

Research Article

Optimization of Central Pattern Generator-Based Torque-Stiffness-Controlled Dynamic Bipedal Walking

William Suliman , Chadi Albitar , and Lama Hassan 

Departement of Electronic and Mechanical Systems, Higher Institute for Applied Sciences and Technology, Damascus, Syria

Correspondence should be addressed to William Suliman; william.suliman@hiast.edu.sy

Received 31 July 2019; Accepted 8 November 2019; Published 23 January 2020

Academic Editor: L. Fortuna

Copyright © 2020 William Suliman et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we propose a central pattern generator-based model to control the walking motion of a biped robot. The model independently controls the joint torque and joint stiffness in real time. Instead of the phase-dependent neural model used by Huang in 2014, we adopt the same structure for all the walking phases, reducing the number of connections between neurons. This reduction enables the employment of the particle swarm algorithm to find the optimal values of these parameters which lead to different solutions with different performance criteria. The simulation of the proposed method on a seven-link bipedal walking model gave a good performance in the range of walking speeds, which is referred to as versatility, and in walking pattern transition. The achieved walking gaits are 1-period cyclic motions for all the input control signals except for few gaits. Besides, these 1-period cyclic motions have a good local and global stability. Finally, we expanded our neural model by adding connections that work only when the robot walks on uneven terrains, which improved the robot's performance against this kind of perturbation.

1. Introduction

The concept of central pattern generators (CPGs) is commonly used in biology. CPGs are neuronal circuits found in the spinal cord of vertebrates and in the segmental ganglia in invertebrates [1]. These circuits have the ability to generate rhythmic outputs suitable for periodic motions such as moving legs to perform walking motion in quadrilateral legs. The level of activity of these neurons is determined by one signal from the upper nervous system. Several models were designed to mathematically represent these generators [2–4], and their rhythmic outputs were used to control several biomimetic robots [5–7].

The significant reduction in the number of control signals, the transitional ability between different motion patterns using a limited number of control signals [1], and the robustness against disturbances [2] encouraged the use of these models in passivity-based biped robots which are based on the concept of passive dynamic walking [8]. The output of CPGs was used to generate periodic signals to control the joints torques [2], angles [9], angular velocity [10], or the Cartesian coordinates of robot's link [11]. Huang

et al. [12] presented a CPG-based control system to control joint torque and stiffness of a seven-link passivity-based biped robot. CPG-based torque-stiffness-controlled dynamic walking exhibits a good performance in walking pattern transition, versatility, disturbance rejection, and energy efficiency.

Despite the simplicity of the control and the reduction in the input signals, the number of parameters needed to generate stable gaits is significant. Several methods were proposed to find the appropriate values for these parameters, based on evolutionary algorithms such as genetic algorithm [10, 13] and particle swarm optimization (PSO) algorithm [13] and reinforcement learning [14].

In this work, we propose a simpler version of the CPG-based model provided by Huang et al. [12] to control a 7-link 6-DoF biped robot. This model is a phase-independent model, and thus, it considers constant parameters for all walking phases instead of using different parameters for each phase. The optimized parameters values are determined using the particle swarm optimization algorithm. We also discuss the importance of adding connections to the neural model to increase the robot's ability to walk on uneven terrain.

This paper is organized as follows: biped model and walking phases are described in Section 2. CPG-based control system is presented in Section 3. Experimental results are depicted in Section 4, and the conclusion is in Section 5.

2. Biped Model and Walking Dynamics

2.1. Biped Model. We adopted a model with seven links, as shown in Figure 1: an upper body and two legs; each leg consists of a thigh, a shank, and a flat foot. The model contains six joints: two hip joints, two knees, and two ankles. The stiffness and equilibrium angle of each joint is controllable except for the knee joints which are considered with a constant stiffness. The friction between the ground and the foot is assumed to be sufficient so that no slippage occurs during the movement, and the collision with the ground is a fully inelastic instantaneous collision that does not result in any slip or bounce.

During walking, the robot's leg passes through different phases, which differ in the number of ground contact points and in the number of degrees of freedom. The robot's leg is called "stance leg" when it has at least one contact point with the ground. In contrast, the leg with no ground contact is called a "swing leg." Depending on the phases of its legs, the robot has different phases. The phase with two legs on the ground is called a double stance phase, while the phase with only one leg touching the ground is called the swing phase. When the walking phases are properly sequenced, the robot generates stable motions called "gait." Walking motions have several gaits depending on the followed scenarios. Adopted gaits have been studied in detail in [15]. Our study focuses on two gaits which are closer to human walking with a moderate speed [15]. The phases of these two gaits are as follows: (1) push-off phase: the robot rotates around two contact points (the heel of the leading leg and the toe of the trailing leg). This phase ends when the trailing leg leaves the ground. (2) Swing phase with free knee: during this phase, the knee constraint in the swing leg is released. The swing leg moves forward until the shank becomes straight with the thigh, and then, the knee collision occurs. (3) Knee collision: after this collision, the knee joint is locked until the next swing phase of this leg. (4) Swing phase with a locked knee: during this phase, the thigh and the shank of the swing leg moves together until the heel of this leg strikes the ground. The heel of the stance leg might leave the ground before the swing foot hits the ground, and this rise is called "premature heel rise." [16] Premature heel rise moves the robot to the second gait, while the absence of this rise holds the robot in the first gait. (5) Heel collision: the heel of the swing leg strikes the ground. (6) Double stance phase: the heel of the leading leg with the toe or all the foot of the trailing leg is in contact with the ground. This phase continues until the toe of the leading leg touches the ground. (7) Foot collision: the whole foot of the leading leg strikes the ground. Repeating phases according to the previous scenario guarantee stable robot motion.

2.2. Walking Dynamics. Here, we tackle the equations of motion (EoM) considering the mechanical parameters

adopted in [12]. The model is described using the Euclidean coordinates, where the x -axis is parallel to the ground and heading forward, while the y -axis is vertical to the ground upwards. Each link is described by the coordinates of its center of mass (CoM) and its angle with the y -axis for the thighs and the shanks or its angle with the x -axis for the feet. Thus, the model's Euclidean coordinates can be written as follows:

$$x = [x_h, y_h, x_b, y_b, \theta_b, x_{t1}, y_{t1}, \theta_{t1}, x_{s1}, y_{s1}, \theta_{s1}, x_{f1}, y_{f1}, \theta_{f1}, x_{t2}, y_{t2}, \theta_{t2}, x_{s2}, y_{s2}, \theta_{s2}, x_{f2}, y_{f2}, \theta_{f2}]. \quad (1)$$

The model is also described by the generalized coordinates:

$$q = [x_h, y_h, \theta_b, \theta_{t1}, \theta_{s1}, \theta_{f1}, \theta_{t2}, \theta_{s2}, \theta_{f2}]. \quad (2)$$

The coordinates change according to the walking phases. For example, when the knee is locked, the angle and the angular velocity of the thigh and of the shank are equal.

Euclidean coordinates are related with the generalized coordinates by

$$\dot{x} = \dot{x}(q) = \frac{\partial x}{\partial q} \dot{q} = J\dot{q}, \quad (3)$$

where J is the Jacobian matrix.

We define the constraints vector $\xi(q)$ which contains the constraints with the ground and the constraints of the locked knee. The previous constraints is guaranteed by the equality: $\xi(q) = 0$.

Lagrange's equations of the first kind could be used to get the EoM [12, 15]:

$$M_q \ddot{q} = F_q + T^T F_f, \quad (4)$$

$$\xi(q) = 0, \quad (5)$$

where $T = (\partial \xi(q)/\partial q)$, M_q is the mass matrix written in the generalized coordinates, F_q is the external active forces acted on the robot written in the generalized coordinates, and F_f is the constraint force vector.

Rearranging equation (5) as follows:

$$T\ddot{q} + \frac{\partial(T\dot{q})}{\partial q} \dot{q} = 0, \quad (6)$$

and from equations (4) and (6), we get

$$\begin{bmatrix} M_q & -T^T \\ T & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_f \end{bmatrix} = \begin{bmatrix} F_q \\ -\frac{\partial(T\dot{q})}{\partial q} \dot{q} \end{bmatrix}. \quad (7)$$

All the walking dynamics phases share the same dynamics form as equation (7), while the component of each matrix may be different during different phases.

The collision is supposed to be a fully inelastic instantaneous collision. Thus, the model achieves the constraints vector after the collision, and we write

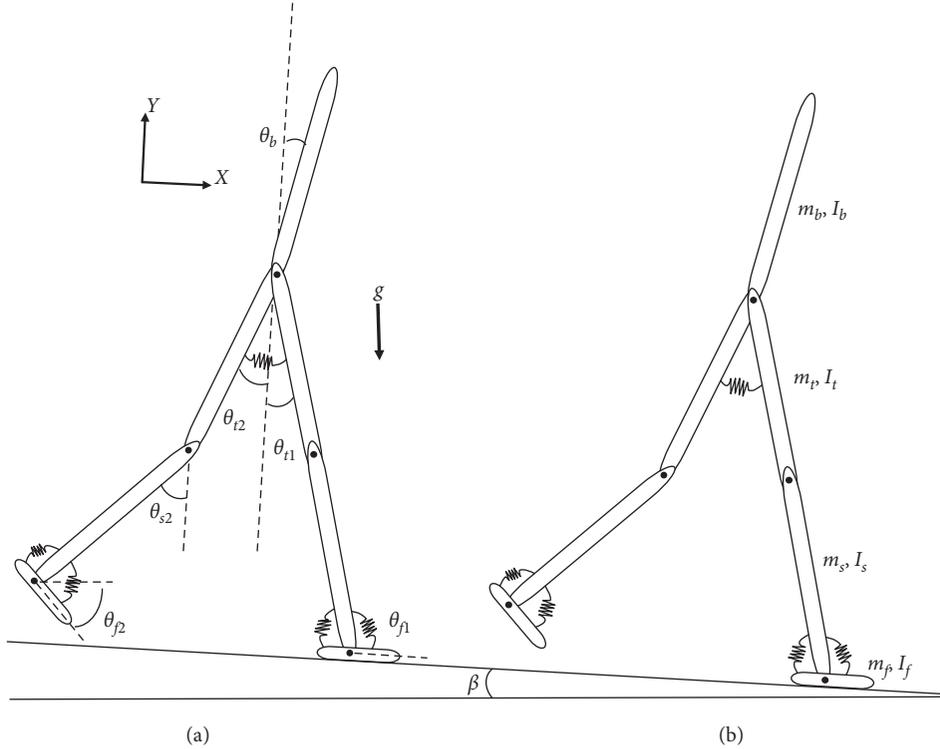


FIGURE 1: 7-link bipedal walking model: (a) the degree of freedom; (b) the mass distribution and the torsional spring of the model.

$$\frac{\partial \xi(q)}{\partial q} \dot{q}^+ = 0. \quad (8)$$

The equation of strike moment can be obtained by integrating equation (4):

$$M_q \dot{q}^+ = M_q \dot{q}^- + T^T I_f, \quad (9)$$

where \dot{q}^- and \dot{q}^+ are the velocities in the generalized coordinates before and after the collision, respectively. I_f is the impulse force acted on the robot at the collision moment.

From equations (8) and (9), we write the collision equations as

$$\begin{bmatrix} M_q & -T^T \\ T & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ I_f \end{bmatrix} = \begin{bmatrix} M_q \dot{q}^- \\ 0 \end{bmatrix}. \quad (10)$$

3. CPG-Based Control System

3.1. Model Structure. Previous studies [17–19] demonstrated the importance of changing the internal dynamic of a biped robot by using adaptable compliance actuators in walking gaits transitions, control of speed, and energy efficiency.

Huang et al. [12] adopted the biologically inspired control method based on CPG in such a way that the joint stiffness and the joint torque could be independently controlled in real time. This method helped in improving the smoothness and the time of gait transitions, and it increased the disturbance rejection with good energy efficiency. This is why we adopted this strategy; therefore, the torque of each joint can be defined by the following equation:

$$T_i = -(K_{is} + K_i)(\theta_i - \tilde{\theta}_i^e) - d_i \dot{\theta}_i, \quad (11)$$

where K_{is} is the joint stiffness nominal value, K_i is the output of the neural oscillator responsible for controlling the joint stiffness, $\tilde{\theta}_i^e$ is the output of the neural oscillator responsible for controlling the joint torque by controlling the joint equilibrium angle, $\dot{\theta}_i$ is the angular velocity of the i th joint, and d_i is the damping coefficient.

The oscillator output is not applied directly to the dynamic model, and instead the response of the motors which control the joint stiffness and the equilibrium angle is taken into account. This response is modeled by a first-order response function with a time constant equal to $\tau_a = 0.1$ sec. The neural model receives information from the dynamic model including the state of the leg (stance or swing) and the relative joint angle and its relative angular velocity. Figure 2 shows the schematic diagram of the proposed coupled neural oscillators.

Our model is based on 9 neurons in total. Five neurons control the swing leg with free knee, and four neurons control the stance leg. Here, we consider that the torque and the stiffness of each joint are controllable except for the knee joint, which is considered as a constant-stiffness joint. The effect of the knee's neuron is eliminated when the knee is locked at the end of the swing phase. Furthermore, the foot of the swing leg needs to be fixed with the shank so that the equilibrium joint for this joint is kept constant during the swing phase with free knee, and then it is diminished after the knee collision until the heel of this leg strikes the ground. The neural model receives two input signals: u_e controls the joint's equilibrium position and thus the joint's torque, and

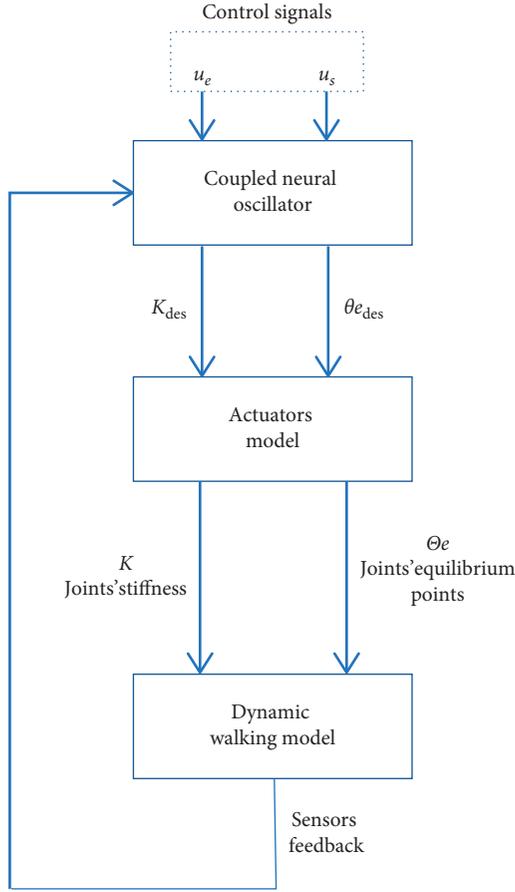


FIGURE 2: The diagram of the proposed control scheme.

u_s controls the joint's stiffness. Interlimb coordination is established by adding inhibitory connections between the unit oscillators of the hip joints in each leg; the knee and foot neurons in each leg are affected by connections from the hip joint in the same leg. No connection was added between the unit oscillators for the torque control and those for the stiffness control in order to keep their changes independent [12]. Figure 3 shows the proposed control system.

We used the model submitted by Matsuoka [20] to mathematically represent the unit oscillators. Therefore, the equations of the torque oscillators are

$$\begin{aligned} \dot{u}_i^e &= \frac{1}{\tau_i^e} \left(c_i^e \tilde{u}_i^e - u_i^e + \sum_{j,j \neq i} \omega_{ij}^e \tilde{\theta}_j - \beta^e v_i^e + \text{Feed}_i^e(q, \dot{q}) \right), \\ \dot{v}_i^e &= \frac{1}{\tau_i^{e'}} (-v_i^e + \tilde{\theta}_i), \\ \tilde{\theta}_i &= \begin{cases} -\frac{\pi}{2}, & u_i^e < -\frac{\pi}{2}, \\ u_i^e, & -\frac{\pi}{2} \leq u_i^e \leq \frac{\pi}{2}, \\ \frac{\pi}{2}, & u_i^e > \frac{\pi}{2}, \end{cases} \\ & i = 1, 2, \dots, 5, \end{aligned} \quad (12)$$

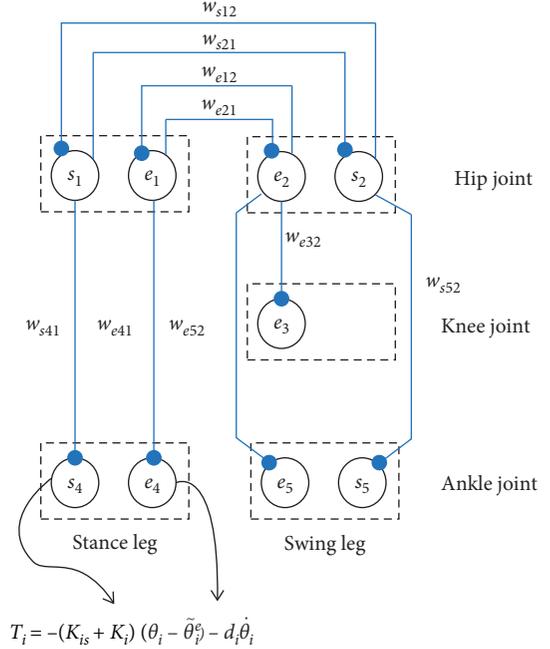


FIGURE 3: Each coupled oscillator sends joint torque and joint stiffness commands. w_{ij}^e and w_{ij}^s represent the weights of inter-connections between the neurons.

where the index i represents the joint number shown in Figure 3, $\tilde{\theta}_j$ is the torque oscillator output (the equilibrium position), \tilde{u}_i^e is the input control signal, τ_i^e and $\tau_i^{e'}$ are time constants, β^e is the coefficient of adaptation effect for equilibrium position control, c_i^e is the weight of the input signal that affects the oscillator i , ω_{ij}^e is the connections weight, and $\text{Feed}_i^e(q, \dot{q})$ is a vector with all the feedback signals from the dynamic system which includes information about the relative angle and the relative angular velocity of the affected joint.

The equations of the stiffness oscillators are

$$\begin{aligned} \dot{u}_i^s &= \frac{1}{\tau_i^s} \left(c_i^s \tilde{u}_i^s - u_i^s + \sum_{j,j \neq i} \omega_{ij}^s k_j - \beta^s v_i^s + \text{Feed}_i^s(q, \dot{q}) \right), \\ \dot{v}_i^s &= \frac{1}{\tau_i^{s'}} (-v_i^s + k_i), \\ k_i &= \begin{cases} -10, & u_i^s < -10, \\ u_i^s, & -10 \leq u_i^s \leq 10, \\ 10, & u_i^s > 10. \end{cases} \end{aligned} \quad (13)$$

All the parameters in the previous equations have the same meaning as in equation (12). We restricted the output of the stiffness neurons in the range $[-10, 10]$ because these values represent the offsets from the nominal values of the joint stiffness.

3.2. *Model's Parameters.* From Figure 3, equation (12) can be rewritten as follows:

$$\begin{pmatrix} \tau_{\text{hip}}^e \dot{u}_1^e \\ \tau_{\text{hip}}^e \dot{u}_2^e \\ \tau_{\text{knee}}^e \dot{u}_3^e \\ \tau_{\text{foot}}^e \dot{u}_4^e \\ \tau_{\text{foot}}^e \dot{u}_5^e \end{pmatrix} = \tilde{u}^e \begin{pmatrix} c_1^e \\ c_2^e \\ c_3^e \\ c_4^e \\ c_5^e \end{pmatrix} - \begin{pmatrix} u_1^e \\ u_2^e \\ u_3^e \\ u_4^e \\ u_5^e \end{pmatrix} - \beta^e \begin{pmatrix} v_1^e \\ v_2^e \\ v_3^e \\ v_4^e \\ v_5^e \end{pmatrix} \\ + \begin{pmatrix} 0 & \omega_{12}^e & 0 & 0 & 0 \\ \omega_{21}^e & 0 & 0 & 0 & 0 \\ 0 & \omega_{32}^e & 0 & 0 & 0 \\ \omega_{41}^e & 0 & 0 & 0 & 0 \\ 0 & \omega_{52}^e & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \tilde{\theta}_1 \\ \tilde{\theta}_2 \\ \tilde{\theta}_3 \\ \tilde{\theta}_4 \\ \tilde{\theta}_5 \end{pmatrix} \\ + \begin{pmatrix} g_{ep}^{\text{hip}}(\theta_{t1} - \theta_b) + g_{ed}^{\text{hip}}(\dot{\theta}_{t1} - \dot{\theta}_b) \\ g_{ep}^{\text{hip}}(\theta_{t2} - \theta_b) + g_{ed}^{\text{hip}}(\dot{\theta}_{t2} - \dot{\theta}_b) \\ g_{ep}^{\text{knee}}(\theta_{s2} - \theta_{t2}) + g_{ed}^{\text{knee}}(\dot{\theta}_{s2} - \dot{\theta}_{t2}) \\ g_{ep}^{\text{foot}}(\theta_{f1} - \theta_{s1}) + g_{ed}^{\text{foot}}(\dot{\theta}_{f1} - \dot{\theta}_{s1}) \\ g_{ep}^{\text{foot}}(\theta_{f2} - \theta_{s2}) + g_{ed}^{\text{foot}}(\dot{\theta}_{f2} - \dot{\theta}_{s2}) \end{pmatrix},$$

$$\begin{pmatrix} \tau_{\text{hip}}^{\prime e} \dot{v}_1^e \\ \tau_{\text{hip}}^{\prime e} \dot{v}_2^e \\ \tau_{\text{knee}}^{\prime e} \dot{v}_3^e \\ \tau_{\text{foot}}^{\prime e} \dot{v}_4^e \\ \tau_{\text{foot}}^{\prime e} \dot{v}_5^e \end{pmatrix} = - \begin{pmatrix} v_1^e \\ v_2^e \\ v_3^e \\ v_4^e \\ v_5^e \end{pmatrix} + \begin{pmatrix} \tilde{\theta}_1 \\ \tilde{\theta}_2 \\ \tilde{\theta}_3 \\ \tilde{\theta}_4 \\ \tilde{\theta}_5 \end{pmatrix},$$

$$\tau_i^{\prime e} = \frac{\tau_i^e}{2}. \quad (14)$$

In the same way, equation (13) can be rewritten as follows:

$$\tau^s \begin{pmatrix} \dot{u}_1^s \\ \dot{u}_2^s \\ \dot{u}_4^s \\ \dot{u}_5^s \end{pmatrix} = \tilde{u}^s \begin{pmatrix} c_1^s \\ c_1^s \\ c_2^s \\ c_2^s \end{pmatrix} - \begin{pmatrix} u_1^s \\ u_2^s \\ u_4^s \\ u_5^s \end{pmatrix} - \beta^s \begin{pmatrix} v_1^s \\ v_2^s \\ v_4^s \\ v_5^s \end{pmatrix} \\ + \begin{pmatrix} 0 & \omega_{12}^s & 0 & 0 \\ \omega_{21}^s & 0 & 0 & 0 \\ \omega_{41}^s & 0 & 0 & 0 \\ 0 & \omega_{52}^s & 0 & 0 \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \\ k_4 \\ k_5 \end{pmatrix} \\ + \begin{pmatrix} g_{sp}^{\text{hip}}(\theta_{t1} - \theta_b) + g_{sd}^{\text{hip}}(\dot{\theta}_{t1} - \dot{\theta}_b) \\ g_{sp}^{\text{hip}}(\theta_{t2} - \theta_b) + g_{sd}^{\text{hip}}(\dot{\theta}_{t2} - \dot{\theta}_b) \\ g_{sp}^{\text{foot}}(\theta_{f1} - \theta_{s1}) + g_{sd}^{\text{foot}}(\dot{\theta}_{f1} - \dot{\theta}_{s1}) \\ g_{sp}^{\text{foot}}(\theta_{f2} - \theta_{s2}) + g_{sd}^{\text{foot}}(\dot{\theta}_{f2} - \dot{\theta}_{s2}) \end{pmatrix}, \\ \tau^{\prime s} \begin{pmatrix} \dot{v}_1^s \\ \dot{v}_2^s \\ \dot{v}_4^s \\ \dot{v}_5^s \end{pmatrix} = - \begin{pmatrix} v_1^s \\ v_2^s \\ v_4^s \\ v_5^s \end{pmatrix} + \begin{pmatrix} k_1 \\ k_2 \\ k_4 \\ k_5 \end{pmatrix}, \\ \tau^{\prime s} = \frac{\tau^s}{2}. \quad (15)$$

Here, the time constants are assumed to be equal for all the stiffness neurons.

However, we may notice that the number of parameters needed to evaluate stable gaits is significant which makes reaching the appropriate parameters by trial and error an uneasy task. Therefore, we employed the PSO algorithm to find the optimal values.

The PSO algorithm is an iterative statistical technique for nonlinear function optimization. It does not rely on the gradient, and instead it uses random values to avoid convergence to local minima. The algorithm was first introduced in [21], and for our problem, we adopt the proposed amendments in [22, 23]. The algorithm starts by generating a set of particles within the working space, assigning initial velocity values to those particles and calculating their respective costs. The positions and velocities of particles are modified during the successive iterations based on the value of a predefined cost function of their neighbors from the previous iteration and based on the best value of these particles during all iterations.

4. Experimental Results

In this section, we present the results of the simulations that were carried out in MATLAB 2017 environment based on the EoM mentioned in Section 2.2. We adopted the mechanical parameters of the biped model used in [12, 15].

4.1. Model's Parameter Optimization Using PSO Algorithm.

The parameters of our neural model include, as shown from equations (14) and (15), the neurons interconnection weights (9 parameters), the self-parameters of each neuron with the two input signals (u_e, u_s) (13 parameters), the sensory feedback weights (10 parameters), the nominal values of the joint stiffness (3 parameters), and the constant foot equilibrium joint during the free swing phase. In addition, we need five parameters related to imposed conditions on the initial state of the walker. These conditions are used to guarantee that the motion begins at the moment just after the robot rises his trailing leg from the ground, the upper body at this moment is in midway between the two thighs, and the leading leg is in full contact with the ground. The parameters include the hip angle of the rear leg θ_{t2} and its angular velocity $\dot{\theta}_{t2}$, the rear foot angle θ_{f2} and its angular velocity $\dot{\theta}_{f2}$, and the leading hip angular velocity $\dot{\theta}_{t1}$. Therefore, a total of 39 parameters need to be optimized. Table 1 presents these parameters with the upper and lower bounds. The parameters range was inspired by a logical approach and some initial tests.

The cost function is defined as follows:

$$F = \frac{Nb \text{ of steps}}{1 + e_1 + e_2 + e_3}, \quad (16)$$

where Nb of steps is the number of steps achieved by the robot; this number is limited by a predefined maximum. $e_1 = a_1 \cdot (v < v_0) \cdot |v - v_0|$ guarantees that the walking velocity v exceeds a certain minimum limit v_0 . $e_2 = a_2 \cdot \text{div}(\theta_b)$ is related to the standard deviation of the body angle during walking. $e_3 = a_3 \cdot \text{max clearance}$ eliminates the foot scuffing of the swing leg with the ground. The coefficients a_i , $1 \leq i \leq 3$, are the weights which determine the importance of each factor in our optimization problem. The self-parameters of the PSO algorithm are shown in Table 2.

By applying the PSO algorithm several times, different solutions have been obtained according to the optimized factors e_i . These solutions differ in the gait patterns, in the versatility, and in the robustness against disturbances. Table 1 shows the optimal values of the best solution, and Figure 4 shows the stick diagram of a stable gait generated for $u_e = 2.4$ and $u_s = 22$.

4.2. Walking Patterns. Based on the parameter values shown in Table 1, the range of walking speed was studied as a function of the two control signals. Figure 5 shows the variations of walking velocities as a function of the control signals. We notice a forward acceleration by increasing u_e and u_s , which can be justified by the increase in the system's input energy. The velocity ranges from $0.34 \text{ m}\cdot\text{s}^{-1}$ to $1.2 \text{ m}\cdot\text{s}^{-1}$ which corresponds to a Froude number (Fr) in the range $[0.12, 0.43]$.

TABLE 1: Neural model parameters, bounds of the PSO algorithm, and the resulted optimized values.

Parameter	Upper bound	Lower bound	Optimized value
$K_{s\text{-hip}}$	30	70	67
$K_{s\text{-knee}}$	5	15	6
$K_{s\text{-feet}}$	30	50	34
\bar{u}^e	1	5	2.4
\bar{u}^s	5	50	22
β^e, β^s	1	5	3.9, 1.3
$\tau_{\text{hip}}^e, \tau_{\text{knee}}^e$	0	1	0.12, 0.4
$\tau_{\text{foot}}^e, \tau^s$	0	1	0.8, 0.66
$\omega_{12}^s, \omega_{21}^s$	-0.5	0.5	-0.07, 0.32
$\omega_{41}^s, \omega_{52}^s$	-0.5	0.5	-0.09, -0.34
$\omega_{22}^e, \omega_{21}^e$	-5	5	-3.9, 1.4
$\omega_{32}^e, \omega_{41}^e, \omega_{52}^e$	-5	5	1.4, 1.4, 0.28
c_1^e, c_2^e	-1	1	0.97, 1
c_3^e, c_4^e, c_5^e	-1	1	0.28, 0.24, -0.16
c_1^s, c_2^s	-1	1	0.94, 0.55
$g_{ep}^{\text{hip}}, g_{ed}^{\text{hip}}$	-1	1	-0.8, 0
$g_{ep}^{\text{knee}}, g_{ed}^{\text{knee}}$	-1	1	0, 0.7
$g_{ep}^{\text{foot}}, g_{ed}^{\text{foot}}$	-1	1	-0.8, 0.2
$g_{sp}^{\text{hip}}, g_{sp}^{\text{foot}}$	-100	100	72.6, -66
$g_{sd}^{\text{hip}}, g_{sd}^{\text{foot}}$	-100	100	57, 5
θ_{t2}	-0.7	-0.4	-0.419
$\dot{\theta}_{t2}$	-1.5	-0.2	-1.5
$\dot{\theta}_{t1}$	-1.5	-0.2	-1.6
θ_{f2}	-0.7	-0.4	-0.45
$\dot{\theta}_{f2}$	-1.5	-0.2	-1.6

TABLE 2: The configuration of the PSO algorithm parameters.

Parameter	Value
W	0.1
c_1	1.49
c_2	1.49
Min neighbors fraction	0.25
Crossover type	Stochastic
Encoding type	Real
Selection strategy	Stochastic
Swarm size	700

Fr is defined as $V/\sqrt{g \cdot l}$ where V is the robot speed, g is the gravitational acceleration, and l is the leg length [12].

The speed contours in Figure 5 show that similar speed values could be obtained by taking several input values, where the difference between these gaits is in the step length and frequency, which can be also noticed on Figures 6 and 7.

Figure 6 shows that u_e is mainly responsible for the changes in the step length especially in the high values of u_s , while Figure 7 shows that the changes in the step frequency depend mainly on u_e for the low values of u_e , while it depends on the value of u_s for the moderate value of u_e and u_s . For the large value of u_s , the frequency is almost constant, and this is because the changes in the stiffness were restricted to the range $[-10, 10]$, which limited the effect of u_s .

4.3. Walking Speed Control. We studied the transitional ability of the simulated walker between different patterns

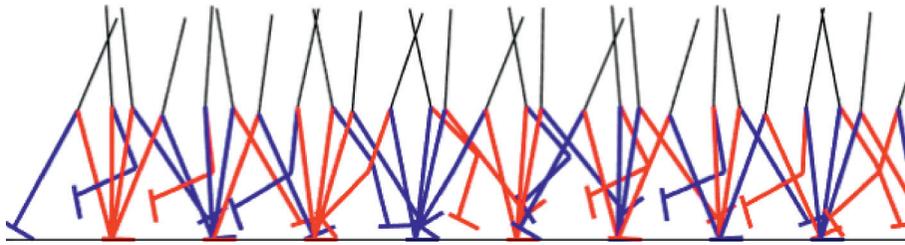


FIGURE 4: Stick diagram of a representative stable gait for $u_e = 2.4$ and $u_s = 22$.

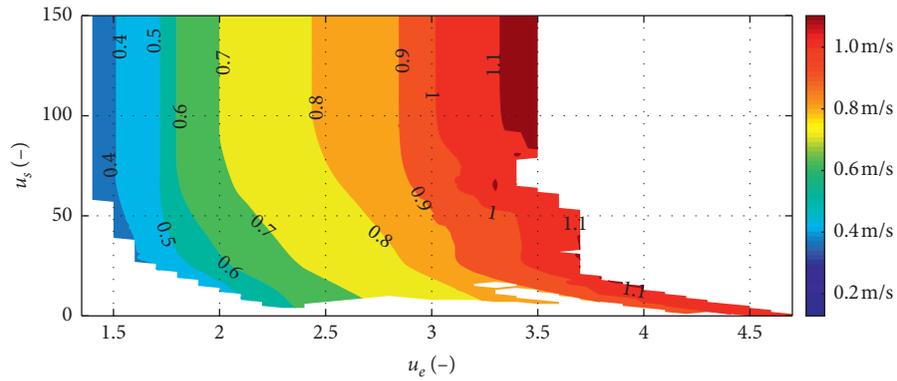


FIGURE 5: Walking velocities as a function of control parameters (u_e, u_s).

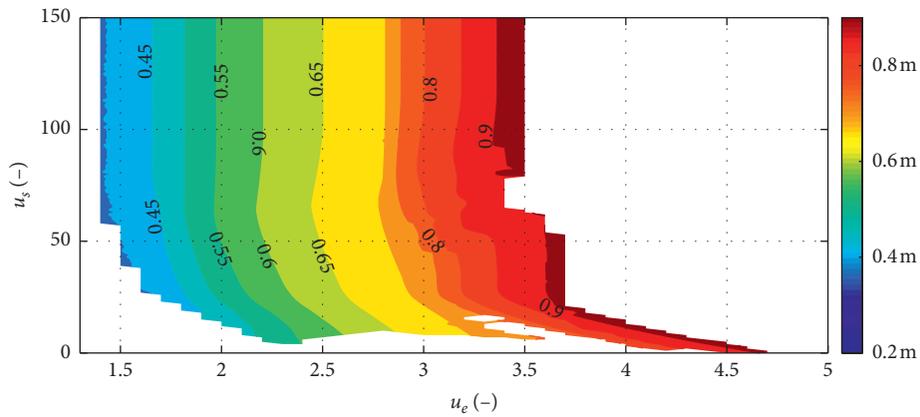


FIGURE 6: Step length as a function of the control parameters (u_e, u_s).

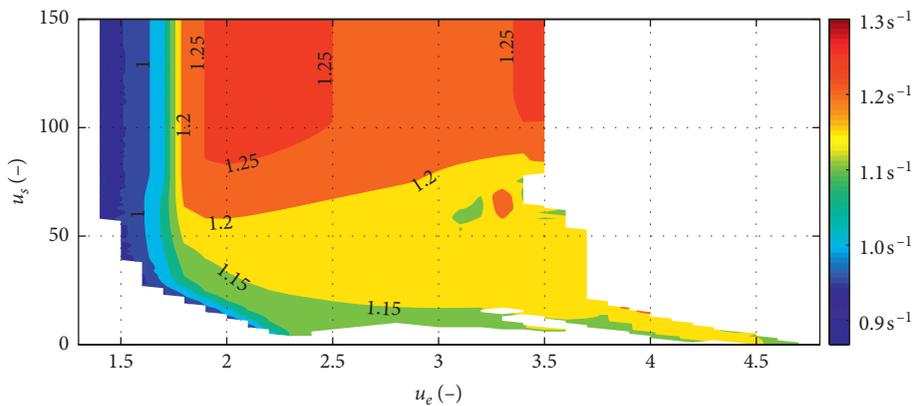


FIGURE 7: Step frequency as a function of the control parameters (u_e, u_s).

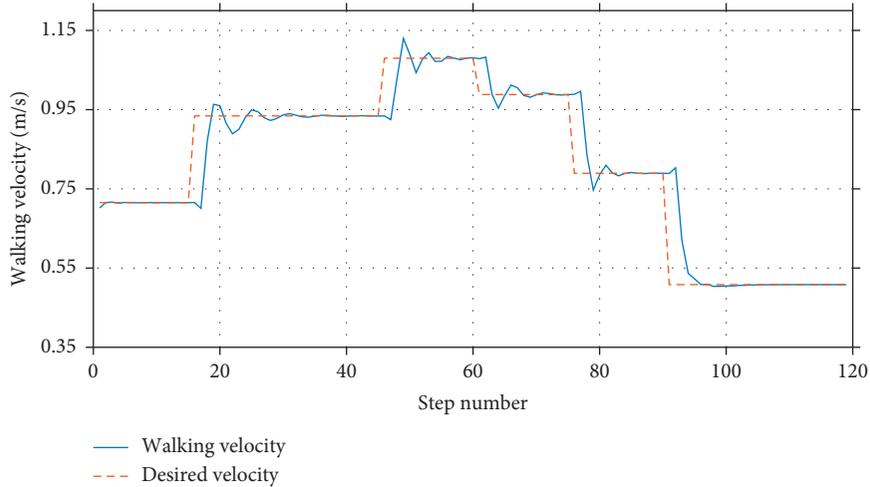


FIGURE 8: Following a predefined speed profile. The overshoot is less than 5% in all the transitions, and it takes three to four steps before the robot settles to its new speed.

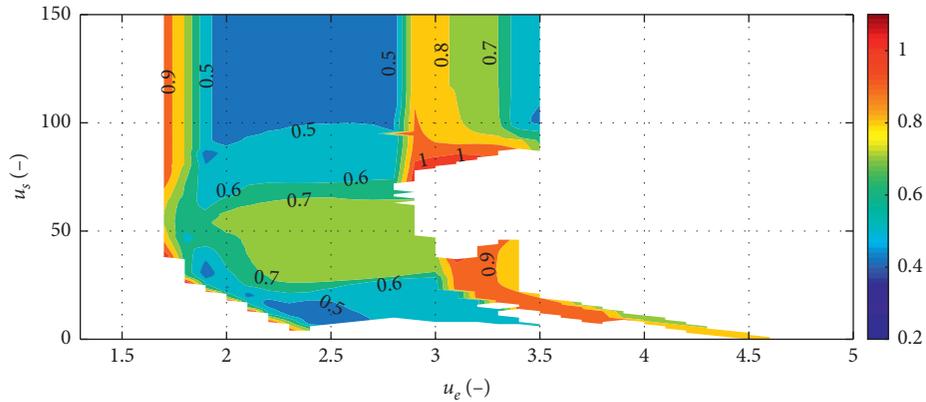


FIGURE 9: The local stability represented by the maximum eigenvalue of the Jacobian matrix.

by changing the control signals depending on former exploration of the walking space of the robot. To this end, a speed profile was defined, and the ability of the walker to change its velocity following this profile by changing the input signals at the beginning of the push-up phase was studied. We observed in Figure 8 the flexibility and the ability of the robot in following the speed profile. The overshoot was less than 5% in all the transitions, and it took three to four steps before the robot settled to its new speed value.

4.4. Stability Analysis. The stability of the walker is analyzed by observing its motion on a step-to-step basis. One step is considered as a function called “stride function $f(u)$ ” [16] which maps the robot’s state $u_m(q, \dot{q})$ at a defined point of a step m with its state at the same point in the next step $m + 1$. Stable walking is achieved when the stride function converges after a few steps to a single point called “fixed point,” or when it alternates between n points. The trajectory of the robot in the state space is called the n -period limit cycle, where n is the number of the previous points. The cyclic stability of this motion is determined by linearizing the

stride function at its fixed points [16, 24]. Assume small deviation Δu_n

$$\Delta u_{n+1} = u^* + J\Delta u_n, \quad (17)$$

where J is the Jacobian matrix calculated using the Newton–Raphson method [24]. When the eigenvalues of the Jacobian matrix are within the unit circle in the complex plane, the cyclic motion of the robot is stable and it can handle small perturbations around its nominal trajectory [16]. The local stability increases when the magnitude of the maximum eigenvalue λ_m is closer to zero.

Figure 9 shows the control signals that generate 1-period cyclic motions and the relative $|\lambda_m|$ at each point. We notice that the motion is stable for the most input control signals.

In addition to the local stability, the global stability of the walker with the CPG-based control system was also considered. The global stability is determined by the basin of attraction (BoA) which is defined by all the initial conditions that lead to stable motions of the walker. We have mentioned before that the moment when the trailing leg leaves the ground is considered as the beginning of the motion, and the

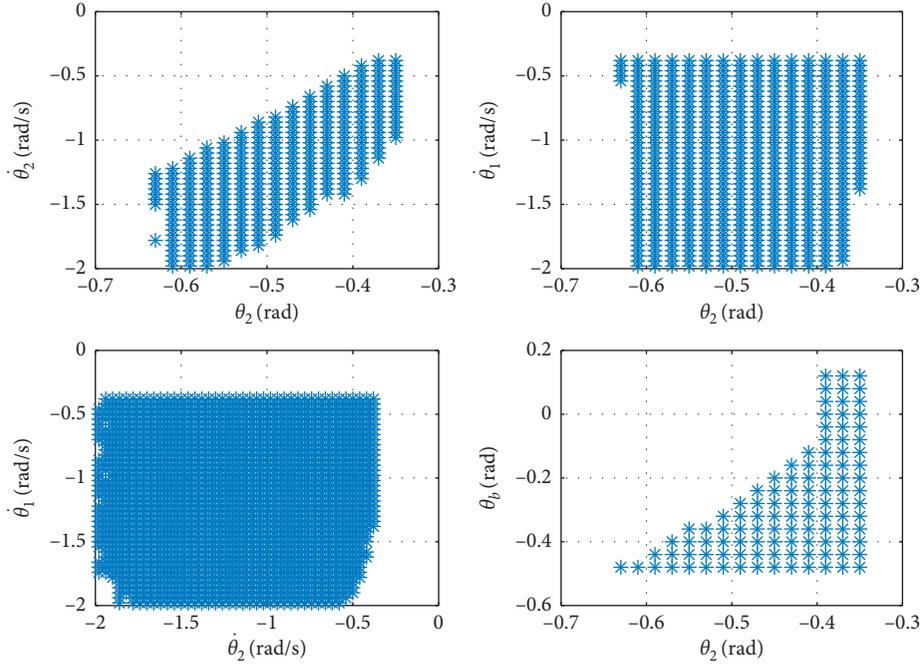


FIGURE 10: Basin of attraction of the CPG-based biped for $u_e = 2.4$ and $u_s = 22$.

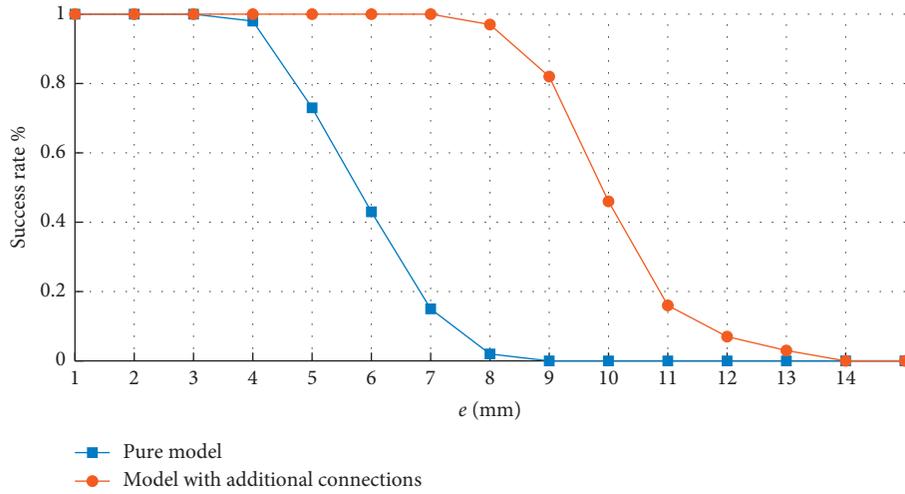


FIGURE 11: Successful trajectory passing with random height variations with and without the additional connections. Adding the connections increased the robustness about 60%.

robot’s initial condition is determined by the angle and the angular velocity of the trailing leg ($\theta_{t2}, \dot{\theta}_{t2}$), the angle and the angular velocity of the trailing foot ($\theta_{f2}, \dot{\theta}_{f2}$), the angular velocity of the leading leg $\dot{\theta}_{t1}$, and the angle and angular velocity of the body ($\theta_b, \dot{\theta}_b$). We restricted our study to the situations where the angular velocity of the body is zero and the trailing foot angle and angular velocity equal to those of the trailing leg. Thus, the BoA is determined with the remaining DoF using the cell mapping method [15, 16]. Figure 10 shows the projection of the BoA in the space defined by the variables $(\theta_{t2}, \dot{\theta}_{t2}, \dot{\theta}_{t1}, \theta_b)$ for $u_e = 2.4$ and

TABLE 3: The result of PSO algorithm of the optimized values of the disturbance rejection connections.

Parameter	Value
c_{ee}^1	20.46
c_{ee}^2	-2.13
c_{ee}^4	23.52
c_{ee}^5	12.48
c_{se}^1	-476
c_{se}^3	9.1×10^3
c_{se}^4	-3.4×10^3
c_{se}^5	5.6×10^3

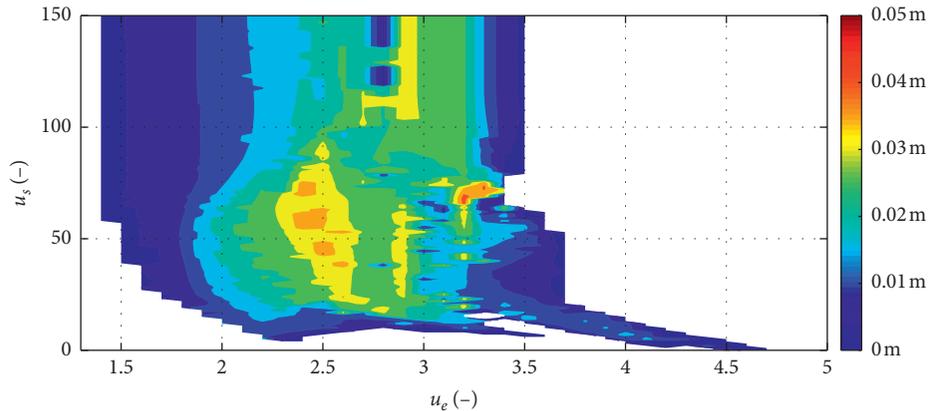


FIGURE 12: The maximal allowable ground disturbance as a function of the input signals u_e and u_s . The maximal allowable disturbance is 0.048 m at $u_e = 3.1$ and $u_s = 67$.

$u_s = 22$. We can see that the stable regions of the walker are relatively large which proves the effectiveness of the proposed control model.

4.5. Disturbance Rejection. We studied the ability of the proposed model to walk on uneven terrain. A random trajectory with 50 step length has been generated, and we added a random change in the ground height in the range $[-e, e]$ for each step. The process was repeated 100 times, and the percentage of the successful trajectory passing was calculated. Figure 11 shows the results as a function of the increasing amplitude of the maximum change e .

To increase the robustness, we added new connections to the neural model. These connections are dependent on the disturbance and work only at some phases on some neurons as follows: (1) Neurons related to the stance foot during the swing phase. (2) Neurons related to the trailing foot during the push-up phase. (3) Neurons related to the hip joints during the swing phase. These connections are inspired by the human behavior to overcome a sudden ground height variation during walking. We also considered the study on a limit cycle walking, which indicates the importance of controlling the swing leg and the stance foot to increase the stability against disturbances [16]. The previous connections work directly after the robot senses a change in the ground height. This disturbance e is calculated depending on the difference between the two feet's height, and it can be added to our model as follows:

$$\begin{aligned} u_{ee}^i &= u_e^i + c_{ee}^i e; \\ u_{ss}^i &= u_s^i + c_{se}^i e, \end{aligned} \quad (18)$$

where i is the joint index, c_{se}^i and c_{ee}^i are the connections weight, and u_e^i and u_s^i are the nominal control signals and e in meters. To calculate the appropriate values of the connections weight, we employed the PSO algorithm again with a cost function defined by the average number of success steps on a distributed ground with random height variation in the range $[-10, 10]$ mm. The same PSO parameters as in Table 2 are used with a swarm size of 150 particles. The obtained

optimal weights are shown in Table 3. Figure 11 shows the increase in the robustness after adding the new connections with the optimized values. The walker can pass a trajectory with a random variation with amplitude up to 2.5% of its leg length with about 50% success rate.

We studied the maximum single step that could be passed by our robot with the added connections for all the control signals (u_e, u_s) . Figure 12 shows the results. The largest ground height variations the walker can overcome is 0.048 m at $u_e = 3.1$ and $u_s = 67$.

5. Discussion

The proposed phase-independent CPG-based control model exhibits good results depending on several performance criteria. The Froude number of the robot ranges from 0.12 to 0.43. This range is close to the speed achieved in [12] which ranged from 0.15 to 0.43 and comparable with previous passivity-based bipedal walkers (e.g., the Fr of "Veronica" ranges from 0.07 to 0.16 [17] and the Fr of "Runbot" ranges from 0.25 to 0.5 [25]).

Besides the wide speed range reachable by the two input signals, the model has a great ability to change its speed by only changing the control signals at the beginning of the push-off phase. However, the transition between working points is not possible for every two points. We observed that some points are not reachable from other points, and the walker have to pass through at least one intermediate point to complete the transition correctly. This is because those former points are not in the basin of attraction of the later points.

Finally, adding new connections to our neural model increased the robot stability and enabled our biped to walk on uneven terrain with a maximum height change up to 20 mm with about 50% of success rate. These new connections could be considered as a new layer added to the neural model, and they only work when the robot discovers a change in the height of the ground. We can generalize this idea by changing the weight of those connections or by using different connections for different environments or situations, while the basic structure of our model is kept fixed.

Our robot can handle a change up to 4.8 cm, which is about 6% of its leg length, whereas the Huang model was able to handle a ground disturbance of maximum 5.25% of leg length [12]. The passive-based walking robot “Meta” deals with ground height changes of 5% of leg length with a control strategy combining the effect of ankle stiffness, local stance ankle control, and ankle push-off control [16]. Existing studies reported that a simulated biped model with reinforcement learning could overcome ground disturbances of height of only 1.67% of leg length [26].

To conclude, our model is simpler than the model proposed by Huang et al. [12] which also controls the external torque and internal dynamics of the walker. Despite the simplicity, our model achieved a close performance to the performance of Huang’s model, and it showed a good local and global stability on all the possible values of the two input signals. Besides, our model is able to change the walking velocity between most of the allowable speed values in three to four steps.

6. Conclusion

In this paper, a walking robot model was presented. The model, based on the central pattern generators, generates periodic signals that control the torque and stiffness of the compliance joints. This neural model is controlled using only two control signals: one for controlling joints torque and the other for controlling joints stiffness. We used the proposed model to control a 2D 7-link biped robot, and the procedure proved good performance in terms of versatility and walking pattern transitions. The robot walks on the 1-period limit cycle and has a good local stability around the fixed point in most working space defined by the values of the control signals which generate stable gaits.

Finally, we proposed a method which helped to increase the robot stability against disturbance in the height of the ground by adding connections to the control model related to the disturbance, and through experiments on the simulated model, we observed a good improvement in the disturbance rejection.

As a future work, we suggest to explore the properties of Hindmarsh–Rose (HR) neural model [27] on the aspect of CPG and reconstruct our CPG network based on the HR model.

Data Availability

The software code used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] 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.
- [2] 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.
- [3] F. Dzeladini, N. Ait-Bouziad, and A. Ijspeert, “CPG-based control of humanoid robot locomotion,” *Humanoid Robotics: A Reference*, pp. 1–35, 2018.
- [4] P. Arena, L. Fortuna, and M. Frasca, “Multi-template approach to realize central pattern generators for artificial locomotion control,” *International Journal of Circuit Theory and Applications*, vol. 30, no. 4, pp. 441–458, 2002.
- [5] A. J. Ijspeert, A. Crespi, D. Ryczko, and J. M. Cabelguen, “From swimming to walking with a salamander robot driven by a spinal cord model,” *Science*, vol. 315, no. 5817, pp. 1416–1420, 2007.
- [6] Y. Fukuoka, Y. Habu, and T. Fukui, “Analysis of the gait generation principle by a simulated quadruped model with a CPG incorporating vestibular modulation,” *Biological Cybernetics*, vol. 107, no. 6, pp. 695–710, 2013.
- [7] Q. Chang and F. Mei, “A bioinspired gait transition model for a hexapod robot,” *Journal of Robotics*, vol. 2018, Article ID 2913636, 11 pages, 2018.
- [8] T. McGeer, “Passive dynamic walking,” *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.
- [9] Z. Wang, *Contact Impact Inhibition Strategy for Biped Robot Walking Based on Central Pattern Generator*, IEEE, Piscataway, NJ, USA, 2015.
- [10] A. G. Baydin, “Evolution of central pattern generators for the control of a five-link bipedal walking mechanism,” *Paladyn, Journal of Behavioral Robotics*, vol. 3, no. 1, pp. 45–53, 2012.
- [11] G. Endo, J. Morimoto, J. Nakanishi, and G. Cheng, “An empirical exploration of a neural oscillator for biped locomotion control,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, New Orleans, LA, USA, April 2004.
- [12] Y. Huang, B. Vanderborght, R. Van Ham, and Q. Wang, “Torque–stiffness-controlled dynamic walking with central pattern generators,” *Biological Cybernetics*, vol. 108, no. 6, pp. 803–823, 2014.
- [13] R. C. de Paiva, A. R. S. Romariz, and G. A. Borges, “Searching CPG parameters for humanoid gait using particle swarm optimization and genetic algorithm,” in *Proceedings of the 2013 16th International Conference on Advanced Robotics (ICAR)*, Montevideo, Uruguay, November 2013.
- [14] T. Mori, Y. Nakamura, M.-A. Sato, and S. Ishii, “Reinforcement learning for CPG-driven biped robot,” in *Proceedings of the 19th National Conference on Artificial Intelligence on AAAI*, vol. 4, San Jose, CA, USA, July 2004.
- [15] Y. Huang, Q. Wang, B. Chen, G. Xie, and L. Wang, “Modeling and gait selection of passivity-based seven-link bipeds with dynamic series of walking phases,” *Robotica*, vol. 30, no. 1, pp. 39–51, 2012.
- [16] D. G. E. Hobbelen and M. Wisse, “Limit cycle walking,” in *Humanoid Robots, Human-Like Machines*, InTech, London, UK, 2007.
- [17] Y. Huang, B. Vanderborght, R. Van Ham et al., “Step length and velocity control of a dynamic bipedal walking robot with adaptable compliant joints,” *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 598–611, 2013.
- [18] B. Vanderborght, B. Verrelst, R. Van Ham et al., “Exploiting natural dynamics to reduce energy consumption by controlling the compliance of soft actuators,” *The International Journal of Robotics Research*, vol. 25, no. 4, pp. 343–358, 2006.

- [19] J. Yamaguchi, D. Nishino, and A. Takanishi, "Realization of dynamic biped walking varying joint stiffness using antagonistic driven joints," in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, vol. 98, Leuven, Belgium, May 1998.
- [20] K. Matsuoka, "Sustained oscillations generated by mutually inhibiting neurons with adaptation," *Biological Cybernetics*, vol. 52, no. 6, pp. 367–376, 1985.
- [21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE Service Center, Perth, Australia, 1995.
- [22] M. E. H. Pedersen, "Good parameters for particle swarm optimization," Tech Rep. vol. 1001, HVASS, Copenhagen, Denmark, 2010.
- [23] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization: past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.
- [24] Y. Huang, Q. Wang, G. Xie, and L. Wang, "Optimal mass distribution for a passive dynamic biped with upper body considering speed, efficiency and stability," in *Proceedings of the Humanoids 2008—8th IEEE-RAS International Conference on Humanoid Robots*, IEEE Daejeon, South Korea, December 2008.
- [25] T. Geng, B. Porr, and F. Worgotter, "Fast biped walking with a sensor-driven neuronal controller and real-time online learning," *The International Journal of Robotics Research*, vol. 25, no. 3, pp. 243–259, 2006.
- [26] E. Schuitema, *Using a Controller Based on Reinforcement Learning for a Passive Dynamic Walking Robot*, IEEE, Piscataway, NJ, USA, 2005.
- [27] L. M. Rosa, M. I. Rabinovich, R. Huerta, H. D. I. Abarbanel, and L. Fortuna, "Slow regularization through chaotic oscillation transfer in an unidirectional chain of Hindmarsh-Rose models," *Physics Letters A*, vol. 266, no. 1, pp. 88–93, 2000.