IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 3, September 2024, pp. 3545~3558
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i3.pp3545-3558 ❒ 3545

A Horner's polynomial based quadrupedal multi-gaits signal generation controller

Kouame Yann Olivier Akansie[1], Rajashekhar C. Biradar[1], Karthik Rajendra[1], Geetha D. Devanagavi[2]
[1]School of Electronics and Communication Engineering, REVA University, Bangalore, India
[2]School of Computing and Information Technology, REVA University, Bangalore, India

Article Info

Article history:

Received Oct 5, 2023
Revised Oct 22, 2023
Accepted Dec 2, 2023

Keywords:

Gait cycle generation
Gait planning
Horner's polynomials
Mobile robots
Quadruped locomotion

ABSTRACT

Animal locomotion is the process through which animals move from one location to another. Self-propelled locomotion is based on the animal performing a series of actions to move towards a predetermined target. All of these motions occur sequentially and repeat themselves during a gait cycle. A gait cycle may be simulated by duplicating each motion in the cycle sequentially. To achieve this goal, a problem known as the gait planning issue was formulated, in which various systems were created to provide suitable signals for the execution of distinct gaits (patterns of steps of an animal at a specified speed). This research approaches the problem using Horner's polynomials for quadruped robots. The approach entails first creating a sequence table for each gait and fit two polynomial equations. In this study, an attempt is made to combine several gaits using Horner's polynomials. An algorithm uses elaborated polynomials to generate the desired gaits signals.

Corresponding Author:

Kouame Yann Olivier Akansie
School of Electronics and Communication Engineering, REVA University
Rukmini knowledge park, Kattigenahalli, Yelahanka, Bangalore 560064, India
Email: akansiekouame@gmail.com

1. INTRODUCTION

Mobile robots have various applications, depending on their features. An important parameter that decides the type of application a mobile robot can be used for is the locomotion mechanism employed. The most suitable terrestrial locomotion mechanism considering the complexity of the terrains to be navigated, is legged locomotion, especially quadruped locomotion. Indeed, quadruped locomotion offers stability over unknown and irregular environments compared to other locomotion types. The main challenge to overcome with quadruped locomotion is the generation of gaits signals, considering the footfall sequence and legs phase difference. Various studies helped researchers understand and therefore mimic some of the mechanisms employed in generating different gaits through different models. Such models can be classified into model-based approaches (zero moment point model) [1]–[4], bio-inspired approaches (central pattern generators (CPG) models) [5]–[12], and learning approaches (reinforcement learning, deep learning approaches) [13]–[16].

A model-based approach for gait generation is discussed in the paper [17], which focuses on bounding gait and proposes a gait generation technique that considers the intensity of ground impact due to short swing periods and high angular accelerations. The use of SLIP led to the reduction of possible oscillations. The simulation results obtained were better than those powered by Bezier polynomials [18]–[21]. Another method is described in the paper [22] for solving static gait planning problems. The authors propose two approaches that increase the walking speed while maintaining the robot's stability.

Ma and Ames [23] discusses a method employing a computational model for gait generation. In their work, the authors correlate quadruped and biped robots to generate a biped walking gait for quadruped robots and vice versa. Fukui *et al.* [24] discusses a bio-inspired approach that uses a vestibular feedback applied to a CPG to generate quadruped motions. The proposed vestibular feedback produces a modulation that allows quadruped robots to get over unforeseen obstacles during high-speed gaits like galloping, ensuring a smooth transition in case of gait change. Takei *et al.* [25] proposes a neuron model constructed around a self-inhibited pulse instead to generate a quadrupedal walking and trotting-like signals. A method to generate gaits for quadruped robots without relying on a CPG is proposed in the paper [26], which discusses using a variable artificial potential field to generate a signal for moving the robot's feet along a specific trajectory. The foot trajectory is modeled considering the stance and swing phases involved in the movement of a single leg. This method is proven effective as the authors could execute a dynamic walking gait.

The most used approach for gait generation employs a CPG. For instance, Ma *et al.* [27] highlights the benefits of a Hopf oscillator in the generation of quadruped. The authors use a Hopf oscillator to generate both gait signal and phase relationship between the legs. Another example is shown in a paper [28], which presents a CPG-based gait generator for hexapod robots with curved legs. The authors use a modified Hopf oscillator as a control unit for the CPG, which allows a smooth transition between different gaits. CPG-based approaches can be combined with various approaches. Such combination is explored in [29], where the authors attempt to solve the gait planning problem with a phase-guided reinforcement learning framework. They mapped the phase differences that exist between the four legs to four-foot trajectories using a long short-term memory (LSTM) neural network, considering the previously established control rules. For the implementation of the CPG, the authors used a Hopf oscillator because it can be used to achieve robust and controllable gaits with smooth transitions. Such a system would solve, by default, the multitasking problem experienced when multiple gaits must be learned. A different learning approach is proposed in the paper [30] for gait generation decision-making. The authors propose a new machine learning algorithm based on Maslow's hierarchy of needs, which can be applied to many fields, including robotics. The proposed algorithm is similar to a traditional machine learning algorithm, with the only difference being that the working principle relies on a different approach, i.e., Maslow's hierarchy of needs. The literature survey in the field revealed two essential aspects: the generation of the gait signal itself, and the generation of the phases between the legs' signals and various gaits. Hence, any suitable technique can be used to substitute any aspect in the gait generation process.

This paper proposes a bio-inspired method that uses a CPG designed with Horner's polynomials to generate quadruped gaits signals. The work explores the possibility of using an alternative method to construct a CPG that requires less parameters to generate various quadrupedal gaits' signals. The methodology of the research consists of studying various quadruped gaits first, then extracting natural gaits sequences, and finally establishing relationships between the gaits' sequences and phase differences through a two-dimensional polynomial model for the generation of quadruped gaits signal.

2. METHOD

Generating a quadrupedal gait involves a set of complex motions that need to be coordinated and controlled. Control systems designed to supervise and control such motions are called CPGs, as they generate the appropriate signal required for the execution of a specific gait. The most commonly used approach in designing a CPG employs a Hopf-based oscillator for its ability to generate sustained oscillations. This research aims to study the possibility of substituting such a system with another powered by Horner's polynomials. The use for this study is as follows: i) study quadruped locomotion behavior and extract the leg motion sequences, then ii) design a controller from the equation elaborated from the leg motion sequences, and finally iii) propose an algorithm for quadrupedal gait signal generation.

2.1. Quadruped locomotion behaviour

Overcoming rugged terrains is possible with an appropriate locomotion mechanism. As felines are agile and versatile, understanding their locomotive behavior is necessary for designing a quadruped locomotion mechanism [31]. Considering cats, we distinguish mainly three types of movements, based on age: i) fetal movements, which concern movements that occur when the cat is still in the womb; ii) pre-walking movements (infant cat), which deal with movements that come at birth, before actual walking movements; and finally iii) walking and others movements (adult movements) [32]. Such movements mature weeks after birth. When this period is reached, the cat can walk, amble, trot, pace, canter, and gallop. Based on the task, the cat selects any of these movements. Each motion behavior offers a different speed that can be reached. The motion behaviors, also known as gaits, range from low speed to high-speed gait, requiring specific leg coordination dictated by the gait being executed [33]. A gait is just a repetition of a sequence of leg movements in a coordinated way, which results in a specific movement. The same leg movement is executed by all the legs with a time shift, creating different sets of movements to achieve the desired speed.

2.1.1. Walking gait sequence

Walking is a four-beat gait which means that each leg contacts the ground at a separate time during the movement. During the execution of this gait, at least two feet must be in contact with the ground, which makes it the easiest and least tiring gait. We distinguish different types of walks based on stride length, limb support, and step speed. Basic variations comprise regular walk, power walk, and quick walk variations. When a usual walk is to be executed, there is an alternation between two limbs and three limbs of support, which results in eight possibilities. Figure 1 represents the walk step sequence of a cat from which a step sequence can be extracted with the following considerations: only two phases out of four (lift, swing, support, thrust) are considered in establishing the leg sequence table, where "1" represents the contact of a leg with the ground, and '0' represents no contact with the ground. Table 1 shows the extracted step sequence for the specific gait.

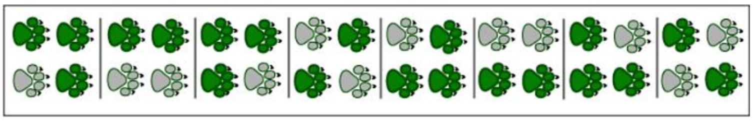

Figure 1. Quadrupedal walking gait step sequence

Table 1. Four-beat walking gait step sequence

Steps (t)	1	2	3	4	5	6	7	8
Leg 1 (front-left)	1	1	1	1	1	0	0	0
Leg 2 (front-right)	1	0	0	0	1	1	1	1
Leg 3 (rear-right)	0	0	1	1	1	1	1	0
Leg 4 (Rear-left)	1	1	1	0	0	0	1	1

2.1.2. Ambling gait sequence

Ambling gait is also a four-beat gait used for average-speed movement. It lies between the walk and pace gait. It's often known as a slow pace or running walk. The leg movements are similar to that of the walking gait, with a slight difference in the timing of each leg. Amble can be done slowly and faster as well. A typical cycle would have an alternation of right diagonal, left lateral, left diagonal, and right lateral (Figure 2). On the other hand, there is an alternation of two different support structures involving a single limb and two limbs. Considering Figure 2, which represents a four-step sequence ambling gait, a status table (Table 2) can be established, following the same considerations as in walking gait sequencing. This four-step sequence will be converted to an eight-step sequence using padding to match the walking gait's table.

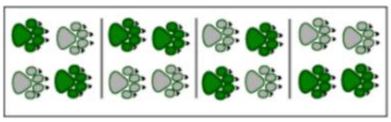

Figure 2. Quadrupedal ambling gait step sequence

Table 2. Four-beat ambling gait step sequence

Steps (t)	1	2	3	4	5	6	7	8
Leg 1 (front-left)	0	0	1	1	1	1	0	0
Leg 2 (front-right)	1	1	0	0	0	0	1	1
Leg 3 (rear-right)	0	0	0	0	1	1	1	1
Leg 4 (Rear-left)	1	1	1	1	0	0	0	0

2.1.3. Trotting gait sequence

When felines have to move at a relatively reasonable speed for long distances, trotting gait is used. Unlike previous gaits, this gait is a two-beat symmetric gait which is, by the way, less tiring. This gait uses less body movement around the center of gravity. As fewer movements are involved, it helps achieve stability while moving. In the execution of this gait, the contralateral hindfeet and forefeet hit the ground at the same time. Trotting can be slow or quick, depending on the speed required. A quick trot or running trot involves a suspension phase between the diagonal phases. However, interferences might occur if this gait is executed for a long duration. The lateral hind paw is made to hit the ground only after the fore paw is lifted to overcome such interferences. The main difference between both slow and running trots is the presence of a suspension phase as there is no suspension, and the whole-body weight is carried diagonally. A typical trotting gait leg sequence is shown in Figure 3. Considering the same assumptions made as in the walking gait, a table (Table 3) for trotting step sequence can be established with padding in order to convert it to an 8-step gait sequence.

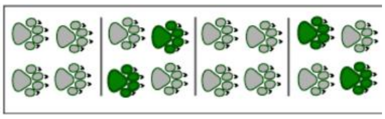

Figure 3. Quadrupedal trotting gait step sequence

Table 3. Two-beat trotting gait step sequence

Steps (t)	1	2	3	4	5	6	7	8
Leg 1 (front-left)	0	0	1	1	0	0	0	0
Leg 2 (front-right)	0	0	0	0	0	0	1	1
Leg 3 (rear-right)	0	0	1	1	0	0	0	0
Leg 4 (Rear-left)	0	0	0	0	0	0	1	1

2.1.4. Pacing gait sequence

Figure 4 is the representation of the steps involved in pacing. Pacing is a two-beat gait as two lateral limbs are used simultaneously to support the whole body. In other words, the forelimb and hind limbs have the same movement, alternating with each side. Sometimes, a slight shift is observed between two lateral limbs. Although this happens, it does not disturb the actual execution of the gait. Pacing involves a suspension period, as observed in trotting. The suspension period is used as a transition between the alternation of left and right limbs, resulting in an extended flying episode. At slow speed, the flying phase is less than the supporting phase, and although lateral supports dominate this gait, diagonal supports might occur between two lateral support phases. Pacing has an advantage over trotting as it requires less muscular exertion, thereby, fewer vertical oscillations. Table 4 represents the step sequence gait table considering the same assumptions as for previous gaits. Since only four steps are used to represent this gait, padding will be used to convert it to an 8-step gait sequence.

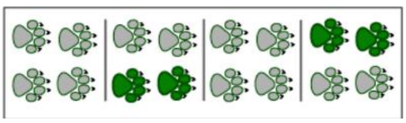

Figure 4. Quadrupedal pacing gait step sequence

Table 4. Two-beat pacing gait step sequence

Steps (t)	1	2	3	4	5	6	7	8
Leg 1 (front-left)	0	0	0	0	0	0	1	1
Leg 2 (front-right)	0	0	1	1	0	0	0	0
Leg 3 (rear-right)	0	0	1	1	0	0	0	0
Leg 4 (Rear-left)	0	0	0	0	0	0	1	1

2.1.5. Cantering gait sequence

Cantering is represented by four main steps, as shown in Figure 5. The speed should be increased gradually from lower speed gaits to higher speed gaits to arrive at cantering. Cantering can therefore be executed only when the previous gait implies suspensions phases. However, the starting suspension phase can be moved to the last step as it does not affect the properties of the gait. Cantering is a relatively high-speed gait, which comes just before galloping. This gait is often called sustained, slow, or middle gallop, as both gaits share some properties. This gait is a three-beat gait that includes high-speed suspension periods. It can be seen from the gait step sequence that this gait is asymmetrical since there is a difference in the limb patterns on each side. The canter gait is preferred for looping or cruising at relatively high speed on rough grounds and irregular terrains. A step sequence table (Table 5) can be elaborated from the figure, using padding to convert it to an eight steps sequence.

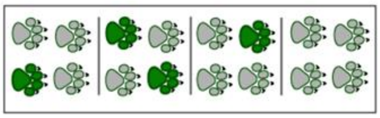

Figure 5. Quadrupedal cantering gait step sequence

Table 5. Three-beat cantering gait step sequence

Steps (t)	1	2	3	4	5	6	7	8
Leg 1 (front-left)	0	0	0	0	1	1	0	0
Leg 2 (front-right)	0	0	1	1	0	0	0	0
Leg 3 (rear-right)	1	1	0	0	0	0	0	0
Leg 4 (Rear-left)	0	0	1	1	0	0	0	0

2.1.6. Golloping gait sequence

Two types of gallops are most used: transverse gallop and rotary gallop. A rotary gallop is faster than a transverse gallop which accounts for five main steps against six steps for a rotary gallop. The rotary gallop is a four-beat gait that involves two suspension phases. It is called a rotary gallop because of the rotation of the limb impact pattern, as seen in Figure 6. The rotary gallop is the fastest gait the animal can execute, thereby requires more energy. Such a gait is used when speed is needed to chase or run out from enemies. A step sequence table (Table 6) can be elaborated from Figure 6, considering the same assumptions as others gaits. Since only six steps are involved in this representation, padding will convert it to an eight steps gait sequence.

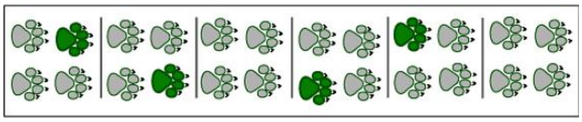

Figure 6. Quadrupedal galloping gait step sequence

Table 6. Four-beat galloping gait step sequence

Steps (t)	1	2	3	4	5	6	7	8
Leg 1 (front-left)	1	0	0	0	0	0	0	0
Leg 2 (front-right)	0	1	0	0	0	0	0	0
Leg 3 (rear-right)	0	0	0	0	1	0	0	0
Leg 4 (Rear-left)	0	0	0	0	0	1	0	0

2.2. Gait generation controller design

The process proposed to generate a multi-gaits quadrupedal signal is shown in Figure 7. The essential parameters required to drive a robot are speed, gait, and direction. As gait is a function of speed in the proposed process, speed and direction are sufficient to drive the robot in a forward direction.

Once speed and direction are provided, the supplied speed is analyzed to determine the corresponding gait. For example, a speed ranging from 0 to 0.8 m/s corresponds to walking, whereas a speed of 2.3 to 3.7 m/s corresponds to canter gait. Once that decision is made, the gait index is extracted based on its position in the speed chart. As a result, the walking gait will be assigned index one as the lowest speed gait, and the gallop gait will be assigned index six as the highest speed gait. Since the prescribed s peed can be achieved with the help of movements executed in each step, the time allocated for each step is calculated. The extracted gait index is employed in two functions: i) the original sequence function and ii) the step shift function. The original sequence function is a n^{th} degree polynomial function that uses the gait index as a variable and produces a sequence of eight steps corresponding to the first limb step sequence. On the other hand, the step shift function is an n^{th} degree polynomial function that generates the step shifts required to generate each limb step sequence from the original limb sequence. Finally, we are left with a step sequence for each limb based on the selected gait. The proposed design presents two aspects: internal steps and external steps. The flow diagram presents focusses on the external steps aspect rather than internal steps. The internal steps take care of the transition during the external steps sequences. Internal steps, considering a single limb motion, can be divided into four steps: i) lift, ii) swing, iii) thrust, and iv) support. The upcoming sections will deal with the original and step-shift sequence generation functions.

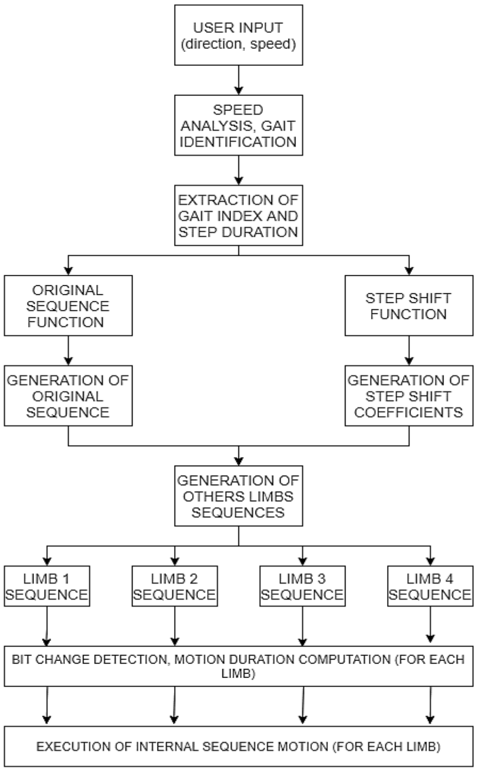

Figure 7. Quadrupedal gait step sequence generation process

2.2.1. Original sequence generation polynomial function

The original sequence generation polynomial function aims to generate the first limb step sequence with respect to the gait to be executed. The methodology that is intended to be used in designing the function is the following: first study different gaits, compare and find similarities between the step sequence of each limb, extract the original step sequence or first limb step sequence, and classify the gaits considering their maximum achievable speed and assign an index to each gait following the classification, draw a table relating all the gaits and their initial steps sequences, fit a 2-dimensional polynomial equation with data available in the gait/step sequence table using Horner's polynomials. Table 7 is a summary of previous step sequence tables, considering the original sequence only, the same sequence is repeated for each leg, but with different time shifts.

Table 7. Multi-gaits cycle sequence relationship table

Gait	Steps	1	2	3	4	5	6	7	8
Walk	1	1	1	1	1	1	0	0	0
Amble	2	0	0	1	1	1	1	0	0
Trot	3	0	0	0	0	1	1	0	0
Pace	4	1	1	0	0	0	0	0	0
Canter	5	1	1	0	0	0	0	0	0
Gallop	6	1	0	0	0	0	0	0	0

Let G(i,t) be the original step sequence function which maps each gait to its original sequence. Let "i" be the gait associated index ranging from 1 to 6 and "t" the time corresponding to the generated step in the step sequence ranging from 1 to 8 with T being the gait cycle such that $T = \sum_{a=1}^{8} t_a$. Considering Table 7, the multi-gaits cycle sequence relationship can be obtained in three dimensions i.e., gait index 'i', step 't' and footfall data 'S' such that G(i,t)=S. The aim is to find the corresponding 2D Horner's polynomial function to fit the values available in the table. The simplified form of a 2D Horner's polynomial function that satisfies the requirements, assuming 'n' and 'm' to be both equal to 1 is given as:

$$\mathrm{G}(\mathrm{i},\mathrm{t}) = \mathrm{a}_1 * \mathrm{i} * t + \mathrm{a}_2 * t + \mathrm{a}_3 * \mathrm{i} + a_4 \quad (1)$$

From (1), only four coefficients need to be found for the corresponding equation. The algorithm used to find these coefficients is shown in Figure 8. In fitting Horner's 2D equation for generating multi-gaits signals, data for 'i', 't,' 'S,' 'm', and 'n' needs to be given first. A graphical representation of the input data as a surface for which an equation should be fitted is generated. The input data for 'i' and 't' is augmented and represented as a matrix of 11 by 11 to match the constraints of the augmentation function (mesh grid). Once the requirement for the 2D curve fitting function (polyFit2D) is satisfied, the corresponding coefficient for the 2D equation is calculated. The fitted curve is evaluated for the initial input data of 'i', 't' and 'S.' once done, the error for each value of 'S' with a combination of input values of 'i' and 't' is calculated to arrive at an average error value. The average value is then added to the whole equation for optimization, along with a constant parameter 'C'; the value of 'C' is varied until the average error reaches zero after rounding off the values obtained during the valuation of the fitted equation. The obtained coefficients for the fitted equations are discussed in the next section.

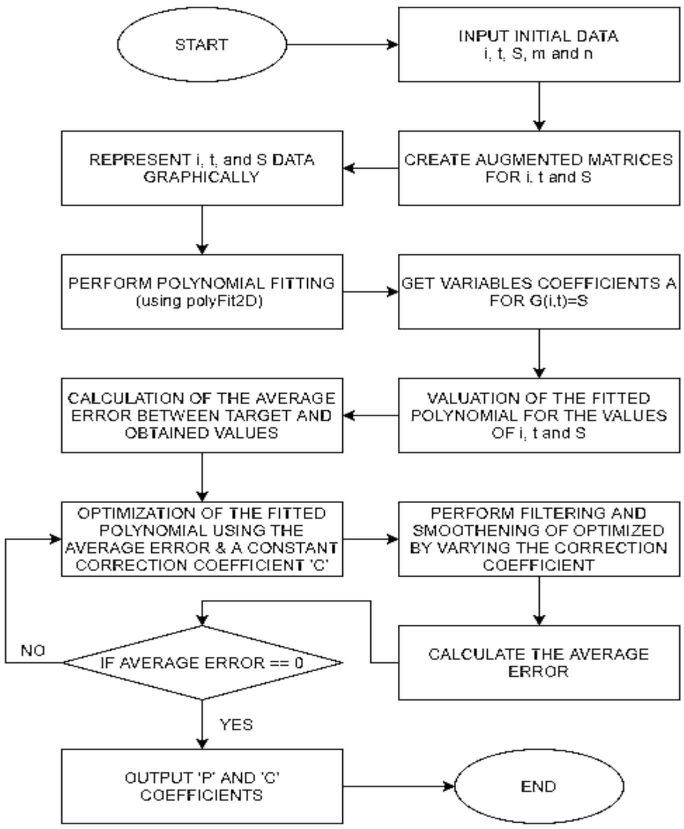

Figure 8. Multi-gaits/phase signal generation process using Horner's method

2.2.2. Step or phase shift coefficient generation polynomial function

Once the original sequence function generates the sequence for a specified gait, the steps' sequences for other limbs should be derived from the generated original step sequence. Table 8 establishes the relationship between the reference (first limb) and others step sequences. L1, L2, L3, and L4 represent the four limbs of the quadruped robot and the columns headings represent the gait index associated to each gait. The same methodology employed to design the original step sequence is applied here to design the step shift coefficient function as well. Let $P(i,l)$ be the phase shift function which generates the steps shifts associated relating the first limb to the others for a given gait. Let "i" be the gait associated index ranging from 1 to 6 and "l" be the limb index ranging from 1 to 4; using the same algorithm depicted in Figure 8, such that $P(i,l) = s$; with 's' being a matrix corresponding to the legs' phases in the table. Considering the polynomial degree 'n' and 'm' for 'i' and 'l' respectively to be equal to 1, the corresponding Horner's equation is given in (2):

$$P(i,l) = a_1 * i * l + a_2 * l + a_3 * i + a_4 \quad (2)$$

Table 8. Multi-gaits phase shift relation table for quadruped robots

Gait	Index/Legs	L1	L2	L3	L4
Walk	1	0	4	2	6
Amble	2	0	4	2	6
Trot	3	0	4	0	4
Pace	4	0	4	4	0
Canter	5	0	6	4	6
Gallop	6	0	1	4	5

2.2.3. Algorithm for gait generation

As the main goal is to generate a quadrupedal gait, let's assume a gait cycle to be a complete motion of all the four legs independent from the execution sequence. Hence, to execute a motion cycle, a user input is required first to supply important parameters to the system. Then, the legs have to move in a specific way to accomplish the desired motion. We divide a gait motion cycle into 2 basic motions: single leg motion sequence and quadrupedal motion sequence. Since a gait is executed by the motion of the four legs in a specific pattern, let it be called the external sequence. The external sequence generator thereby decides when each leg has to move, and the time for which the leg is in contact or not with the ground. Once the time schedule for each leg's motion is available, the internal sequence generator takes the lead and generates each leg individual motion by controlling the legs joints. The internal sequence generator gets activated through the external sequence generator from the leg motion time schedule. Algorithm 1 shows the algorithm proposed to achieve a quadrupedal gait using the 2D polynomial functions G(i, t) and P(i,l).

Algorithm 1: Multi-gaits step sequence generation algorithm using Horner's polynomials

```
Require: maxSpeed, angle, speed
Ensure: gaitIndex

Analyse input
step 1. for i=1 to 6 do
step 2. if ((i-1) *maxSpeed/6≤ speed ≤ i*maxSpeed/6), then
Step 3. gaitIndex = i
step 4. end if
step 5. end for

//Generation of original sequence array OSA
step 6. supportDuration = 0
step 7.  for t=1 to 8 do
step 8.  OSA[i]= round(G(index, t),0)
step 9. if (OSA[i] == 1) then
step 10. supportDuration ++
step 11. end if
step 12. end for

//Generation of phase shift coefficients PSC
step 13: for l=1 to 4 do
step 14: LES[i]= circshift(OSA, PSC[i])
step 15: end for
step 16: disp(LES)
//Change bit detection, triggering internal step loop
step 17. for l=1 to 4 do
step 18. for t=1 to 8 do
step 19. curLegStates[l]= OSA[t]
step 20: if (prevLegState [l] == curLegState[l]) then
step 21. fintStep(1, supportDuration)
step 22. end if
step 23. prevLegState[1]=curLegStates[1]
```

```
step 24. end for
step 25. end for
```

Initially, the user is required to input certain crucial parameters required for further calculations. The main parameter to be supplied is the speed of the motion to be executed. Based on the speed, a calculation is attempted to detect the appropriate gait for such speed, based on the relationship that exists between gaits and speed. Once the speed range is identified, a gait index from 1 to 6 is assigned to the gait. Further calculations are performed only for the identified gait, as the algorithm has the ability to generate multi-gait signals using (1) and (2). The step sequence and the phase shift for the gait are generated and combined through a circular shift operation to obtain four different signals for each leg. These operations are carried out within a loop that checks continuously for a state change in the legs' signals. In case a state change is detected for a specific leg, the internal step function is triggered to allow the leg's motion for a set period that is defined during the generation of original sequence array. The particularity of this algorithm is the fact that it updates the output signal with respect to the speed and keeps on generating the same signal forever.

3. RESULTS AND DISCUSSIONS

The simulation aims to generate multi-gaits signal using Horner's polynomials. The first step in the simulation process is the generation of the original sequence polynomial function and phase shift coefficients using the multi-gaits step sequence generation algorithm. The algorithm is implemented through MATLAB software and generates each polynomial's coefficients (Table 9). It is vital to notice that a minimum of 8 decimals are required to have sufficient accuracy out of the polynomial, when the polynomial's degree is 1. Table 9 shows the polynomials' coefficients, the average error, and constant correction factor values obtained from the algorithm for both original sequence generation and phase shift coefficients.

Table 9. Multi-gaits Horner's polynomials generated coefficients

Coefficients	a_1	a_2	a_3	a_4	E_{avg}	b	C
G(i, t)	0.0002	-0.0159	-0.0175	0.5911	0.4688	0	0.4688
P(I, l)	0.0023	0.0225	0.0489	2.1742	2.0150	0.7	2.7150

Figure 9 shows the generated original sequence signal (in red color) for each gait against the target signal (in green color). In contrast, Figure 10 shows the generated phase shift signal (in red color) for each gait against the target signal (in green color). For both equations, the resultant graphs show a mismatch between the target and generated values, resulting in a significant error. The average error 'E_{avg}' for the defined working range for both equations is calculated and added to optimize the equations and thereby minimize the generated error (Table 9). The original sequence generation function accuracy increases considerably after adding the average error to the equations (Figure 11). In contrast, a constant coefficient 'b' must be added to the average error to minimize the error generated by the phase shift generation equation. Figure 12 shows a mismatch between both generated (green color) and target (blue color) signals, even after attempting to mitigate the generated error by adding the average error to the equation.

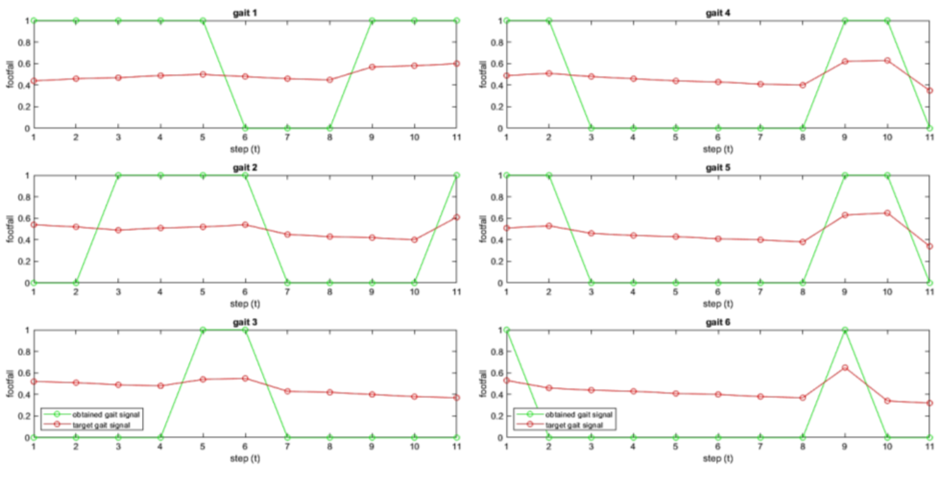

Figure 9. Multi-gaits original sequence signal generation through Horner's polynomials against targeted signals

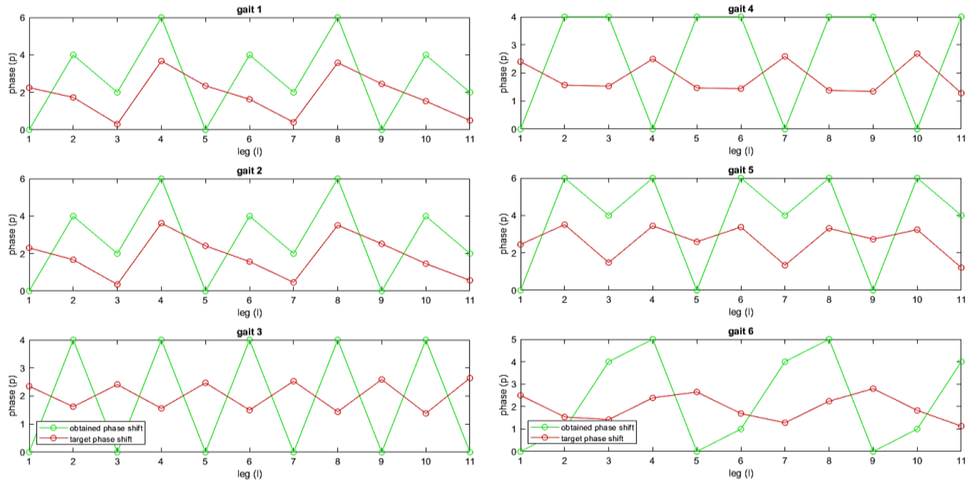

Figure 10. Multi-gaits/multi-legs phase shift coefficients generation through Horner's polynomials against targeted signals

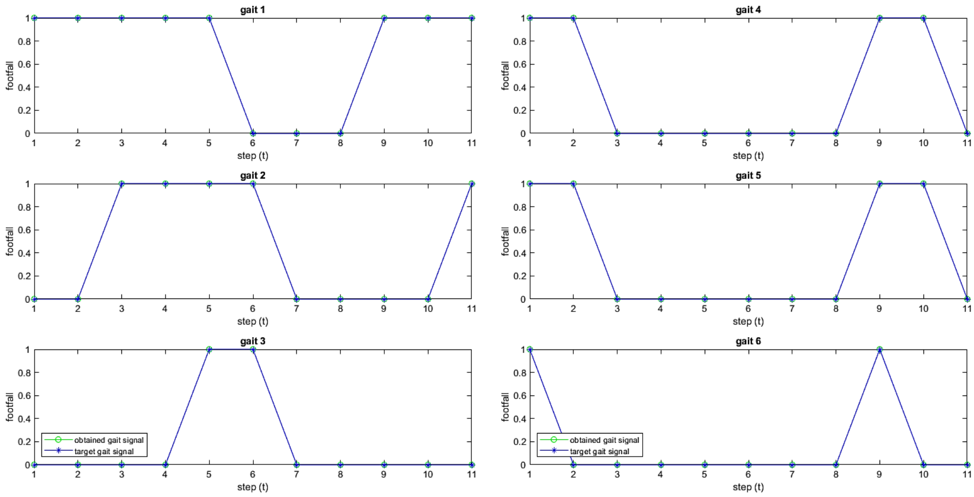

Figure 11. Multi-gaits original sequence signal generation through optimizes Horner's polynomials against targeted signals

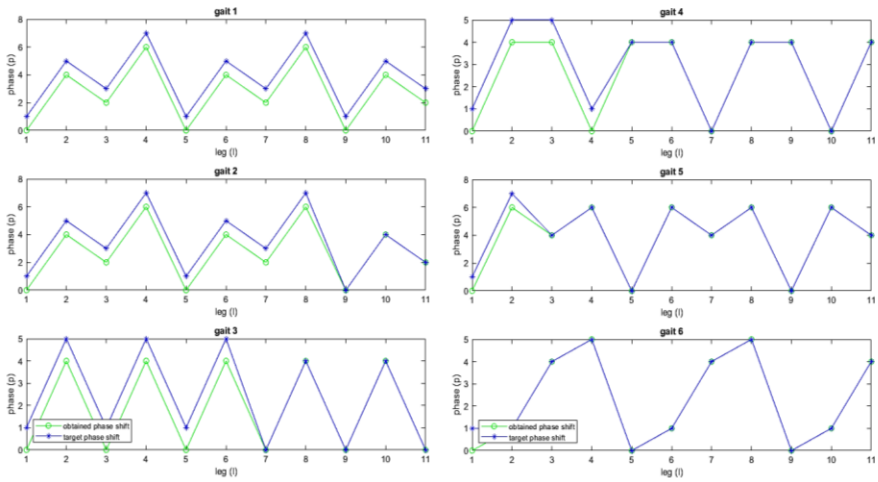

Figure 12. Multi-gaits/multi-legs phase shift coefficients generation through non-optimized Horner's polynomials against targeted signals

Figure 13 shows an improvement in the accuracy, as a constant coefficient ‘b’ is added to the average error value, resulting in an overall correction coefficient ‘C’ as shown in the updated (1) and (2) as follows:

$$G(i,t) = 0.0002 * i * t - 0.0159 * t - 0.0175 * i + 0.5911 + 0.4688$$

$$P(i,l) = 0.0023 * i * l + 0.0225 * l + 0.0489 * i + 2.1742 + 2.7150$$

A comparison between generated and target signals is made from the graphs, which shows the same signal for both generated and target signals, proving that the combination of both equations can generate multi-gaits signals with a single parameter: the gait index. Both equations are combined through an algorithm to generate each leg’s signal for a specific gait once the desired speed is provided. Figure 14 shows the four legs’ signals obtained from (1) and (2) for a quadruped walking gait. The proposed approach relies on the correlation between speed and the actual gait to assign an index that acts as an encoding for generating a specific gait. Such an approach is advantageous as it lowers the complexity of gait signal generation and reduces it to a single parameter, allowing it to generate various gaits signals with a single parameter. A qualitative comparison showing the possible advantages of employing a Horner's polynomial-based oscillator over a Hopf oscillator in the context of creating quadrupedal signals is given in Table 10.

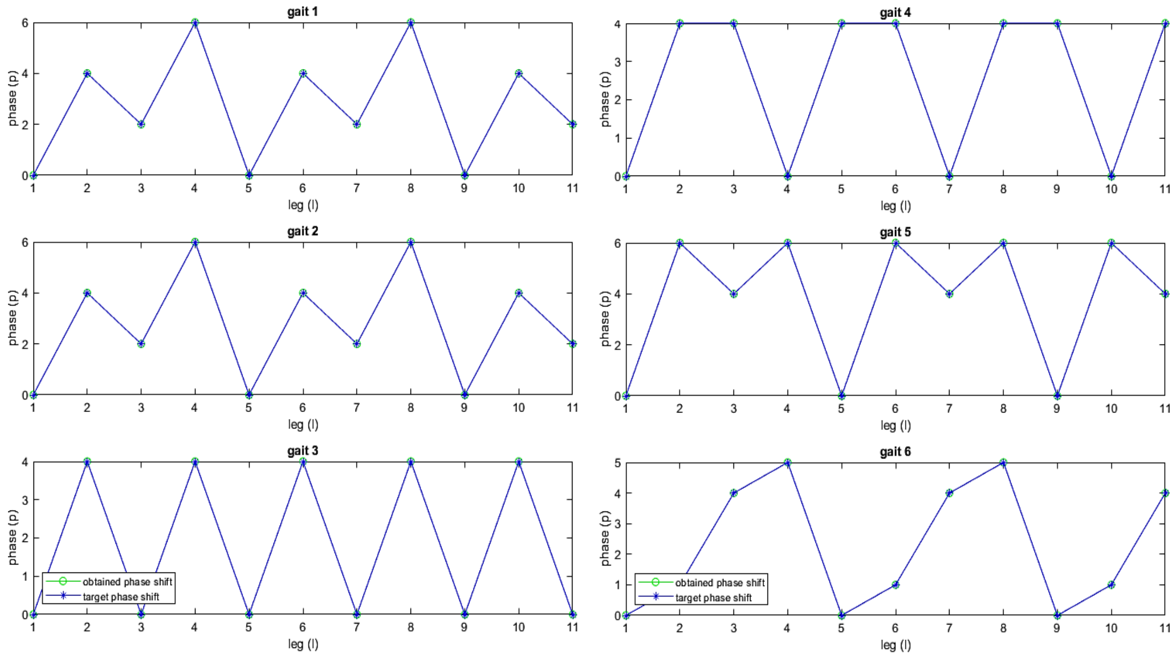

Figure 13. Multi-gaits/multi-legs phase shift coefficients generation through optimized Horner’s polynomials against targeted signals

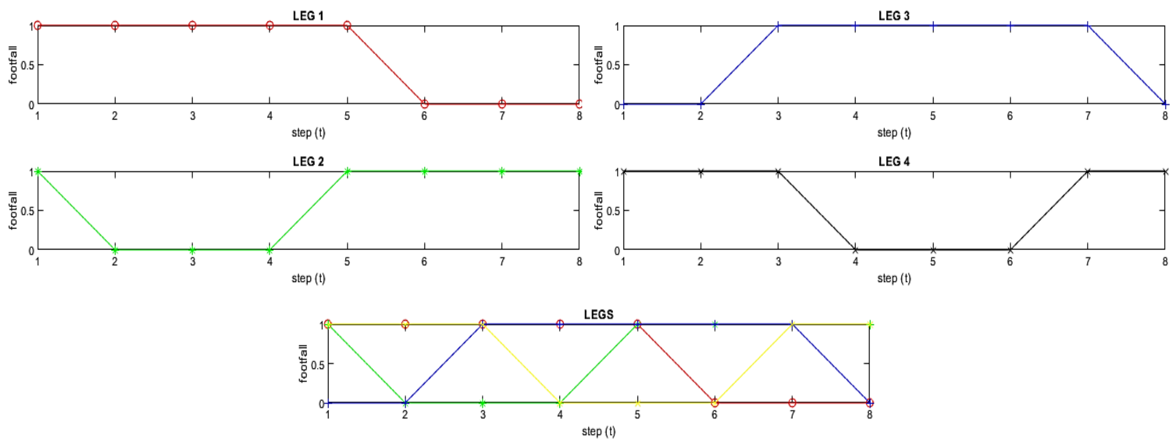

Figure 14. Quadruped walking gait signal generated through Horner’s polynomials

Table 10. Comparison between Horner's polynomial-based oscillators and Hopf oscillators

	Horner's polynomial-based oscillators	Hopf oscillators
Overall control	Allows for more fine-grained control of each limb's trajectory	Limited control over a precise gait pattern
Trajectory planning	Easier to handle specific locomotion requirements	Require additional computation for specific limb motion trajectories.
Terrain Handling	Easy adaptation to the environment by adjusting the polynomial coefficients	Need complex sensory feedbacks and mechanisms for terrain adaptation
Motion	Resulting in smooth movements with higher degrees polynomials	Generation of fixed frequencies that limits the possibility of gaits variations
Transitions	Ensure smooth transitions between different phases of the gait cycle	Requires additional control mechanisms for smooth transitions
Flexibility	Greater flexibility in designing the motion patterns	May lack the adaptability and real-time adjustment capabilities

4. CONCLUSION

The researchers have widely addressed the generation of suitable signals for the locomotion of quadruped robots. Famous approaches for solving such problems include the design of controllers through Hopf oscillators. This paper addresses the same challenge with a different approach based on Horner's method. A study is made about the locomotion behavior of quadruped animals, from which logic tables corresponding to different gaits sequences are established. A merged table for all the gaits sequences, involving a single leg sequence with a corresponding table for each leg's phase shift considering the first leg, is established. A gait-speed relationship is established, leading to an index assigned to each gait for further equations designs. Two is for phase difference and oscillation frequency are designed and optimized using Horner's polynomials. A simulation in MATLAB software showed that the designed models could generate expected signals with minimal error when optimization and filtering are applied conveniently. Implementing the designed model through a dedicated algorithm could generate accurate quadruped gaits' signals, when provided with the desired motion speed or gait index. When it comes to generating quadrupedal signals, Horner's polynomial-based oscillators and Hopf oscillators serve different purposes and have unique advantages and disadvantages. Hopf oscillators are used to generate stable periodic oscillations, whereas Horner's polynomial-based oscillators are used to generate piecewise polynomial functions. The proposed approach lowers the complexity of quadruped gait signal generation for various gaits by reducing the number of parameters required in the process. A related advantage is the quasi-automation of gait selection for a desired locomotion speed, as a relationship exists between the type of gait and maximum achievable speed. As a future work, a software simulation involving a quadruped robot will be done, along with studies on gaits transition and active gaits generation through machine learning.

REFERENCES

[1] C. Jing, "Application of variable zero-moment point in walking control of the biped robot," *Sains Malaysiana*, vol. 48, no. 12, pp. 2759–2766, 2019, doi: 10.17576/jsm-2019-4812-17.
[2] A. I. Haldar and N. D. Pagar, "Predictive control of zero moment point (ZMP) for terrain robot kinematics," *Materials Today: Proceedings*, vol. 80, pp. 122–127, 2023, doi: 10.1016/j.matpr.2022.10.286.
[3] L. H. Juang, "Humanoid robot runs up-down stairs using zero-moment with supporting polygons control," *Multimedia Tools and Applications*, vol. 82, no. 9, pp. 13275–13305, Sep. 2023, doi: 10.1007/s11042-022-13723-0.
[4] M. A. Elhosseini, A. Y. Haikal, M. Badawy, and N. Khashan, "Biped robot stability based on an A–C parametric whale optimization algorithm," *Journal of Computational Science*, vol. 31, pp. 17–32, 2019, doi: 10.1016/j.jocs.2018.12.005.
[5] H. Yu, H. Gao, and Z. Deng, "Enhancing adaptability with local reactive behaviors for hexapod walking robot via sensory feedback integrated central pattern generator," *Robotics and Autonomous Systems*, vol. 124, 2020, doi: 10.1016/j.robot.2019.103401.
[6] K. Takeda and H. Torikai, "A novel hardware-efficient central pattern generator model based on asynchronous cellular automaton dynamics for controlling hexapod robot," *IEEE Access*, vol. 8, pp. 139609–139624, 2020, doi: 10.1109/ACCESS.2020.3012706.
[7] S. Grillner and A. El Manira, "Current principles of motor control, with special reference to vertebrate locomotion," *Physiological Reviews*, vol. 100, no. 1, pp. 271–320, 2020, doi: 10.1152/physrev.00015.2019.
[8] B. A. Sauerbrei *et al.*, "Cortical pattern generation during dexterous movement is input-driven," *Nature*, vol. 577, no. 7790, pp. 386–391, 2020, doi: 10.1038/s41586-019-1869-9.
[9] K. Nakada, T. Asai, and Y. Amemiya, "An analog CMOS central pattern generator for interlimb coordination in quadruped locomotion," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1356–1365, Sep. 2003, doi: 10.1109/TNN.2003.816381.
[10] Z. Bhatti, "Oscillator driven central pattern generator (CPG) system for procedural animation of quadruped locomotion," *Multimedia Tools and Applications*, vol. 78, no. 21, pp. 30485–30502, 2019, doi: 10.1007/s11042-019-7641-1.
[11] Y. Habu, K. Uta, and Y. Fukuoka, "Three-dimensional walking of a simulated muscle-driven quadruped robot with neuromorphic two-level central pattern generators," *International Journal of Advanced Robotic Systems*, vol. 16, no. 6, Nov. 2019, doi: 10.1177/1729881419885288.
[12] G. Bellegarda and A. Ijspeert, "CPG-RL: learning central pattern generators for quadruped locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12547–12554, 2022, doi: 10.1109/LRA.2022.3218167.
[13] A. I. Guillaume Bellegarda, "Visual CPG-RL: learning central pattern generators for visually-guided quadruped navigation," *arXiv-Computer Science*, pp. 1-8, 2022.
[14] Y. Kim, B. Son, and D. Lee, "Learning multiple gaits of quadruped robot using hierarchical reinforcement learning," *arXiv-*

Computer Science, pp. 1-9, 2021.
[15] J. Wang, C. Hu, and Y. Zhu, "CPG-based hierarchical locomotion control for modular quadrupedal robots using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7193–7200, 2021, doi: 10.1109/LRA.2021.3092647.
[16] A. S. Lele, Y. Fang, J. Ting, and A. Raychowdhury, "Learning to walk: spike based reinforcement learning for hexapod robot central pattern generation," in *Proceedings - 2020 IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2020*, IEEE, 2020, pp. 208–212, doi: 10.1109/AICAS48895.2020.9073987.
[17] O. K. Adak and K. Erbatur, "Bound gait reference generation of a quadruped robot via contact force planning," *International Journal of Mechanical Engineering and Robotics Research*, vol. 11, no. 3, pp. 129–137, 2022, doi: 10.18178/ijmerr.11.3.129-137.
[18] H. W. Park, S. Park, and S. Kim, "Variable-speed quadrupedal bounding using impulse planning: untethered high-speed 3D running of MIT cheetah 2," in *Proceedings - IEEE International Conference on Robotics and Automation*, IEEE, 2015, pp. 5163–5170, doi: 10.1109/ICRA.2015.7139918.
[19] M. Hutter *et al.*, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *IEEE International Conference on Intelligent Robots and Systems*, IEEE, 2016, pp. 38–44, doi: 10.1109/IROS.2016.7758092.
[20] Y. Fujimoto and A. Kawamura, "Simulation of an autonomous biped walking robot including environmental force interaction," *IEEE Robotics and Automation Magazine*, vol. 5, no. 2, pp. 33–42, 1998, doi: 10.1109/100.692339.
[21] K. Erbatur *et al.*, "SURALP: a new full-body humanoid robot platform," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, IEEE, 2009, pp. 4949–4954, doi: 10.1109/IROS.2009.5354276.
[22] Y. Wang, L. Ye, X. Wang, N. Cheng, H. Liu, and B. Liang, "A static gait generation for quadruped robots with optimized walking speed," in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, IEEE, 2020, pp. 1899–1906, doi: 10.1109/SMC42975.2020.9282997.
[23] W. L. Ma and A. D. Ames, "From bipedal walking to quadrupedal locomotion: full-body dynamics decomposition for rapid gait generation," in *IEEE International Conference on Robotics and Automation*, IEEE, 2020, pp. 4491–4497, doi: 10.1109/ICRA40945.2020.9196841.
[24] T. Fukui, H. Fujisawa, K. Otaka, and Y. Fukuoka, "Autonomous gait transition and galloping over unperceived obstacles of a quadruped robot with CPG modulated by vestibular feedback," *Robotics and Autonomous Systems*, vol. 111, pp. 1–19, 2019, doi: 10.1016/j.robot.2018.10.002.
[25] Y. Takei, K. Morishita, R. Tazawa, K. Katsuya, and K. Saito, "Non-programmed gait generation of quadruped robot using pulse-type hardware neuron models," *Artificial Life and Robotics*, vol. 26, no. 1, pp. 109–115, Sep. 2021, doi: 10.1007/s10015-020-00637-z.
[26] F. Zhuo, W. Jia, S. Ma, J. Yuan, and Y. Sun, "A variable artificial potential field method for gait generation of quadruped robot," in *2021 IEEE International Conference on Mechatronics and Automation, ICMA 2021*, IEEE, 2021, pp. 1176–1181, doi: 10.1109/ICMA52036.2021.9512753.
[27] Z. Ma, Y. Liang, and H. Tian, "Research on gait planning algorithm of quadruped robot based on central pattern generator," in *Chinese Control Conference, CCC*, IEEE, 2020, pp. 3948–3953, doi: 10.23919/CCC50068.2020.9188942.
[28] Y. Shao, Y. Jin, X. Liu, W. He, H. Wang, and W. Yang, "Learning free gait transition for quadruped robots via phase-guided controller," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1230–1237, Apr. 2022, doi: 10.1109/LRA.2021.3136645.
[29] L. Bai, H. Hu, X. Chen, Y. Sun, C. Ma, and Y. Zhong, "CPG-based gait generation of the curved-leg hexapod robot with smooth gait transition," *Sensors*, vol. 19, no. 17, 2019, doi: 10.3390/s19173705.
[30] H. Zhang, J. Yin, and H. Wang, "A needs learning algorithm applied to stable gait generation of quadruped robot," *Sensors*, vol. 22, no. 19, Sep. 2022, doi: 10.3390/s22197302.
[31] T. Atkinson, *Practical feline behaviour: understanding cat behaviour and improving welfare*. Wallingford, UK: CAB International, 2018, doi: 10.1079/9781780647838.0000.
[32] B. V. Beaver, "Canine grooming behavior," in *Canine Behavior*, Elsevier, 2009, pp. 285–293, doi: 10.1016/b978-1-4160-5419-1.00010-9.
[33] B. V. Beaver, "Feline grooming behavior," in *Feline Behavior*, Elsevier, 2003, pp. 311–321, doi: 10.1016/b0-72-169498-5/50011-3.

BIOGRAPHIES OF AUTHORS

Kouame Yann Olivier Akansie obtained the B.E. (2018) degree in Electrical and Electronics Engineering from East Point College of Engineering and Technology, India. He further obtained the M.Tech. (2020) degree in Digital Communications and Networking from REVA University, India. He is currently a Ph.D. scholar at REVA University, India, within the school of Electronics and Communication Engineering. His thesis is about the design and development of autonomous hybrid wheel-legged robot for terrestrial navigation. He can be contacted at email: akansiekouame@gmail.com.

Rajashekhar C. Biradar completed his B.E. (Electronics and Communication Engineering) in 1990, M.E. (Digital Electronics) in 1997 from Karnataka University Dharwad, India, and Ph.D. in 2011 under Visvesvaraya Technological University (VTU), Belgaum, India. He is currently working as Professor and Director of School of ECE, REVA university, Bangalore, India. To his credit, he has many national/international journal and conference publications. His research areas include multicast routing in mobile ad hoc networks, wireless internet, group communication in MANETs, and agent technology. He is a member of IEEE (USA), member of IETE (MIETE, India), member of ISTE (MISTE, India), member of IE (MIE, India) and member of ACM. He can be contacted at email: rcbiradar@reva.edu.in.

Karthik Rajendra received M.Tech. degree from the Visvesvaraya Technological University, India and Ph.D. degree from VIT University, India. His Ph.D. thesis research work was carried out at one of the Labs of Center for Nano electronics, Indian Institute of Technology – Bombay, India. He was the Professor at Department of Electronics and Communication and Dean R&D at MLR Institute of Technology, Hyderabad. Earlier, he was working as a faculty member at VIT University, Vellore. He received best researcher award from VIT University for his contribution to Nano dielectrics in 2013 and 2014. Also, he received Best researcher award in 2017, 2018, and 2019 at MLR Institute of Technology. His current area of research includes fabrication and modeling of nano electronic or optoelectronic material-based devices, microwave antennas, medical image processing, and transformation in engineering education. At present, he is guiding 1 Ph.D. research scholar. He has completed 5 sponsored research projects worth Rs. 44 Lakhs. At present he has 2 ongoing research projects funded by DST, Govt. of India worth Rs. 3 Crores. He has published more than 110 research papers in reputed journals and conferences, 4 book chapters and filed 3 patents. He is one of the co-designers for developing a nano-size high performance capacitor in 2013 and 2020. He can be contacted at email: karthik.rajendra@reva.edu.in.

Geetha D. Devanagavi received her Ph.D., M.Tech., and B.E. degrees in 2014, 2005, and 1993, respectively. She is currently working as associate Professor in the school of Computing and Information Technology at Reva University, Bangalore, India. She has 24 years of teaching experience. Her research interests include wireless sensor networks, network security, and computer networks. She can be contacted at email: dgeetha@reva.edu.in