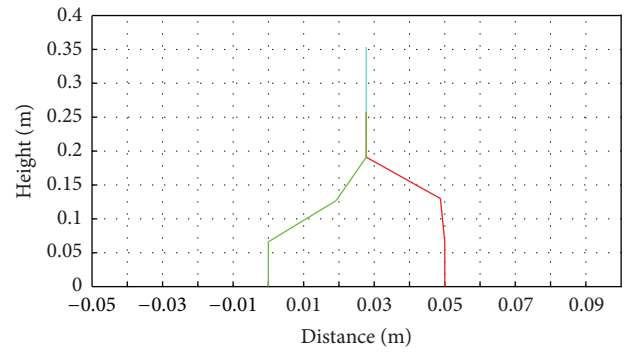

Figure 17: The landing instant of right swing leg (original CPG model).

that mutation strategy 2 and strategy 3 do not favor for searching feasible walking gait since they are suppressed completely after 60 generations (Figure 14). Figure 5 shows that scores of target vectors tend to converge in the range of generation 100 to 200. Strategy 1 can help trial vector converge faster since it involves the best trial vector in the mutation operation. Hence, from generation 100 to 200, the probability for choosing strategy 1 is always higher than that of strategy 4.

6.5. Results of Original CPG Model. Except constraint 1, the original CPG model [6] is searched by SaDE under the same setting. Safety factor is set as 0 m since it becomes difficult to search a feasible walking gait for original CPG model if safety factor is set as 0.01 m. Based on Figure 15, the scores of the best trial vector is −1105.9. Also, the searched parameters are [0.0519 0.0012 0.0070 −0.5237 −0.0053 −0.0068 0.3579 0.1459 0.0025 0.0891 0.1736]. The values of fitness functions and constraints are [0.0694 ms^{-1}, 0.0484 ms^{-1}] and [0.0002 m, 0 ms^{-1}, 0 ms^{-1}, 0 rad, 0 rad, 0 m, 0 m, 0 m, 0 m], respectively. Compared with the performance of modified CPG model, the original CPG model is less satisfactory. The walking gait of original CPG model searched by SaDE is visualized in Figure 16. Also, in Figure 17, desired step length (0.05 m) is successfully achieved. In Figure 18, maximum height of swing foot is satisfactory based on Table 3. Also, swing foot lands on the ground at desired landing time ($t_2 = 0.7$ s and $t_5 = 1.7$ s).

The joint trajectories of modified CPG model are shown in Figure 19. Based on Figures 20 and 21, angular velocity of each joint trajectory is observed to be discontinuous. The standard deviation ($\sigma_{\text{original}} = 0.0694$ ms^{-1}) of $V_{\text{Trunk},y}$ is larger than that ($\sigma_{\text{modified}} = 0.018$ ms^{-1}) of modified CPG model. The prime reason is that, in Figure 22, $V_{\text{Trunk},y}$ is observed to be discontinuous because of discontinuity in angular velocity. In Figure 23, obvious abrupt change in ZMP$_y$ is observed. Also, ZMP$_y$ exceeds the area of support polygon ($S_1 = 0.0002$ m). This shows that dynamic equilibrium of bipedal robot is affected by discontinuous angular velocity. Although the area of foot sole can be made larger to tolerate the abrupt change of ZMP$_y$, the agility of the robot is sacrificed. For the original CPG model, a larger step is not allowed since ZMP$_y$ has exceeded the support polygon when the robot walks with 5 cm step length.

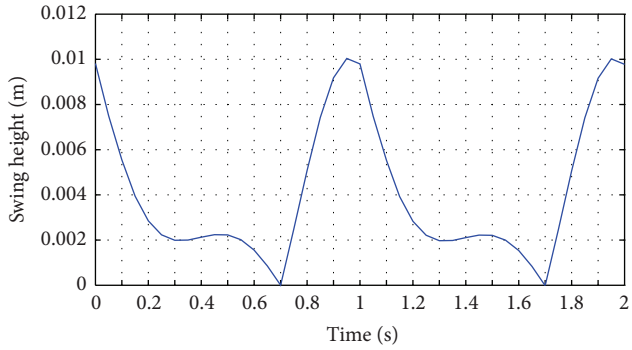

Figure 18: Variation of height of swing foot with time (original CPG model).

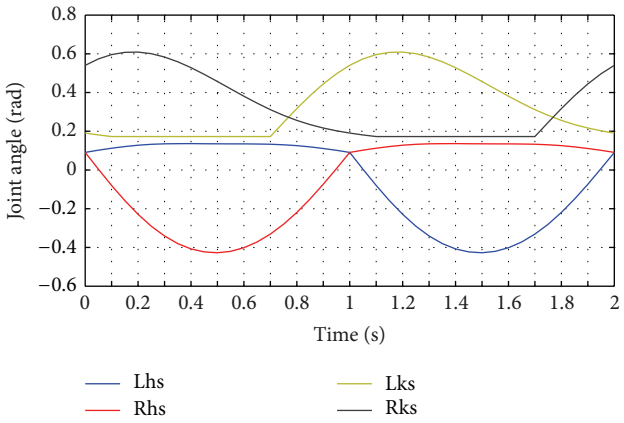

Figure 19: Joint trajectories of original CPG model searched by SaDE.

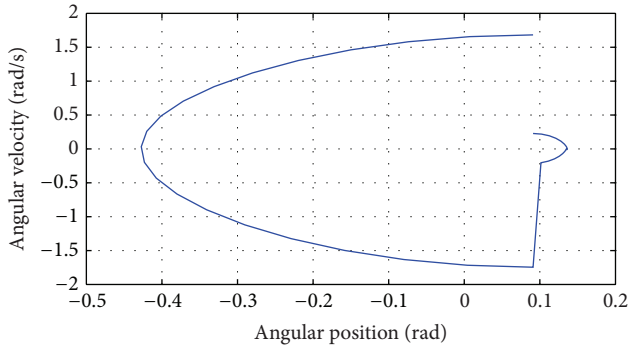

Figure 20: $\dot{\theta}_{hs}$ versus θ_{hs} of original CPG model.

6.6. Results of $R_{hs,support}$, $R_{ks,support}$, $R_{hs,swing}$, and $R_{ks,swing}$ Searched by SaDE. $R_{hs,support}$, $R_{ks,support}$, $R_{hs,swing}$, and $R_{ks,swing}$ searched by SaDE are shown in Table 4. In Figure 24, it shows that the bipedal robot can achieve desired step length (4 cm, 3 cm, 2 cm, and 1 cm). Also, in Figure 25, the maximum swing height corresponding to different desired step length reaches reasonable swing height based on Table 3 and no premature landing occurs during walking.

In Figure 26, ZMP_y is always within the area of support polygon [0.061, −0.061]. Hence, the dynamic equilibrium of

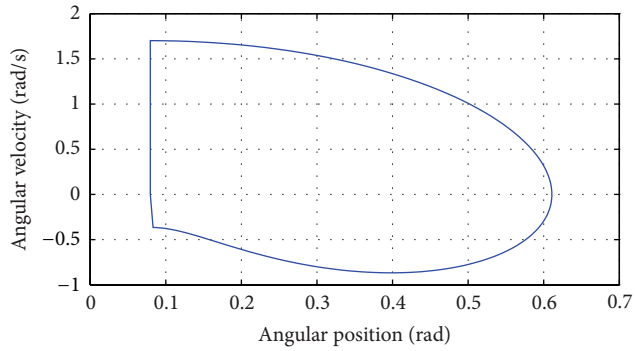

Figure 21: $\dot{\theta}_{ks}$ versus θ_{ks} of original CPG model.

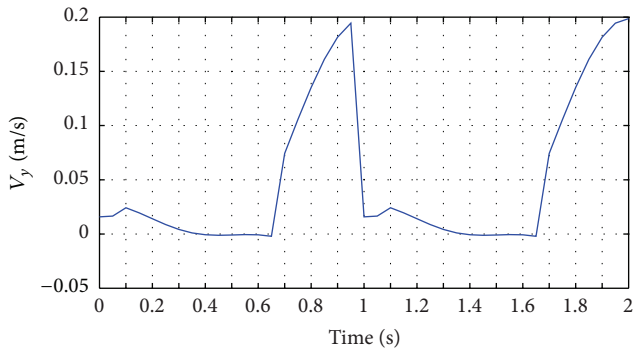

Figure 22: Variation of $V_{Trunk,y}$ with time (original CPG model).

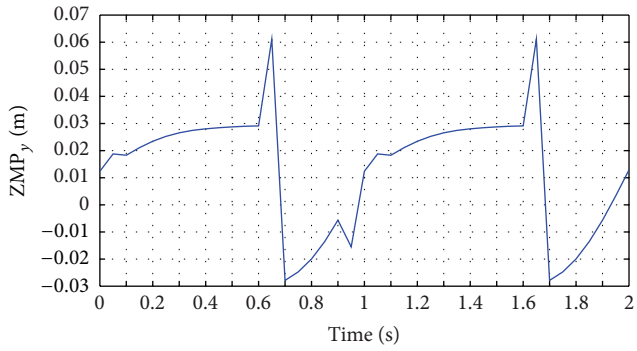

Figure 23: Variation of ZMP_y with time (original CPG model).

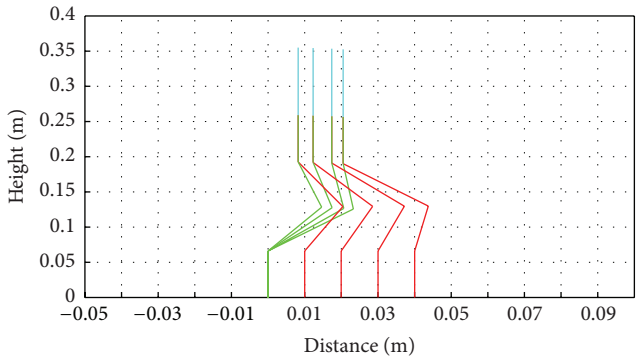

Figure 24: Different step length achieved by bipedal robot.

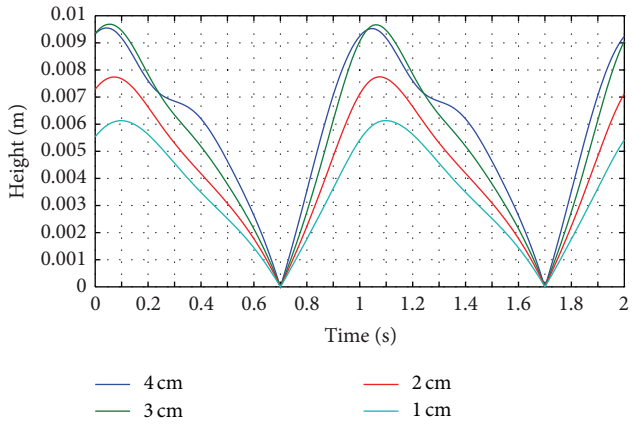

Figure 25: Variation of height of swing foot with time (different desired step length).

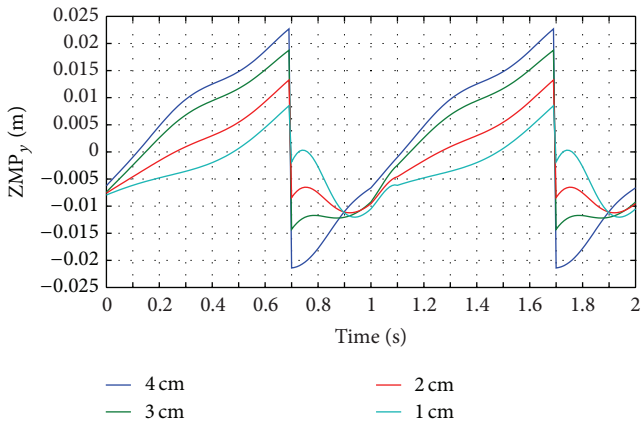

Figure 26: Variation of ZMP$_y$ with time (different desired step length).

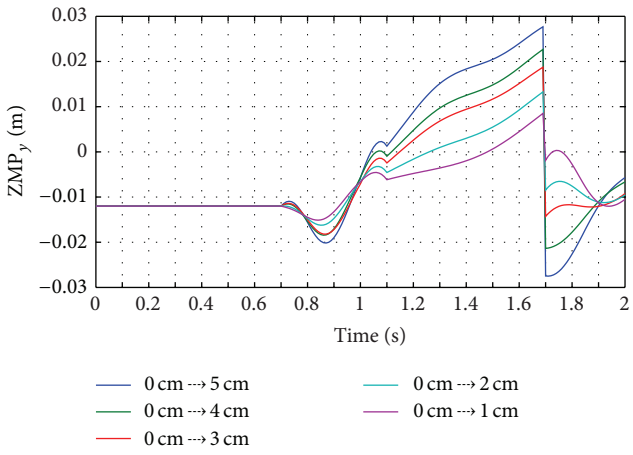

Figure 27: Variation of ZMP$_y$ corresponding to different desired step length change.

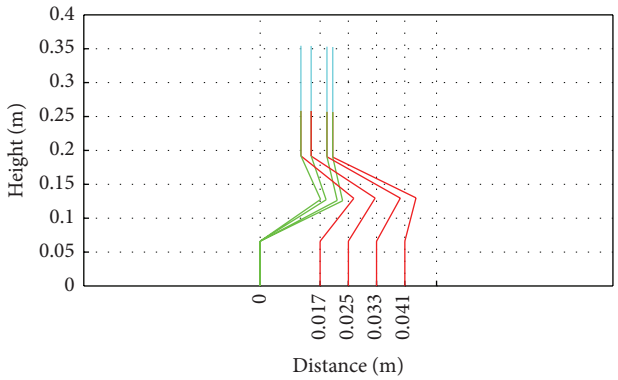

Figure 28: Step length achieved by bipedal robot (arbitrary desired step length command).

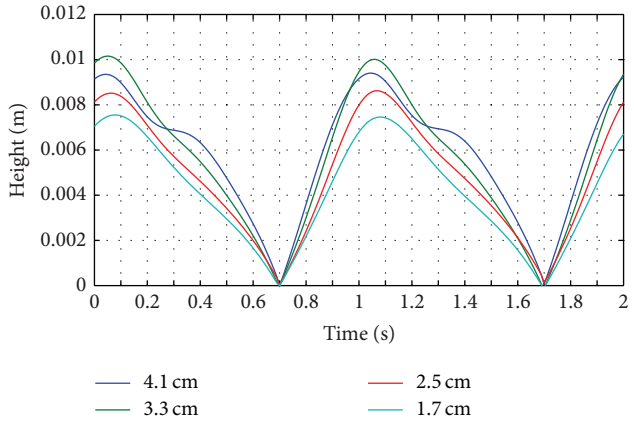

Figure 29: Height of swing foot (arbitrary desired step length command).

the robot for different desired step length is satisfactory. Then, an attempt is made to show that when desired step length is adjusted in the next step, ZMP$_y$ can still be maintained within the area of support polygon $[-0.061, 0.061]$. The bipedal robot is at rest initially and is commanded to change its step length at $t_2 = 0.7$ s. In Figure 27, it shows that if change in step length is larger, then change in ZMP$_y$ is larger. It also shows that, in the case of the largest change in step length (0 cm to 5 cm), ZMP$_y$ is still within area of support polygon. Hence, robot can keep its dynamic balance well when desired step length is changed in the next step.

6.7. Relationship between $R_{hs,support}$, $R_{ks,support}$, $R_{hs,swing}$, and $R_{ks,swing}$ and Desired Step Length Learnt by FIS. The parameters of FLS tuned by LSE are shown in Table 5. Arbitrary desired step lengths (0.041 cm, 0.033 cm, 0.025 cm, and 0.017 cm) within a specific range (0 cm–5 cm) are commanded to the robot. In Figure 28, it shows that the robot can achieve arbitrary desired step length successfully. In Figure 29, premature landing does not occur throughout the whole walking cycle. Hence, the relationship between (1) $R_{\text{hs,support}}$, $R_{\text{ks,support}}$, $R_{\text{hs,swing}}$ and $R_{\text{ks,swing}}$, and (2) desired step length is learnt successfully.

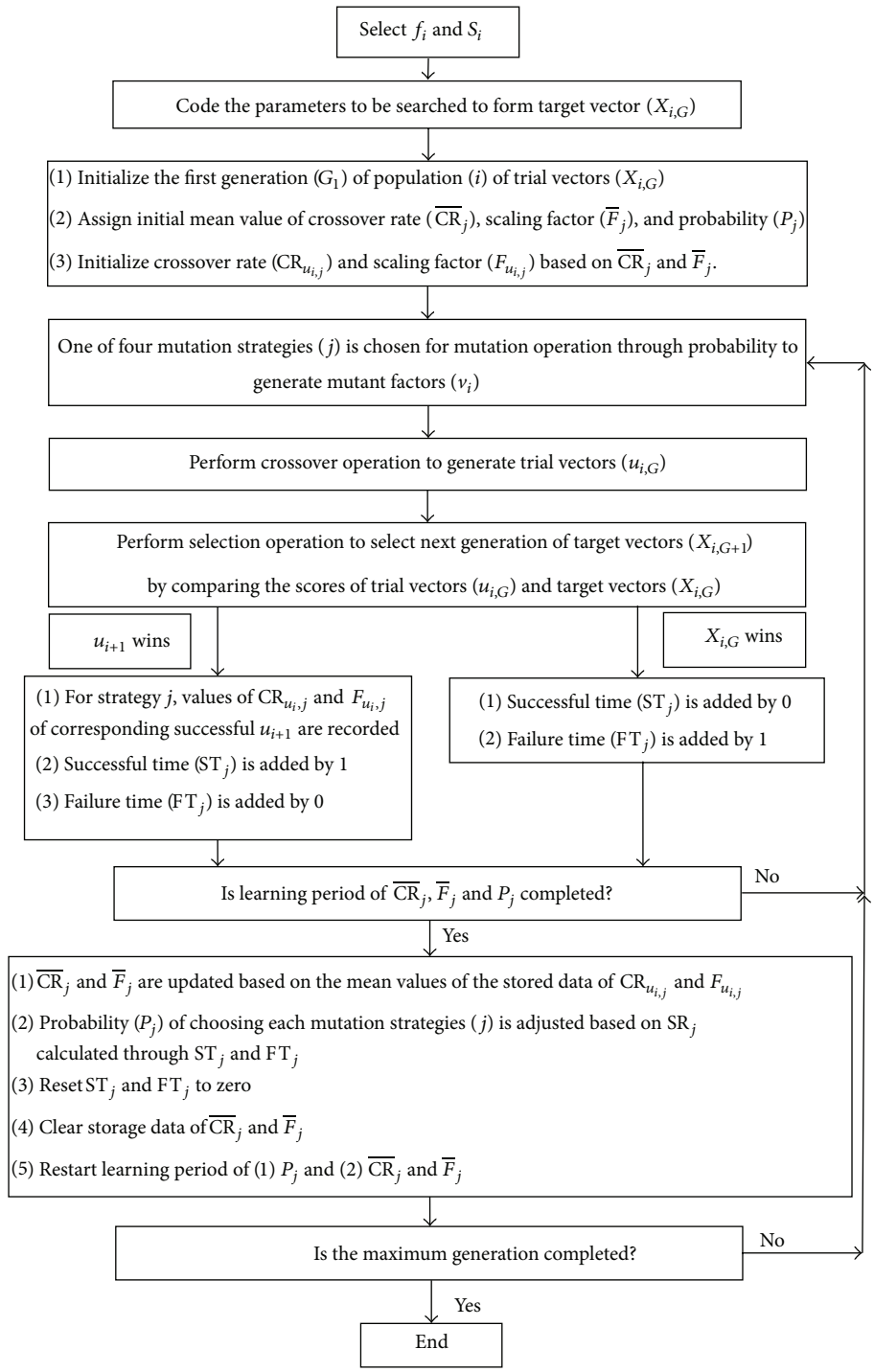

Figure 30: Flow chart of SaDE.

7. Conclusions

In this paper, GAOFSF [6] is modified to ensure that trajectories generated are continuous in angular position and its first and second derivatives. Through simulations, bipedal robot with modified CPG model yields better dynamic balance. SaDE is firstly applied to search the parameters of CPG model of bipedal walking. Performance of modified CPG model searched by SaDE is found to be satisfactory. Four adjustable parameters ($R_{\mathrm{hs,support}}$, $R_{\mathrm{ks,support}}$, $R_{\mathrm{hs,swing}}$, and $R_{\mathrm{ks,swing}}$) are added to modified CPG model and searched by SaDE. The robot is able to adjust its step length by simply changing these four factors. Instead of using look-up table [6], the relationship between (1) $R_{\mathrm{hs,support}}$, $R_{\mathrm{ks,support}}$, $R_{\mathrm{hs,swing}}$, and $R_{\mathrm{ks,swing}}$ and (2) desired step length is successfully learnt. Simulation results show that the robot is able to walk with

arbitrary desired step length within specific range (0 cm–5 cm). The desired joint angles can be obtained without the aid of inverse kinematics.

Appendix

For more details see Figure 30.

Symbols

A, B, C: Coefficients of truncated Fourier series
CR: Crossover rate
$\overline{\text{CR}}$: Mean crossover rate
f: Fitness functions
F: Mutation rate
$\overline{F}$: Mean mutation rate
g: Gravity
$H_{\text{swing foot}}$: Height of swing foot
H_{desired}: Desired height of swing foot
H_{ground}: Height of ground
m_i: Mass of link i
M: Membership function
q: Gene number = $1, 2, 3, 4, \ldots, 11$
S: Constraints
u: Trial vector
v: Mutant vector
V_{trunk}: Velocity of trunk
$\overline{V}_{\text{trunk}}$: Mean velocity of trunk
V_{strike}: Velocity of swing foot at landing time
ω_h, ω_k: Natural frequency of hip and knee trajectory
$\omega_{o/p}$: Weighting factor of fitness functions/constraints
X: Target vector
y: y-Coordinate of CoM of link i
z: z-Coordinate of CoM of link i
$\ddot{y}$: Linear acceleration in y-direction at CoM of link i
$\ddot{z}$: Linear acceleration in z-direction at CoM of link i
R: Scaling factor of truncated Fourier series
$\theta_{\text{hs/ks}}$: Hip/knee joint angle in sagittal plane
μ_{A_m}: Degree of membership function
σ: Standard deviation.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to thank the Department of Mechanical Engineering of The Hong Kong Polytechnic University for providing the research studentship.

References

[1] S. Kajita, F. Kanehiro, K. Kaneko et al., "Biped walking pattern generation by using preview control of zero-moment point," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1620–1626, September 2003.

[2] J.-Y. Kim, I.-W. Park, and J.-H. Oh, "Walking control algorithm of biped humanoid robot on uneven and inclined floor," *Journal of Intelligent and Robotic Systems*, vol. 48, no. 4, pp. 457–484, 2007.

[3] K. Erbatur, O. Koca, E. Taşkiran, M. Yilmaz, and U. Seven, "ZMP based reference generation for biped walking robots," *World Academy of Science, Engineering and Technology*, vol. 58, pp. 943–950, 2009.

[4] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.

[5] G. Taga, Y. Yamaguchi, and H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment," *Biological Cybernetics*, vol. 65, no. 3, pp. 147–159, 1991.

[6] L. Yang, C.-M. Chew, T. Zielinska, and A.-N. Poo, "A uniform biped gait generator with offline optimization and online adjustable parameters," *Robotica*, vol. 25, no. 5, pp. 549–565, 2007.

[7] N. Shafii, L. P. Reis, and N. Lau, "Biped walking using coronal and sagittal movements based on truncated Fourier series," in *Proceedings of 14th Annual Robo Cup International Symposium*, pp. 324–335, 2010.

[8] E. Yazdi, V. Azizi, and A. T. Haghighat, "Evolution of biped locomotion using bees algorithm, based on truncated Fourier series," in *Proceedings of the World Congress on Engineering and Computer Science*, pp. 378–382, 2010.

[9] J. Or, "A hybrid CPG-ZMP control system for stable walking of a simulated flexible spine humanoid robot," *Neural Networks*, vol. 23, no. 3, pp. 452–460, 2010.

[10] S. Aoi and K. Tsuchiya, "Adaptive behavior in turning of an oscillator-driven biped robot," *Autonomous Robots*, vol. 23, no. 1, pp. 37–57, 2007.

[11] Y. Farzaneh, A. Akbarzadeh, and A. A. Akbaria, "Online bio-inspired trajectory generation of seven-link biped robot based on T-S fuzzy system," *Applied Soft Computing*, 2013.

[12] D. Gong, J. Yan, and G. Zuo, "A review of gait optimization based on evolutionary computation," *Applied Computational Intelligence and Soft Computing*, vol. 2010, Article ID 413179, 12 pages, 2010.

[13] H. Inada and K. Ishii, "Bipedal walk using a central pattern generator," in *Proceedings of International Congress Series*, pp. 185–188, 2004.

[14] K. Miyashita, S. Ok, and K. Hase, "Evolutionary generation of human-like bipedal locomotion," *Mechatronics*, vol. 13, no. 8-9, pp. 791–807, 2003.

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

[16] B. Hegery, C. C. Hung, and K. Kasprak, "A comparative study on differential evolution and genetic algorithms for some combinatorial problems," in *Proceedings of 8th Mexican International Conference on Artificial Intelligence*, 2009.

[17] T. Tušar and B. Filipič, "Differential evolution versus genetic algorithms in multi-objective optimization," in *Proceedings of*

4th International Conference on Evolutionary Multi-Criterion Optimization, 2007.

[18] J. Vesterstrøm and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 1980–1987, June 2004.

[19] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC '05)*, pp. 1785–1791, September 2005.

[20] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[21] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Self-adaptive differential evolution," in *Proceedings of International Conference on Computational Intelligence and Security*, pp. 192–199, 2005.

[22] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[23] L. Yang, C.-M. Chew, and A.-N. Poo, "Real-time bipedal walking adjustment modes using truncated fourier series formulation," in *Proceedings of the 7th IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS '07)*, pp. 379–384, December 2007.

[24] C. Hern ndez-Santos, E. Rodriguez-Leal, R. Soto, and J. L. Gordillo, "Kinematics and dynamics of a new 16DOF Humanoid biped robot with active toe joint," *International Journal of Advanced Robotic Systems*, vol. 9, no. 190, 2012.

[25] K.-C. Choi, H.-J. Lee, and M. C. Lee, "Fuzzy posture control for biped walking robot based on force sensor for ZMP," in *Proceedings of the SICE-ICASE International Joint Conference*, pp. 1185–1189, October 2006.

[26] D. Tlalolini, Y. Aoustin, and C. Chevallereau, "Design of a walking cyclic gait with single support phases and impacts for the locomotor system of a thirteen-link 3D biped using the parametric optimization," *Multibody System Dynamics*, vol. 23, no. 1, pp. 33–56, 2010.

[27] A. P. Sudheer, R. Vijayakumar, and K. P. Mohanda, "Stable gait synthesis and analysis of a 12-degree of freedom biped robot in sagittal and frontal planes," *Journal of Automation, Mobile Robotics and Intelligent System*, vol. 6, no. 4, pp. 36–44, 2012.

[28] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.