Heliyon 10 (2024) e31847

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Workspace trajectory generation with smooth gait transition using CPG-based locomotion control for hexapod robot

Kifah Helal *, Ahed Albadin, Chadi Albitar, Michel Alsaba

Higher Institute for Applied Sciences and Technology, Damascus, Syria

ARTICLE INFO

Keywords:
CPG (Central Pattern Generators)
Mobile robots
Hexapod robot
Gait transition
Foot trajectory
Online adjustability

ABSTRACT

—This paper presents a new control methodology for achieving smooth gait transitions for a hexapod robot using Central Pattern Generators (CPGs). The approach involves modifying the Phase Oscillator within the CPG network to enable smooth transitions between different gaits in order to improve the adaptability to changing environmental conditions. The foot trajectory generator is designed based on the CPG output, allowing the possibility of online adjustment of foot trajectory parameters, such as step height and size, as well as the robot's speed and direction. Our simulation demonstrates the effectiveness of the modified oscillator in achieving smoother gait transitions with a transition time falls close to the output period of the CPG oscillators, and experiments on a real hexapod robot validate the feasibility and efficiency of our approach in considering online adjustability of trajectory parameters, confirming the potential of this methodology to enhance the locomotion capabilities of legged robots for navigating complex terrains.

1. Introduction

Legged robots have emerged as versatile and adaptable machines capable of traversing various terrains with superior mobility. Their unique locomotion capabilities make them ideal for tasks requiring navigation through complex and unpredictable environments. Compared to wheeled or crawler robots, legged robots exhibit remarkable agility besides the ability to overcome obstacles, enabling them to operate effectively in challenging terrains [1,2].

Smooth gait transitions play a key role in the locomotion of legged robots, allowing them to seamlessly adapt to different movement patterns and environmental conditions. This adaptability is especially related to unstructured terrains of complex environments with varying degrees of stiffness, where legged robots are designed to operate in. [3]. Achieving smooth gait transitions is a major challenge due to the complex coordination required between multiple legs during locomotion [4]. The complexity arises from the need to synchronize leg movements while maintaining stability and efficiency during gait transitions [5]. However, maintaining the stability and synchronization of robots during locomotion or path following under external disturbances is an extended challenging problem in the field of robotics [6,7].

In recent years, extensive research has focused on addressing the challenge of smooth gait transitions in legged robots using Central Pattern Generators (CPGs). CPGs are network structures that generate rhythmic patterns, emulating the neural mechanisms observed in animals during locomotion [8]. The Matsuoka Oscillator [9], a prominent example of a CPG, is widely used in robotics. It has been

* Corresponding author.
E-mail addresses: kifah.hlal@hiast.edu.sy (K. Helal), ahed.albadin@hiast.edu.sy (A. Albadin), shadi.albitar@hiast.edu.sy (C. Albitar), michel.alsaba@hiast.edu.sy (M. Alsaba).

https://doi.org/10.1016/j.heliyon.2024.e31847
Received 16 February 2024; Received in revised form 2 May 2024; Accepted 22 May 2024
Available online 25 May 2024

extensively studied in various research works, demonstrating its effectiveness in generating coordinated and adaptive locomotion [10–12]. Another influential oscillator in CPG research is the Hopf Oscillator, known for its ability to generate sustained oscillations. It has been investigated in the context of biological motor control [13] and fish robots and legged hexapod robots with sensory feedback [14,15]. Phase oscillators, which synchronize their phase with other oscillators, have been extensively studied for coordination and synchronization of rhythmic movements in complex systems [16,17]. Furthermore, Neural CPG Models have proven their importance, especially for their ability to incorporate complex neural dynamics and sensory feedback [18]. These models have been applied in enhancing neural CPGs with sensory feedback for malfunction compensation [19] and using a multi-layered neural CPG for humanoid robots [20]. Several studies have successfully utilized CPG-based control approaches to achieve coordinated leg movements and smooth gait transitions in various legged robots. Haitao Yu et al. [21] accomplished a smooth gait transition of a Hexapod robot by deriving an analytical limit cycle approximation of the Van der Pol oscillator, but the transition time was relatively long. Other research, Linlin Shang et al. [22] considered the time of conversion of CPG in almost 1 s using Hopf-based oscillator CPG. Still, there was no physical verification on a real robot and the CPG network output was unsymmetrical which prevented the network from considering online adjustment of foot trajectory parameters. Sai Gu et al. [23] proposed a low-level impedance controller to integrate the work of Hopf oscillator as a high-level controller for solving the problem of sharp inflection points between steps, the results worked well in simulation. However, the research lacks foot trajectory equations and has no consideration for adjusting the parameters of foot trajectory online. Wei Zhang et al. [24] proposed a solution to the problem of mutual repulsion caused by the trajectory of the foot end swinging around the respective heel joint as an arc during the forward movement of the hexapod robot. The research verified the feasibility of the proposed method in simulation and on a real robot. Nevertheless, the research has not considered gait transition. Most of the previous research did not consider the most important aspects of legged robots' locomotion: **1)** Online adjustability of gait parameters (direction, step height and size, and speed). **2)** Smooth gait transition between different gait patterns. **3)** Transition time between gaits. While some of them showed results on simple transitions only. Therefore, we considered combining most strength points in other architectures (speed and stable smooth gait transition, online adjustable parameters, and physical verification for various gaits patterns) in our control architecture.

In this paper, we present an improved CPG-based locomotion control method for a hexapod robot to achieve gait generation and online adjustability of its feet trajectory parameters. The CPG network is modeled as a group of coupled modified oscillators. Various gait patterns and online transition between them can be achieved by manipulating the phase differences between these oscillators within the network. Furthermore, the CPG parameters are responsible for controlling the foot trajectory, providing stability for robot during a smooth transition, and enabling independent online adjustments (independence between foot trajectory parameters). Notably, the control method provides the capability to dynamically modify the step size and height for each leg, speed, and direction of the robot during operation. To validate the effectiveness of the proposed control method, extensive simulation has been performed using a hexapod robot model (Fig. 1), combined with experimental demonstrations on a physical robot (Fig. 2), evaluating its performance in achieving smooth gait transitions and adjustability of gait parameters.

The main contributions of this paper are as follows.

1) The design of a CPG network architecture to generate various gaits for a hexapod robot and to guarantee a smooth transition between them. The proposed CPG network considers fast transition between gaits and enables walking in all directions with robot speed adjustability.
2) The design of a workspace trajectory generator to map the output signals of the CPG network to 3-D workspace trajectories online for a legged robot. The proposed mapping method enables online adjustability of the foot trajectory parameters (step size, step height, direction).
3) The validation of the proposed control architecture in simulation and on a real Hexapod robot.

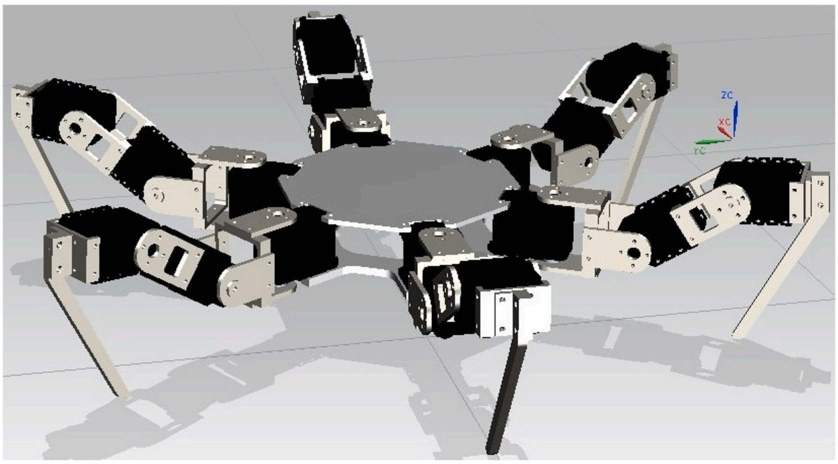

Fig. 1. Simulated hexapod robot.

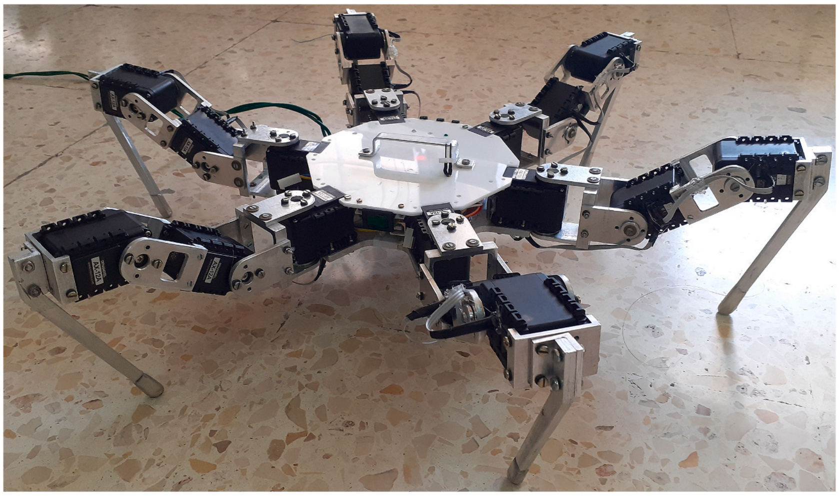

Fig. 2. Real hexapod robot.

The structure of this paper is organized as follows. Section.2 introduces the Hexapod Robot modeling and its inverse kinematics model. Section.3 shows the proposed control architecture and the workspace trajectory generator. Simulation and experimental in Section.4 verify the feasibility and efficiency of our approach. In Section.5 this paper is concluded.

2. Hexapod robot modeling

The considered hexapod robot platform in this work consists of a base with six legs, all featuring an identical structure. Each leg consists of three joints. The hip joint (Joint- 1) controls the forward and backward movements of the leg, the knee joint (Joint-2) controls the rise and fall of the leg, and the ankle joint (Joint-3) controls the leg's extension and flexion movements, as shown in Fig. 3.

As shown in Fig. 3, R_B is the centroid coordinate system of the robot, R_i is the foot coordinate system (of the i th leg) whose axes are parallel to the axis of R_B and centered at the initial position of leg i (fixed relative to R_B). O_j is the coordinate system of the j th joint. Each leg has a D-H parameter as shown in Table 1.

Suppose that the foot position coordinates $p_i^{R_i}$ w.r.t R_i are $\left(x_i^{R_i}, y_i^{R_i}, z_i^{R_i}\right)$, where i is the leg index. The foot position coordinates $p_i^{R_B}\left(x_i^{R_B}, y_i^{R_B}, z_i^{R_B}\right)$ w.r.t R_B can be calculated using $p_{R_i}^{R_B}\left(x_{R_i}^{R_B}, y_{R_i}^{R_B}, z_{R_i}^{R_B}\right)$ (R_i center coordinates w.r.t R_B) as follows:

$$p_i^{R_B} = \begin{bmatrix} x_i^{R_B} \\ y_i^{R_B} \\ z_i^{R_B} \end{bmatrix} = \begin{bmatrix} x_i^{R_i} + x_{R_i}^{R_B} \\ y_i^{R_i} + y_{R_i}^{R_B} \\ z_i^{R_i} + z_{R_i}^{R_B} \end{bmatrix} \tag{1}$$

Equation (1) allows us to generate the foot trajectory locally w.r.t the frame R_i and then transfer the foot position to the global coordinates (coordinates w.r.t R_B) to be used in the inverse kinematic model to calculate the joints' angles.

To determine the relationship between the coordinates of each foot of the hexapod robot in the robot centroid coordinate system (Base frame), the forward and inverse kinematic models of each hexapod leg must be determined.

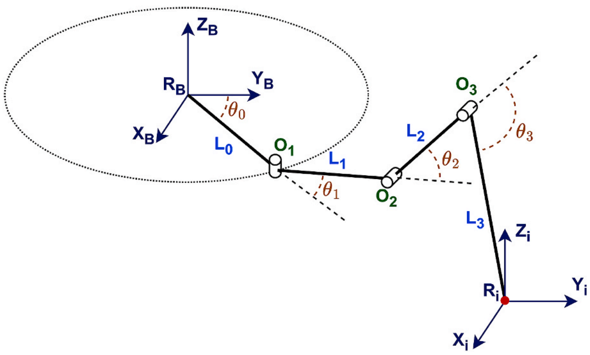

Fig. 3. Coordinate systems defined for joint angles of a Hexapod's leg.

Table 1
Single leg D-H parameters of Hexapod Robot.

Joint	α_j	d_j	θ_j	r_j
1	0	0	θ_1	0
2	0	L_1	θ_2	0
3	$\pi/2$	L_2	θ_3	0
4	0	L_3	0	0

To calculate the inverse kinematic model of the leg, we start with the matrix ${}^{j-1}T_j$ which is called the homogeneous transformation matrix from the coordinate system O_j to the coordinate system O_{j-1} (Eq. (2)).

$$ {}^{j-1}T_j = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Where $C\theta_j = cos\ \theta_j$ and $S\theta_j = sin\ \theta_j$. Using the D-H parameter and the homogeneous transformation matrix, the position of the foot in the centroid base coordinate system of the robot $p_i^{R_B}$ can be obtained from the last column of ${}^B T_4$ as follows:

$$ {}^B T_4 = {}^B T_0 \cdot {}^0 T_1 \cdot {}^1 T_2 \cdot {}^2 T_3 \cdot {}^3 T_4 \tag{3}$$

$$ {}^B T_4 = \begin{bmatrix} C_{01}C_{23} & -C_{01}S_{23} & 0 & L_0C_0 + C_{01}(L_1 + L_2C_2 + L_3C_{23}) \\ S_{01}C_{23} & S_{01}S_{23} & 0 & L_0S_0 + S_{01}(L_1 + L_2C_2 + L_3C_{23}) \\ S_{23} & C_{23} & 1 & L_2S_2 + L_3S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$ p_i^{R_B} = \begin{bmatrix} x_i^{R_B} \\ y_i^{R_B} \\ z_i^{R_B} \end{bmatrix} = \begin{bmatrix} L_0C_0 + C_{01}(L_1 + L_2C_2 + L_3C_{23}) \\ L_0S_0 + S_{01}(L_1 + L_2C_2 + L_3C_{23}) \\ L_2S_2 + L_3S_{23} \end{bmatrix} \tag{5}$$

where $C_j = cos\ \theta_j, C_{ij} = cos(\theta_i + \theta_j)$ and similarly for S_j and S_{ij} respectively.

As for the inverse kinematic solution for each leg of the hexapod robot, we have to determine the three joint angles of the leg $[\theta_1, \theta_2, \theta_3]^T$, given the coordinates of the foot of a leg $\left[x_i^{R_B}, y_i^{R_B}, z_i^{R_B}\right]^T$ in the robot centroid coordinate system using the following equations:

$$ \theta_1 = \tan^{-1}\left(\frac{y - L_0S_0}{x - L_0C_0}\right) - \theta_0 \tag{6}$$

$$ d = \sqrt{(x - L_0C_0)^2 + (y - L_0S_0)^2} \tag{7}$$

$$ \theta_3 = -\cos^{-1}\left(\frac{(d - L_1)^2 + z^2 - L_2^2 - L_3^2}{2L_2L_3}\right) \tag{8}$$

$$ S_2 = \frac{(L_2 + L_3C_3)z - (L_3S_3)\left(\frac{x - L_0C_0}{C_{01}} - L_1\right)}{(L_3S_3)^2 + (L_2 + L_3C_3)^2} \tag{9}$$

$$ C_2 = \frac{(L_3S_3)z + (L_2 + L_3C_3)\left(\frac{x - L_0C_0}{C_{01}} - L_1\right)}{(L_3S_3)^2 + (L_2 + L_3C_3)^2} \tag{10}$$

$$ \theta_2 = atan\ 2(S_2, C_2) \tag{11}$$

The angles will be calculated according to the adopted sampling time which means that to guarantee the smoothness of foot motion, the angular velocity for each joint should be calculated as follows:

$$ \dot{\theta}_1 = \frac{\dot{y} + S_{01}(\dot{\theta}_2L_2S_2 + [\dot{z} - \dot{\theta}_2L_2C_2]\tan_{23})}{C_{01}(L_1 + L_2C_2 + L_3C_{23})} \tag{12}$$

$$\dot{\theta}_2 = \frac{C_{01}(\dot{x} + \dot{y}\tan_{01}) + \dot{z}\tan_{23}}{L_2(C_2\tan_{23} - S_2)} \tag{13}$$

$$\dot{\theta}_3 = \frac{\dot{z} - \dot{\theta}_2 L_2 C_2}{L_3 C_{23}} - \dot{\theta}_2 \tag{14}$$

3. The proposed control architecture

Central Pattern Generators (CPGs) are biological neural networks that produce rhythmic output patterns without the need for sensory feedback. This means that either individual neurons or the way they are connected to each other can lead to oscillatory dynamics that are useful for various biological functions, such as walking gaits, respiration, or even circadian rhythms [25,26]. CPGs are important for hexapod robots, which emulate insect locomotion, offering a robust and adaptable control system. These bioinspired mechanisms can be refined and enhanced to allow robots to navigate dynamic environments and compensate for any potential failures or injuries that the robot may encounter. Unlike traditional control systems that struggle with variability, CPGs provide the flexibility necessary for real-world tasks, highlighting the constructive collaboration between biology and robotics for optimal locomotor skills [27–29].

The control architecture comprises a network of Central Pattern Generators (CPGs) alongside a workspace trajectory generator (Fig. 4). The CPG network consists of six individual CPG oscillators; each assigned to control one of the six legs. These oscillators interact with each other to collectively generate coordinated rhythmic patterns. The network's output appears as six sine waves, where the amplitude and the offset of each wave control the step size and height for each leg individually. In addition, the phase difference between the six sine waves determines the gait type, while the frequency controls the speed of the robot. The workspace trajectory generator employs the network output to generate the foot trajectory. In this process, the desired direction of movement is directly

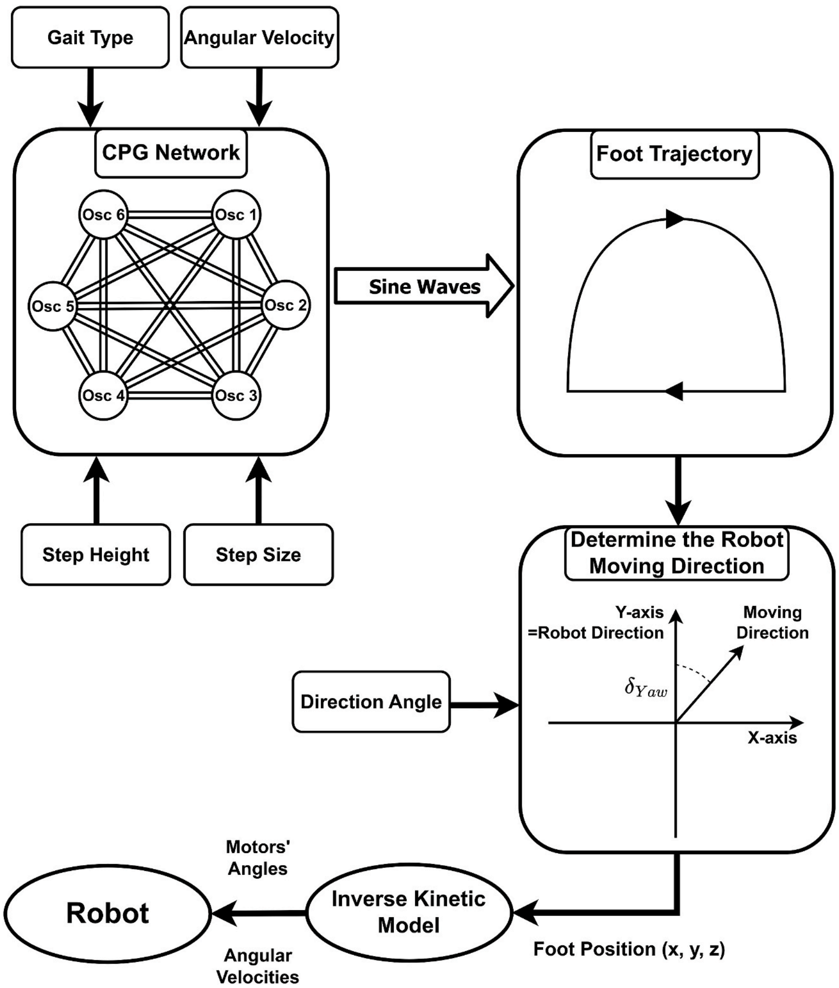

Fig. 4. The proposed Control Architecture.

incorporated as a parameter influencing the trajectory.

Kuramoto [30] considered the phase oscillator model as a CPG unit, and it was transformed by Wang et al. [31] into a CPG network. This network consists of N oscillators (N is the number of legs), and each oscillator has its own frequency and amplitude. However, the phase difference is controlled by the desired gait. The output signals of the network are described as follows:

$$\ddot{a}_i = \mu_{a_i}^2 (A_i - a_i) - 1.5\mu_{a_i}\dot{a}_i \tag{15}$$

$$r_i = a_i \, sin(\theta_i) + c_i \tag{16}$$

Where r_i is the output signal of the i th oscillator and it has a sinusoidal shape with an amplitude a_i, a phase θ_i, and a constant off-set c_i. Equation (15) ensures that the amplitude of the output signals a_i will asymptotically and monotonically converge to A_i which is the desired amplitude. Therefore, the amplitude of oscillations can be smoothly modulated. μ_{a_i} is a positive constant determining the convergence speed of state a_i to A_i.

We proposed to modify the previous adopted network in order to consider smooth gait transition as follows:

$$D_i = \alpha \sum_{j=1}^{i} \frac{1}{v_i} sin\left(\frac{(\theta_j - \theta_i - \Delta\varphi_{i,j})_{\in[-\pi,+\pi]}}{2}\right) \tag{17}$$

$$\dot{\theta}_i = 2v_i(\pi + arctan(D_i)) \tag{18}$$

$$\ddot{c}_i = \mu_{c_i}^2 (C_i - c_i) - 1.5\mu_{c_i}\dot{c}_i \tag{19}$$

Each oscillator in equations 15–18 runs independently at its own frequency v_i, while the synchronization term D_i tends to synchronize the frequency of oscillator i with the other oscillators. α is a constant used to control the value of D_i. $\Delta\varphi_{i,j}$ denotes the desired phase shift between oscillator i and j. c_i is chosen to satisfy equation (19) in order to make it adjustable smoothly, so we can use it in the design of the trajectory modulator. Adjusting c_i and a_i allows to make the step size and step high independent as it will be shown later.

D_i is responsible for realizing the synchronization between the oscillators. Therefore, in equation (17) we considered the following aspects: **First,** the expression $\frac{(\theta_j - \theta_i - \Delta\varphi_{i,j})_{\in[-\pi,+\pi]}}{2}$ is used in this form to ensures that the j th oscillator effectively contributes to the synchronization of the i th oscillator; In other word, for achieving the desired phase difference $\Delta\varphi_{i,j}$ between θ_j and θ_i, the j th oscillator induces excitations that guide the phase of the i th oscillator with the minimum possible change in θ_i. Thus, we ensured updating the phases in the minimal time required to achieve the desired phase difference. **Second,** synchronizing the i th oscillator with just the preceding oscillators in the network and not all the oscillators (asymmetrical synchronization), in this way we broke the symmetry between the contributions of all oscillators into one, thus, enhancing stability and accelerating the transition when modifying the required phase differences $\Delta\varphi_{i,j}$. This modification has a crucial impact on the time of transition while maintaining the sinusoidal shape of the CPG output. **Third,** the term $1/v_i$ will decrease the value of D_i when v_i increases and vice versa. This technique is used as a tradeoff to consider the value of the term $2v_i \arctan(D_i)$ in equation (18) which becomes relatively large at high frequency v_i.

Equation (18) determines the new state of θ_i based on two terms; $2v_i\pi$ (the oscillator angular velocity) which is only related to the oscillator frequency v_i and represents the steady state value of $\dot{\theta}_i$ when there is no change in the gait type, and $2v_i \, arctan(D_i)$ which is related to the oscillator frequency and the synchronization term D_i and represents the added value to the oscillator angular velocity in the transition phase between two gaits. Our purpose of the structure of the second term (v_i multiplied with $arctan(D_i)$) is to superimpose the error D_i to the derivative of the phase $\dot{\theta}_i$ in a form of an angle $arctan(D_i)$ multiplied by an angular velocity v_i to make the sum of $2v_i\pi$ and $2v_i \, arctan(D_i)$ meaningful and make it possible to compare between the two previous terms in order consider the right conditions on the value of $\dot{\theta}_i$ correctly.

We did not use the term D_i in equation (18) directly but through the arctan function to maintain discriminability based on the value of D_i, while also imposing a saturation limit to ensure that $\dot{\theta}_i$ remains within the range $[v_i\pi, 3v_i\pi]$. This restriction is important to prevent $\dot{\theta}_i$ from becoming too large or negative. The underlying concept of using *arctan* function is that if D_i is large enough, so $arctan(D_i) \approx \pi/2$, and D_j showed the same behavior, then there is no need for the i th oscillator to adjust itself differently from the j th oscillator, because they are both so far from the desired state and they just need to move towards the desired state. However, when D_i is relatively small, the precise value of D_i becomes crucial in determining the appropriate adjustment in angular velocity $\dot{\theta}_i$ because in this case the actual value of θ_i is close to the desired value of θ_i so it must be exactly synchronized with the network.

Our proposed CPG architecture is able to realize gait transition between any different gaits while ensuring convergence to the desired gait due to the asymmetry in synchronization structure. Moreover, using $arctan(D_i)$ (instead of D_i directly) which, due to the limitations of the *arctan* value lying in the range $[-\pi/2, +\pi/2]$, allows to achieve fast transition by benefiting from the large values of D_i without losing smoothness, as well as without losing complete synchronization of the network by maintaining the discrimination at small phase differences. The smooth transition is ensured because the condition $\dot{\theta}_i \in [v_i\pi, 3v_i\pi]$ is always satisfied which makes the output signals maintaining their sinusoidal shape even during transition.

The position of each foot on the 3D workspace trajectory is given as functions of the state of the corresponding oscillator r_i. We adopted this technique in our design of the mapping functions to give the foot trajectory a shape of a semi-ellipse with adjustability and independence between height and size (Fig. 5). Nevertheless, the direction of the robot is superimposed to the generated trajectory to produce the desired walk in the desired direction.

We proposed to convert the output of the CPG to a trajectory in the coordinate R_i by individually generating the height z_i^j and the horizontal displacement u_i^j of the foot depending on the instantaneous value of r_i as follows:

$$A_{i,j} = \frac{\alpha_{i,j}}{\beta_{i,j}} = \frac{r_i - c_i - a_i \sin(\psi_i)}{a_i(1 - \sin(\psi_i))} \tag{20}$$

$$z_i^j = \begin{cases} A_{i,j}^2 \cdot h_i \; r_i > c_i + a_i \sin(\psi_j) \\ 0 \quad otherwise \end{cases} \tag{21}$$

$${}_1B_{i,j} = \frac{\arcsin\left(\frac{r_i - c_i}{a_i}\right) - \psi_j}{\frac{\pi}{2} - \psi_j}, \; {}_2B_{i,j} = \frac{\psi_j - \arcsin\left(\frac{r_i - c_i}{a_i}\right)}{\frac{\pi}{2} + \psi_j} \tag{22}$$

$$u_i^j = \begin{cases} \mathrm{sign}(\Delta r_i)\frac{a_i}{2}\left(-1 + {}_1B_{i,j}{}^3\right) \; r_i > c_i + a_i \sin\psi_j \\ \mathrm{sign}(\Delta r_i)\frac{a_i}{2}\left(-1 + {}_2B_{i,j}\right) \; r_i < c_i + a_i \sin\psi_j \end{cases} \tag{23}$$

z_i^j in equation (21) is the height of the i th foot when the robot is moving in the j th gait pattern; each leg has its own step height $h_i = \frac{a_i + c_i}{2a_i}H$ which depends on the parameters of the oscillator and a fixed parameter H (The maximum height that can be achieved for a leg). The first row in equation (21) represents the swing phase while the second row represents the stance phase. ψ_j is the angle in range $[0, \pi/2]$ at which consecutive oscillators output intersect, and it is the dividing point between the swing phase and stance phase. Here we considered $\psi = [\pi/3, \pi/4, \pi/6, 0]$ for the gaits: Wave, Tetrapod, Rice, and Tripod, respectively. Fig. 6 shows the shape of $z_{1,2,3}^{Rice}$ at Rice gait (feet (4, 5, 6) are the same as feet (1, 2, 3) respectively). The term $A_{i,j}$ has a value within the range [0, 1] depending on the value of the output of the CPG network r_i, and the value of a_i and c_i. When there is no adjustment to the foot trajectory parameters, the value of $A_{i,j}$ is controlled by r_i and this is why we considered maintaining the continuity and sinusoidal shape of r_i during gait transition, to guarantee the validity of the foot trajectory. On the other hand, when there is an online adjustment to the foot trajectory parameters, the value of A_i and C_i in equations (15) and (19) respectively will be modified in order to achieve the desired new maximum height $h_{i,max} = \frac{A_i + C_i}{2A_i}H$ and then each of a_i and c_i will start to converge smoothly to A_i and C_i respectively according to equations (15) and (19).

u_i^j in equation (23) is the horizontal displacement of the i th foot when the robot is moving in the j th gait pattern; each leg has its own step size a_i which is the amplitude of its oscillator. The first row in equation (23) corresponds to the swing phase (when the foot is not in touch with the ground and moves from $-a_i/2$ to $a_i/2$ w.r.t R_i frame). The second row corresponds to the stance phase (when the foot is in touch with the ground and moves from $a_i/2$ to $-a_i/2$ w.r.t R_i frame). In stance phase, the horizontal velocity of all feet is linear w.r.t the time to avoid the feet sliding. The term Δr_i is used to determine if the instantaneous value of r_i is increasing or decreasing and it has the value $\Delta r_i = r_i(t) - r_i(t - \Delta t)$. Fig. 6 shows the shape of $u_{1,2,3}^{Rice}$ at Rice gait. The terms ${}_{1,2}B_{i,j}$ take their values in the range $[0, 1]$ depending on the value of the output of the CPG network r_i and the value of a_i and c_i while the maximum value of u_i^j is just related to A_i. As in z_i^j When there is no adjustment to the foot trajectory parameters, the values of ${}_{1,2}B_{i,j}$ are controlled by r_i. On the other hand, when there is an online adjustment to the foot trajectory parameters, the value of A_i in equation (15) will be modified in order to achieve the desired new maximum step size $A_i/2$ and then a_i will start to converge smoothly to A_i according to equation (15). Due to the dependency of z_i^j parameters on the value of a_i, the value of C_i in equation (19) will also be modified accordingly to maintain the desired step height. Thus, the independence between step height and size is accomplished.

The Robot can move in any direction, which means that there will be an angle between the direction of the robot and the direction of the movement for each leg i called $\delta_{yaw,i}$ which is expressed as follows:

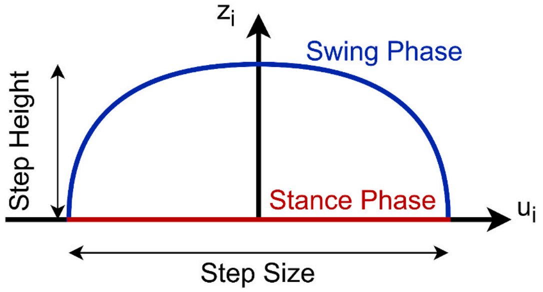

Fig. 5. Foot trajectory shape.

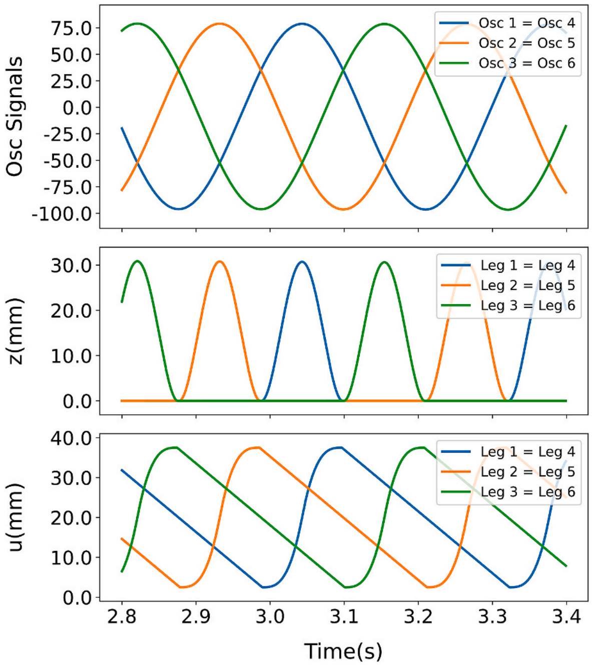

Fig. 6. Oscillators' outputs, vertical displacement z, and horizontal displacement u of the feet with respect to time. The gait here is Rice and $v_i = 3\ rad.s^{-1}$.

$$\ddot{\delta}_{yaw,i} = \mu_{yaw,i}^2\left(\delta_{target,i} - \delta_{yaw,i}\right) - 1.5\mu_{yaw,i}\dot{\delta}_{yaw,i} \tag{24}$$

$\delta_{yaw,i}$ can be modified online through equation (24) which ensures that $\delta_{yaw,i}$ will asymptotically and monotonically converge to the desired angle $\delta_{target,i}$. $\mu_{yaw,i}$ is a positive constant determining the convergence speed of state $\delta_{yaw,i}$ to $\delta_{target,i}$. Finally, the position of the i th foot w.r.t R_B can be obtained from previous equations as follows:

$$p_i^{R_B} = \begin{bmatrix} x_i^{R_B} \\ y_i^{R_B} \\ z_i^{R_B} \end{bmatrix} = \begin{bmatrix} x_{R_i}^{R_B} + sin\left(\delta_{yaw,i}\right) * u_i^j \\ y_{R_i}^{R_B} + \cos\left(\delta_{yaw,i}\right) * u_i^j \\ z_{R_i}^{R_B} + z_i^j \end{bmatrix} \tag{25}$$

Table 2
Matrices of different gaits.

$\Delta\varphi_{Wave}$	$\begin{bmatrix} 0 & -2 & -4 & -1 & -3 & -5 \\ 2 & 0 & -2 & 1 & -1 & -3 \\ 4 & 2 & 0 & 3 & 1 & -1 \\ 1 & -1 & -3 & 0 & -2 & -4 \\ 3 & 1 & -1 & 2 & 0 & -2 \\ 5 & 3 & 1 & 4 & 2 & 0 \end{bmatrix} \times \pi/3$
$\Delta\varphi_{Tetrapod}$	$\begin{bmatrix} 0 & -3 & -2 & 0 & -1 & -2 \\ 3 & 0 & 1 & 3 & 2 & 1 \\ 2 & -1 & 0 & 2 & 1 & 0 \\ 0 & -3 & -2 & 0 & -1 & -2 \\ 1 & -2 & -1 & 1 & 0 & -1 \\ 2 & -1 & 0 & 2 & 1 & 0 \end{bmatrix} \times \pi/2$
$\Delta\varphi_{Rice}$	$\begin{bmatrix} 0 & -2 & -1 & 0 & -2 & -1 \\ 2 & 0 & 1 & 2 & 0 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 \\ 0 & -2 & -1 & 0 & -2 & -1 \\ 2 & 0 & 1 & 2 & 0 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 \end{bmatrix} \times 2\pi/3$
$\Delta\varphi_{Tripod}$	$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \times \pi$

To maintain the validity of the foot trajectory during the transition from gait i to gait j, the term ψ is changed linearly from ψ_i to ψ_j over a period equal to the output signal's period $1/\nu_i$ of the oscillators which is close to the transition time.

As the considered robot consists of six legs that have an identical structure, the gait type of the hexapod robot primarily depends on the phase relationship between its six legs because it determines the alternation of these six legs during the walk. Therefore, an accurate phase relationship between these legs must be established to achieve movement in a specific gait pattern. For each gait type, the legs are divided into groups (let it M) so that in each group the legs are synchronized. The main factor that differentiates the gait types is M because it determines the maximum speed and stability of the robot using the given gait. Table 2 shows the phase relationship matrices corresponding to various gait types (each shown gait has a different M). The matrices are presented as a simple matrix (determines the alternation between the legs using simple indices) multiplied by a constant (which is 2π divided by M). For example, for $M = 6$ we have the wave gait, and $\Delta\varphi_{i,j} = -\pi/3$ indicates that the i th oscillator ahead in phase of the j th oscillator by $\pi/ 3$.

4. Experimental results

To the feasibility of our proposed method, we built a hexapod robot model (Fig. 1) based on the PyBullet simulation environment in Python. The model simulates the real robot based on the real dimensions and weights of its parts. The sample time is fixed at 1 *ms* and we set the coefficient of friction with the ground to 0.8.

The proposed control architecture is verified on a real hexapod robot (Fig. 2) in the laboratory environment. The robot consists of 18 servo motors of type *Dynamixel AX-12A* that represent the joints of the robot, *Arduino Mega (Microcontroller: ATmega2560)*, and *Bluetooth Module (HC-05)* to receive the online commands from PC. The dimensions defined in Table 1 and Fig. 3 are $L_0 = 100\,mm$, $L_1 = 53\,mm, L_3 = 68\,mm, and\ L_4 = 107\,mm$. The sampling period used in the verification process is 35 *ms* and it meets the minimum possible time for the Arduino board to do all the required calculations. The snapshot of robot's performance during the experimental process is shown in Fig. 7.

Generating an accurate foot trajectory requires prior knowledge of the shape of the output signals of the CPG network. This is due to the dependence of the foot trajectory generator on the specific shape properties of the CPG outputs. Ensuring the preservation of the shape of these output signals appears to be a critical factor during gait transitions.

Our control architecture underwent extensive testing on various gait transitions, as shown in Fig. 9, where the CPG output consistently maintained a semi-sinusoidal shape throughout the transition process, thus ensuring efficient foot trajectory generation. The term D_i is shown to highlight its relationship with the output signals frequency. Notably, the frequency of an oscillator decreases when the value of D_i is negative and vice versa as demonstrated by the observed patterns. The integration of sharp and fast response of D_i to the gait changing coupled with the utilization of the *arctan* function facilitates fast transitions without distortion. It can be noticed that changing the gait while changing the velocity has no effect on the efficiency of the proposed architecture.

Online changing of foot trajectory parameters (step size and step height) has been verified in Fig. 8 where we can notice the resulted change in the oscillator signal so that increasing the step size mainly appeared as an increase in the amplitude of the oscillator signal, while changing the step size affected the DC offset of the oscillator output.

In order to demonstrate the improvements achieved in our modified CPG, a comparison is made between our architecture and the CPG presented in Ref. [31] (will be referred as traditional CPG). However, the inherent challenge with the traditional CPG is the strong correlation between smoothness and the network latency in responding to the transition. Fig. 10 shows a transition from Rice gait to Tripod gait using our modified CPG, and we notice that the shape of signals almost maintained its sinusoidal shape. At the same time, Fig. 11 shows a transition using the traditional CPG. However, it is accompanied by a small delay in response and an obvious distortion in the signals, making it unsuitable for generating an accurate foot trajectory. The problem of the distortion appears clearly in the blue

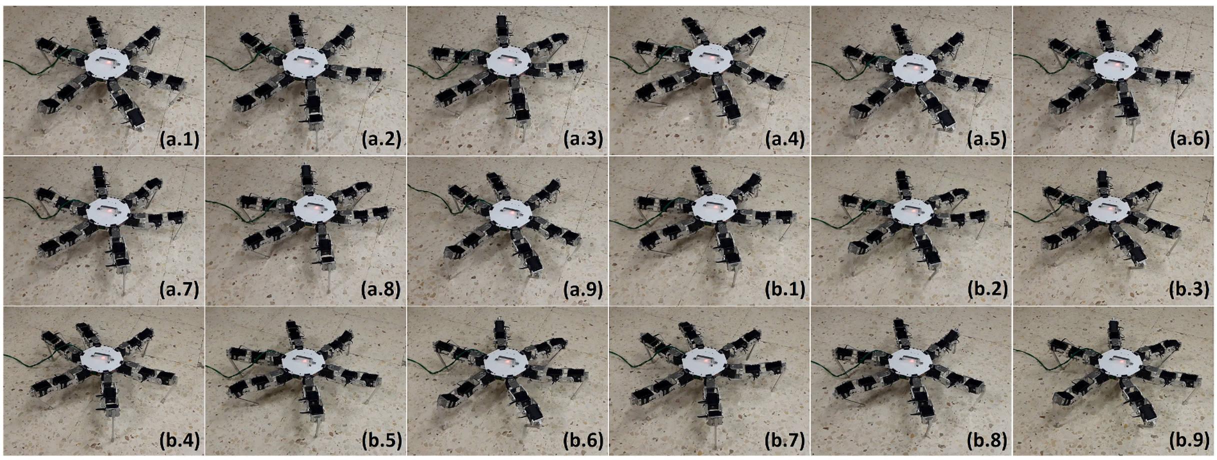

Fig. 7. Implementation of our proposed CPG on the real Hexapod robot. The first 9 snapshots show transition from Tetrapod to Tripod, and the second 9 snapshots show transition from Tetrapod to Rice.

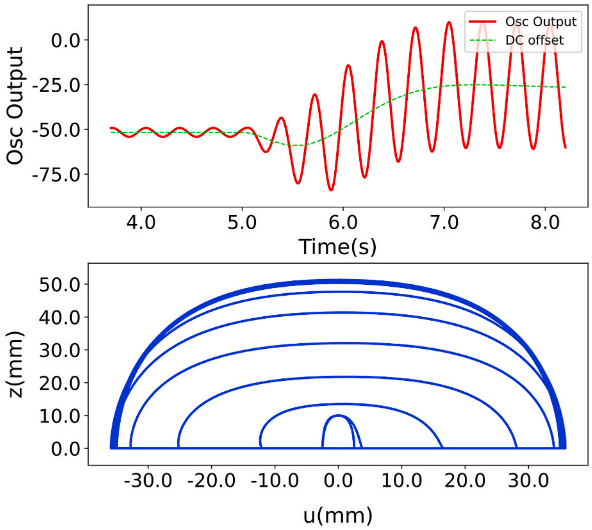

Fig. 8. *Foot trajectory with oscillator output while changing the step size from (5 to 70) mm and step height from (10 to 50) mm simultaneously using* $\mu_{a_i} = \mu_{c_i} = 3$.

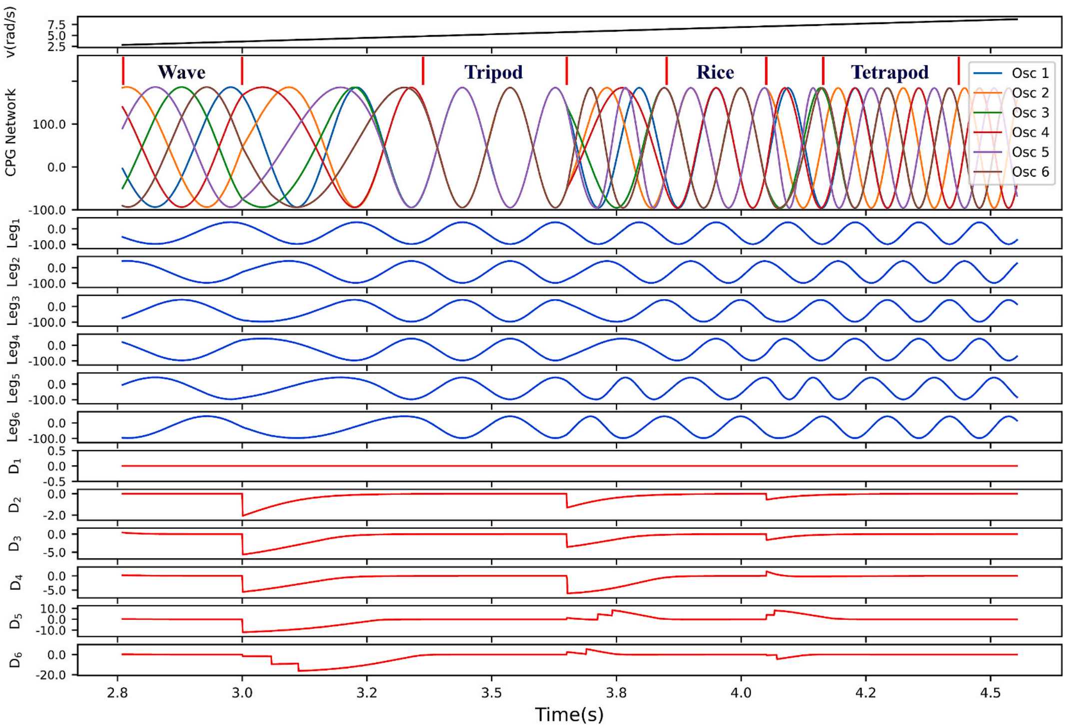

Fig. 9. Gait transitions between different gaits while changing the angular velocity of oscillators from (2.5–7.5) $rad.s^{-1}$. The term D_i is shown to demonstrate how it affects the instantaneous frequency of oscillator to synchronize the network. $\alpha = 15$ in equation (17).

and orange lines (Osc 1 and Osc 2) as they changed their directions without completing their duty cycle which made the signals lose its sine shape. Besides, the realization of a less distortion results required significant long delay response as illustrated in Fig. 12. However, the distortion still exists, and it cannot be eliminated by just adjusting the parameter of the network.

To summarize the results, our control architecture was able to complete the transition between any different gaits in almost the output period of the CPG oscillators. This transition time (determined relative to the frequency of the CPG network) is relatively small and well defined for real applications that have variety in operating frequency. At the same time, the shape of the oscillators output (during gait transition) has met some specification by which we consider that the CPG signals have a semi-sinusoidal shape, and the

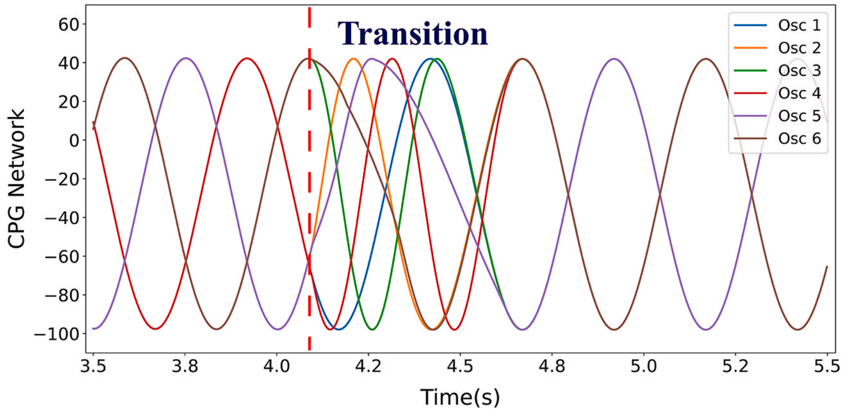

Fig. 10. *The transition from Rice gait to Tripod gait at the second 4.1 using our modified CPG where* $\alpha = 15$ in equation (17).

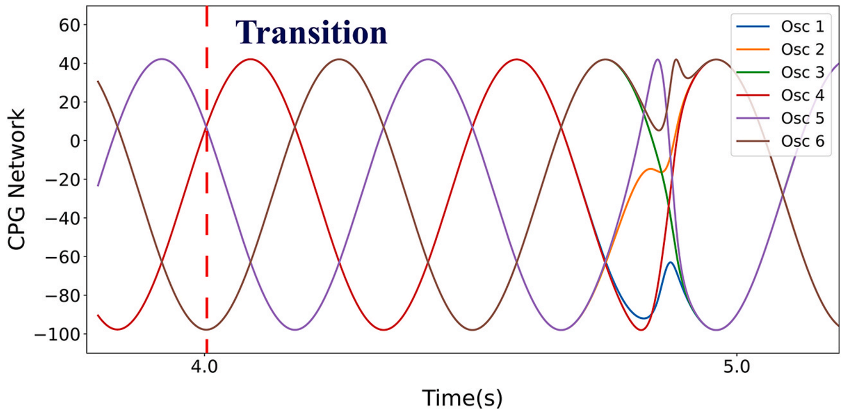

Fig. 11. transition from Rice gait to Tripod gait at second 4 using the traditional CPG with $\lambda = 15$ [31].

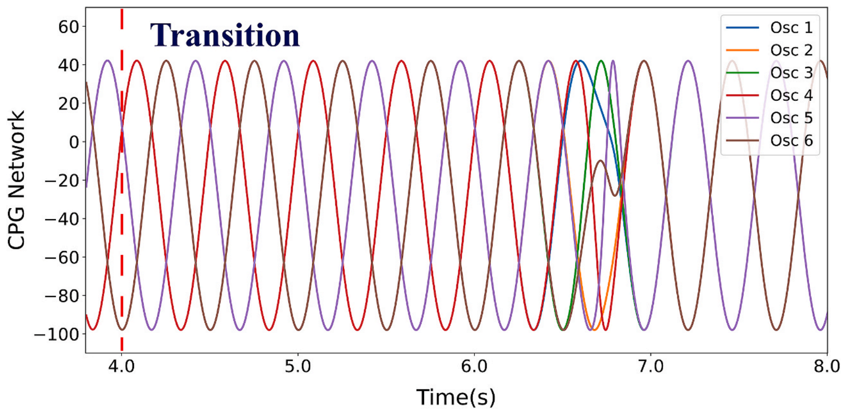

Fig. 12. transition from Rice gait to Tripod gait at second 4 using the traditional CPG with $\lambda = 5$ [31].

transition is smooth. **First,** the CPG output signals are derivable w.r.t time. **Second,** at any small interval of time, the shape of the output is part of a sinusoidal shape with a specific frequency, and this is guaranteed by equation (18). **Third,** at any point of time if the derivative of the output signal is zero, then the value of the output signal at that point is $c_i \pm a_i$ which means that there are no invalid turning points like the one illustrated in *Osc 6 in* Fig. 11. We can see that the CPG outputs shown in Fig. 9 meet the previous three conditions and the shape of the signals during transition appears as a sinusoidal shape with continuously changing frequency between two frequencies. As a last point, our foot trajectory generator was able to generate a valid semi-ellipse shape during an online adjustment to the step parameters as shown in Fig. 8.

Despite all the mentioned features accomplished in our proposed control methodology, the robustness of our CPG network in real environments still requires some improvements. The proposed method adopts an open loop type of control which prevents the robot from interacting with the surrounding environment. Thus, the proposed architecture needs to be integrated with sensory feedback and this is the subject of an ongoing project.

5. Conclusion

In this paper, we have proposed an improved control methodology to achieve a smooth gait transition for a hexapod robot using Central Pattern Generators (CPGs). We used a modified CPG network consisting of six individual CPG oscillators to generate rhythmic signals. The coupling relationship between oscillators is considered to enhance transition time and smoothness. The CPG output is used to generate the foot trajectory. Independence between gait parameters was considered in our workspace trajectory generator by the well integration with the oscillators' phases. The proposed CPG network architecture enables the generation of different gaits and guarantees smooth transitions between them as the transition time falls close to the output period of the CPG oscillators. The architecture considered online adjustability of trajectory parameters (step size and height) besides the speed and the direction of the robot, enhancing the robot's adaptability to varying environmental conditions. Simulation results confirmed the effectiveness of the modified oscillator in achieving smoother gait transitions, while experiments with a physical robot demonstrated the feasibility and efficiency of our approach. However, this study contributes to a larger ongoing project focused on compensating for malfunctions using pain sensing as a feedback mechanism, which allows the robot to engage effectively with its environment. This work contributes to advancing the locomotion capabilities of legged robots, enabling them to navigate complex terrains with improved stability, efficiency, and fast adaptability.

Data availability statement

Data will be made available upon request.

CRediT authorship contribution statement

Kifah Helal: Conceptualization, Methodology, Software, Investigation, Writing – Original Draft, Writing – Review & Editing. **Ahed Albadin:** Conceptualization, Writing – Review & Editing, Supervision. **Chadi Albitar:** Writing – review & editing. **Michel Alsaba:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Zhijun Chen, Jimu Liu, Feng Gao, Real-time gait planning method for six-legged robots to optimize the performances of terrain adaptability and walking speed, Mech. Mach. Theor. 168 (2022) 104545.
[2] J. Coelho, F. Ribeiro, B. Dias, G. Lopes, P. Flores, Trends in the control of hexapod robots: a survey, Robotics 10 (3) (Aug. 2021) 100.
[3] Y. Zhao, J. Wang, G. Cao, Y. Yuan, X. Yao, L. Qi, Intelligent control of multilegged robot smooth motion: a review ([CrossRef], IEEE Access (2023).
[4] G. Haynes, R.I.Z.Z.I. Clark, A. Alfred, Gaits and gait transitions for legged robots, in: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006, IEEE, 2006, pp. 1117–1122.
[5] Z. Chen, J. Li, S. Wang, J. Wang, L. Ma, Flexible gait transition for six wheel-legged robot with unstructured terrains, Robot. Autonom. Syst. 150 (2022 Apr 1) 103989.
[6] A. Altan, Performance of metaheuristic optimization algorithms based on swarm intelligence in attitude and altitude control of unmanned aerial vehicle for path following, in: 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2020, pp. 1–6. Istanbul, Turkey.
[7] A. Altan, R. Hacıoğlu, Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances, Mech. Syst. Signal Process. 138 (2020 Apr 1) 106548.
[8] Auke Jan Ijspeert, Central pattern generators for locomotion control in animals and robots: a review, Neural Network. 21 (4) (2008) 642–653.
[9] Kiyotoshi Matsuoka, Mechanisms of frequency and pattern control in the neural rhythm generators, Biol. Cybern. 56 (5–6) (1987) 345–353.
[10] Volker Durr, et al., Motor flexibility in insects: adaptive coordination of limbs in locomotion and near-range exploration, Behav. Ecol. Sociobiol. 72 (2018) 1–21.
[11] Auke Jan Ijspeert, et al., From swimming to walking with a salamander robot driven by a spinal cord model, Science 315.5817 (2007) 1416–1420.
[12] Eve Marder, Dirk Bucher, Central pattern generators and the control of rhythmic movements, Curr. Biol. 11 (23) (2001) R986–R996.
[13] Chengju Liu, Qijun Chen, Danwei Wang, CPG-inspired workspace trajectory generation and adaptive locomotion control for quadruped robots, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 41 (3) (Jun. 2011) 867–880.
[14] Deniz Korkmaz, et al., Locomotion control of a biomimetic robotic fish based on closed loop sensory feedback CPG model, Journal of Marine Engineering & Technology 20 (2) (2021) 125–137.
[15] B. Wang, K. Zhang, X. Yang, X. Cui, The gait planning of hexapod robot based on CPG with feedback, Int. J. Adv. Rob. Syst. 17 (3) (May 2020) 172988142093050.
[16] Steven Strogatz, Synchronization: a universal concept in nonlinear sciences, Phys. Today 56 (1) (2003) 47, 47.
[17] J. Gómez-Gardeñes, Y. Moreno, A. Arenas, Paths to synchronization on complex networks, Phys. Rev. Lett. 98 (3) (Jan. 2007).
[18] H. Kimura, Y. Fukuoka, Adaptive dynamic walking of a quadruped robot on irregular terrain using a neural system model, in: Robotics Research: the Tenth International Symposium, vol. 6, Springer Berlin Heidelberg, Oct. 2000, pp. 147–160, no. 978–3540-005506.
[19] Azhar Aulia Saputra, Auke Jan Ijspeert, N. Kubota, A neural primitive model with sensorimotor coordination for dynamic quadruped locomotion with malfunction compensation, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct. 2020, pp. 3783–3788, no. 2153-0866.

[20] J. Nassour, P. Hénaff, F. Benouezdou, G. Cheng, Multi-layered multi-pattern CPG for adaptive locomotion of humanoid robots, Biol. Cybern. 108 (3) (Feb. 2014) 291–303.
[21] Y.U. Haitao, et al., Gait generation with smooth transition using CPG-based locomotion control for hexapod walking robot, IEEE Trans. Ind. Electron. 63 (9) (2016) 5488–5500.
[22] Linlin Shang, et al., Smooth gait transition based on CPG network for a quadruped robot, in: 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), IEEE, 2019, pp. 358–363.
[23] Sai Gu, et al., Gait transition method for quadruped robot based on CPG network and impedance control, in: 2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM), IEEE, 2020, pp. 387–391.
[24] Wei Zhang, et al., Central pattern generators for locomotion control in hexapod robot legs, in: 2022 34th Chinese Control and Decision Conference (CCDC), IEEE, 2022, pp. 3988–3993.
[25] S. Grillner, Biological pattern generation: the cellular and computational logic of networks in motion, Neuron 52 (5) (Dec. 2006) 751–766.
[26] A. Frigon, Central pattern generators of the mammalian spinal cord, Neuroscientist 18 (1) (Apr. 2011) 56–69.
[27] J. Yu, M. Tan, J. Chen, J. Zhang, A survey on CPG-inspired control models and system implementation, IEEE Transact. Neural Networks Learn. Syst. 25 (3) (2013 Sep 20) 441–456.
[28] C. Liu, Y. Chen, J. Zhang, Q. Chen, CPG driven locomotion control of quadruped robot, in: Proc. IEEE Int. Syst. Man Cybern., Oct. 2009, pp. 2368–2373. San Antonio, TX, USA.
[29] J.-K. Ryu, N.Y. Chong, B.J. You, H.I. Christensen, Locomotion of snake-like robots using adaptive neural oscillators, Intelligent Service Robotics 3 (1) (Aug. 2009) 1–10.
[30] Yoshiki Kuramoto, Self-entrainment of a population of coupled non-linear oscillators, in: International Symposium on Mathematical Problems in Theoretical Physics: January 23–29, 1975, Kyoto University, Kyoto/Japan, Springer Berlin Heidelberg, 1975, pp. 420–422.
[31] Chengju J. Liu, Danwei W. Wang, Qijun J. Chen, Locomotion control of quadruped robots based on workspace trajectory modulations, Int. J. Robot Autom. 27 (4) (2012) 345.