



# Design of a Stable and Intelligent Controller for a Quadruped Robot

Ammar A. Aldair<sup>1</sup> · Auday Al-Mayyahi<sup>1</sup> · Weiji Wang<sup>2</sup>

Received: 22 July 2019 / Revised: 29 October 2019 / Accepted: 10 December 2019 / Published online: 20 December 2019  
© The Korean Institute of Electrical Engineers 2019

## Abstract

Quadruped robots have increasingly been used in complex terrains where barriers and gaps exist. In this paper, a four-legged robot with intelligent controllers is designed and simulated. The designed architecture comprises 12 servo motors, three per leg, to provide considerable flexibility in movement and turning. Proportional Integral Derivative (PID) controllers and Fuzzy controllers are proposed to control and stabilize the motion of the quadruped robot. An ant colony optimization algorithm has been utilized to tune the parameters of the PID controller and the Fuzzy controller. After obtaining the optimal values of both controllers, the entire architecture is implemented using the Multibody Simscape package in MATLAB which models multidomain physical systems. The simulation results are conducted in a 3-dimensional environment and they are demonstrated in three case studies; firstly, when the system is simulated without using a controller which leads to a collapse of the quadruped robot. Secondly, when the PID controller is combined with the system, better movement is obtained. However, the quadruped is unable to complete its path and collapses after a few meters. Thirdly, when the Fuzzy controller is integrated into the designed architecture, a significant improvement is observed in terms of minimizing elapsed time and improving the overall performance of the motion. The stability of the Fuzzy controller has been examined using Lyapunov criteria to validate its overall performance. Comparisons are conducted based on control efforts and travelled distances to demonstrate the suitability and effectiveness of Fuzzy controllers over PID controllers.

**Keywords** Quadruped robots · PID controller · Fuzzy inference system · Ant colony optimization · Lyapunov stability

## 1 Introduction

Robots have been developed widely in the last two decades. Starting from a simple mobile robot that can be easily driven by using two motorized wheels [1] to new structures where four wheels are utilized to drive the motion of a mobile robot [2]. When four motorized wheels are in use, four controllers are needed at least to control the movement of such mobile robots.

On sloppy, rough or irregular terrains, wheeled robots may lose their ability to move whereas, legged robots have the ability to move on complex types of terrains [3, 4]. In some tasks such as climbing upstairs or crossing over ditches, legged robots are more suitable to perform those types of missions rather than wheeled robots. Therefore, significant efforts have been devoted to designing legged robots that imitate a gait of animals such as horses or dogs [5–10].

From the early 70 s of the last century, researchers simulated a gait of human beings. The authors in Refs. [11, 12] used the inverted pendulum structure to study some characteristics of human motion. The authors in Ref. [13] suggested a three-link legged robot structure to simulate human gaits and to study more characteristics of human locomotion.

In recent years, more complex architectures of robotics systems have been introduced based on quadruped robots [14–18]. This type of robotics can be applied in new applications that require complex motions where quadruped robots can perform jumping and traverse barriers. In other words, it can exhibit feasible performance over irregular terrains. As illustrated in the literature, the most utilized types are two

---

✉ Ammar A. Aldair  
mmr.ali2@googlemail.com

Auday Al-Mayyahi  
auday.almayyahi@hotmail.com

Weiji Wang  
W.J.Wang@sussex.ac.uk

<sup>1</sup> Department of Electrical Engineering, College of Engineering, University of Basrah, Basrah, Iraq

<sup>2</sup> Department of Engineering and Design, School of Engineering and Informatics, University of Sussex, Brighton, UK

links per leg. In complex applications, quadruped robots are mostly used instead of human beings such as military applications, space exploration, dangerous sites, harmful sites, and repeated tasks.

Quadruped robots demonstrate great advantages in the movement which can reach a high degree of freedom. Uncertainties and high nonlinearity inherently exist in the mathematical model of quadruped robots; therefore, the design of stable and robust controllers for the quadruped robot is a considerable challenge. In the state of the art, researchers have conducted different approaches to control the motion of quadruped robots. In Ref. [19], a controller is designed using slide mode to drive a quadruped robot. In that method, tracking error was improved regardless of the presence of disturbances. In Ref. [20], a novel control scheme was presented for quadruped locomotion control. The control scheme was based on a proportional-integral (PI) controller and the Jacobian to drive legs' joints based on applied torques. Another controller was proposed based on neural networks [21] in which fuzzy neural networks were applied to control joints' motion to follow the desired trajectory. In addition, a robust controller [22] was designed to control and stabilize a legged robot system. The robustness was investigated based on parametric uncertainty for a constant bound and unstructured bound of a position-velocity.

The authors in Ref. [23] presented a novel approach for controlling the hopping height of a monopod instantaneously, i.e., during the flight phase, the controller computes the required action to achieve the demanded height of the next step. In Ref. [24], the authors presented a learning approach to quadruped robot galloping control. The control function is obtained through directly approximating real gait data by a learning algorithm, without consideration of the robot's model and the environment where the robot is located.

Although some attempts were provided in the survey to control the motion of legged robots, those controllers were only applied for two links per leg. Additionally, the modeling analysis was mainly dedicated to the same rather than three links per leg. Therefore, in this paper, we extend the scope of work to three links per leg and this can exhibit a higher degree of freedom and flexibility while moving. However, such an extension will increase the complexity of designing a robust controller in which 12 controllers will be required. Three controllers are applied per leg to control the servo motors over the three links per leg. The designed controllers are firstly based on a proportional, integral and derivative (PID) controllers and secondly, based on a Fuzzy controller. The novelty of this paper can be understood by designing a stable and robust controller for quadruped robots where uncertainties and high nonlinearity inherently exist in the mathematical model of a quadruped robot. Moreover, an ACO algorithm is used to find the optimal parameters of the 12 PID controller and the 12 Fuzzy controllers. Furthermore, the mathematical model

has been analyzed for three links per leg quadruped robot which new modelling has been proposed in this work. When the proposed Fuzzy controllers are compared with the PID controllers, it has been observed significant improvements in terms of the stability and time response.

The rest of the paper is organized as follows: The modelling of a quadruped robot is described in Sect. 2. The explanation of the control system design is detailed in Sect. 3. Particle swarm optimization is demonstrated in Sect. 4. The Lyapunov stability test for the Fuzzy system has been performed in Sect. 4. The simulation results are conducted and analyzed in Sect. 5. In Sect. 6, comparisons have been provided and discussed. Finally, the conclusion has been highlighted in Sect. 7.

## 2 Modelling of a Quadruped Robot

A quadruped robot with multiple degrees of freedom will exhibit highly nonlinear behavior. The modelling of a quadruped robot can be derived based on both kinematic and dynamic characteristics [25–27]. Figure 1 below demonstrates the 3D structure of the quadruped robot. As shown, there are four legs, in any of which, three links are given to provide a high range of flexibility during the motion. This structure is implemented using the Multibody Simscape package. The front and rear of the right-handed side legs are utilized in deriving the mathematical models of kinematic and dynamic characteristics that govern the motion of the quadruped robot in a given workspace. The other two legs behave similarly to the front and the rear of the right-handed side legs.

The inspiration to derive the mathematic model is based on two links per leg which extend to three links per leg taking into consideration a new structure of the movement.

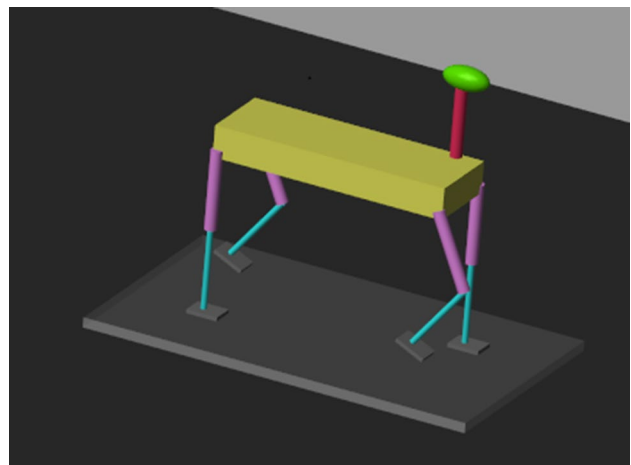


Fig. 1 The 3D structure of the quadruped robot

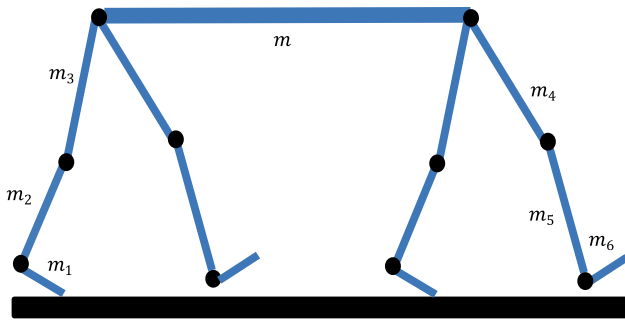


Fig. 2 Systematic diagram of the quadruped robot

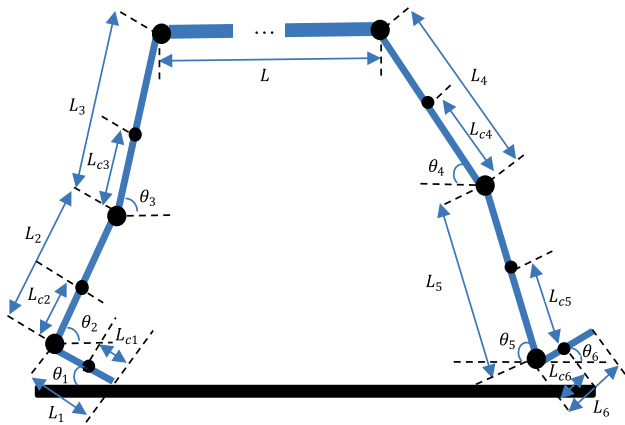


Fig. 3 The three kinematic analysis of the quadruped robot

The systematic diagram of the quadruped robot is depicted in Fig. 2 which also shows the distribution of masses the elements that comprise the body of the quadruped robot. The front and the rear of right-handed side legs are shown in Fig. 3 which illustrates the kinematic analysis of the body’s links. These two legs will be only used in deriving the kinematic and dynamic models as the other two legs are likewise. The parameters utilized in this structure are identified as follows:

- $m$  : mass of the body of a quadruped robot.
- $L$  : length of the body of the quadruped robot.
- $m_i$  : mass of  $i$ th link.
- $L_i$  : length of  $i$ th link.
- $L_{ci}$  : the distance between the center of mass and the lower joint of  $i$ th link.
- $\theta_i$  : angle of  $i$ th link with respect to the horizontal axis.
- $(x_{ci}, y_{ci})$  : are the coordinates of the center of mass of  $i$ th link.

The modelling of the quadruped robot can be analyzed based on both kinematic and dynamic characteristics as follows:

(a) Kinematic Model

This section deals with the derivation of the center positions of each link of the two legs shown in Fig. 3 with respect to the ground plane. For the first three links of the rear right-handed side, the center positions can be determined using the following equations:

$$x_{c1} = -L_{c1} \cos \theta_1 \tag{1}$$

$$y_{c1} = L_{c1} \sin \theta_1 \tag{2}$$

$$x_{c2} = -L_1 \cos \theta_1 + L_{c2} \cos \theta_2 \tag{3}$$

$$y_{c2} = L_1 \sin \theta_1 + L_{c2} \sin \theta_2 \tag{4}$$

$$x_{c3} = -L_1 \cos \theta_1 + L_2 \cos \theta_2 + L_{c3} \cos \theta_3 \tag{5}$$

$$y_{c3} = L_1 \sin \theta_1 + L_2 \sin \theta_2 + L_{c3} \sin \theta_3 \tag{6}$$

Similarly, for the first three links of the rear left-handed side, the center positions can be determined using the following equations:

$$x_{c4} = -L_1 \cos \theta_1 + L_2 \cos \theta_2 + L_3 \cos \theta_3 + L + L_4 \cos \theta_4 - L_{c4} \cos \theta_4 \tag{7}$$

$$y_{c4} = L_1 \sin \theta_1 + L_2 \sin \theta_2 + L_3 \sin \theta_3 - L_4 \sin \theta_4 + L_{c4} \sin \theta_4 \tag{8}$$

$$x_{c5} = -L_1 \cos \theta_1 + L_2 \cos \theta_2 + L_3 \cos \theta_3 + L + L_4 \cos \theta_4 + L_5 \cos \theta_5 - L_{c5} \cos \theta_5 \tag{9}$$

$$y_{c5} = L_1 \sin \theta_1 + L_2 \sin \theta_2 + L_3 \sin \theta_3 - L_4 \sin \theta_4 - L_5 \sin \theta_5 + L_{c5} \sin \theta_5 \tag{10}$$

$$x_{c6} = -L_1 \cos \theta_1 + L_2 \cos \theta_2 + L_3 \cos \theta_3 + L + L_4 \cos \theta_4 + L_5 \cos \theta_5 - L_{c6} \cos \theta_6 \tag{11}$$

$$y_{c6} = L_1 \sin \theta_1 + L_2 \sin \theta_2 + L_3 \sin \theta_3 - L_4 \sin \theta_4 - L_5 \sin \theta_5 + L_{c6} \sin \theta_6 \tag{12}$$

(b) Dynamic Model

The dynamic model of a quadruped robot can be analyzed based on its overall structure which is composed of different bodies, joints, and links. The elements if the quadruped robot’s body are acting based on applied forces in addition to internal and external characteristics such as the moment of inertia, friction, gravity and body’s constraints. In this paper, the Lagrange dynamic equation is utilized to model the motion of a quadruped robot as in the following equation:

$$D(\theta)\ddot{\theta} + H(\theta, \dot{\theta})\dot{\theta} + G(\theta) = T \tag{13}$$

where  $\theta = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6]$  is the vector angle of  $i$ th link with respect to the vertical ground plane.

$D(\theta) = [d_{ij}(\theta)](i, j = 1, \dots, 6)$  is the inertia matrix;  $H(\theta, \dot{\theta}) = \text{col} \left[ \sum_{j=1(j \neq i)}^6 (h_{ij}(\dot{\theta}_j)^2) \right]$  is the centrifugal matrix;  $G(\theta) = \text{col}[G_i(\theta)]$  is the gravity vector;  $T = [T_1 \ T_2 \ T_3 \ T_4 \ T_5 \ T_6]^T$  is the vector of external torque applied at joints.

The  $D(\theta)$  is a  $6 \times 6$  inertia matrix and can be given as follows:

$$D(\theta) = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} \\ d_{41} & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} \\ d_{61} & d_{62} & d_{63} & d_{64} & d_{65} & d_{66} \end{bmatrix} \quad (14)$$

The parameters of the inertia matrix  $D(\theta)$  can be determined as in below:

$$d_{11} = I_1 + m_1 L_{1c}^2 + (m_2 + m_3 + m_4 + m_5 + m_6 + m) L_1^2$$

$$d_{12} = p_{12} \cos(\theta_1 + \theta_2), \quad d_{13} = p_{13} \cos(\theta_1 + \theta_3)$$

$$d_{14} = p_{14} \cos(\theta_1 - \theta_4), \quad d_{15} = p_{15} \cos(\theta_1 - \theta_5)$$

$$d_{16} = p_{16} \cos(\theta_1 + \theta_6), \quad d_{21} = d_{12},$$

$$d_{22} = I_2 + m_2 L_{2c}^2 + (m_3 + m_4 + m_5 + m_6 + m) L_2^2$$

$$d_{23} = p_{23} \cos(\theta_2 - \theta_3), \quad d_{24} = p_{24} \cos(\theta_2 + \theta_4)$$

$$d_{25} = p_{25} \cos(\theta_2 + \theta_5), \quad d_{26} = p_{26} \cos(\theta_2 - \theta_6)$$

$$d_{31} = d_{13}, \quad d_{32} = d_{23},$$

$$d_{33} = I_3 + m_3 L_{3c}^2 + (m_4 + m_5 + m_6 + m) L_3^2$$

$$d_{34} = p_{34} \cos(\theta_3 + \theta_4), \quad d_{35} = p_{35} \cos(\theta_3 + \theta_5)$$

$$d_{36} = p_{36} \cos(\theta_3 - \theta_6)$$

$$d_{41} = d_{14}, \quad d_{42} = d_{24}, \quad d_{43} = d_{34}$$

$$d_{44} = I_4 + m_4 L_{4c}^2 - 2m_4 L_4 L_{4c} + (m_5 + m_6) L_4^2$$

$$d_{45} = p_{45} \cos(\theta_4 - \theta_5), \quad d_{46} = p_{46} \cos(\theta_4 + \theta_6)$$

$$d_{51} = d_{15}, \quad d_{52} = d_{25}, \quad d_{53} = d_{35}, \quad d_{54} = d_{45},$$

$$d_{55} = I_5 + m_5 L_{5c}^2 - 2m_5 L_5 L_{5c} + (m_5 + m_6) L_5^2$$

$$d_{56} = p_{56} \cos(\theta_5 + \theta_6)$$

$$d_{61} = d_{16}, \quad d_{62} = d_{26}, \quad d_{63} = d_{36}, \quad d_{64} = d_{46}$$

$$d_{65} = d_{56}, \quad d_{66} = I_6 + m_6 L_{6c}^2$$

$I_i$  is the moment of inertia with respect to an axis that passes through the mass center of  $i$ th link and it is perpendicular to the motion plane.

The centrifugal matrix  $H(\theta, \dot{\theta})$  is also a  $6 \times 6$  matrix and its parameters can be calculated as in the following equations:

$$H(\theta) = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} & h_{16} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} & h_{26} \\ h_{31} & h_{32} & h_{33} & h_{34} & h_{35} & h_{36} \\ h_{41} & h_{42} & h_{43} & h_{44} & h_{45} & h_{46} \\ h_{51} & h_{52} & h_{53} & h_{54} & h_{55} & h_{56} \\ h_{61} & h_{62} & h_{63} & h_{64} & h_{65} & h_{66} \end{bmatrix} \quad (15)$$

$$h_{11} = 0, \quad h_{12} = -p_{12} \sin(\theta_1 + \theta_2), \quad h_{13} = -p_{13} \sin(\theta_1 + \theta_3),$$

$$h_{14} = p_{14} \sin(\theta_1 - \theta_4), \quad h_{15} = p_{15} \sin(\theta_1 - \theta_5),$$

$$h_{16} = -p_{16} \sin(\theta_1 + \theta_6),$$

$$h_{21} = h_{12}, \quad h_{22} = 0, \quad h_{23} = p_{23} \sin(\theta_2 - \theta_3)$$

$$h_{24} = p_{24} \sin(\theta_2 + \theta_4), \quad h_{25} = p_{25} \sin(\theta_2 + \theta_5)$$

$$h_{26} = p_{26} \sin(\theta_2 - \theta_6)$$

$$h_{31} = h_{13}, \quad h_{32} = -h_{23}, \quad h_{33} = 0, \quad h_{34} = -p_{34} \sin(\theta_3 + \theta_4)$$

$$h_{35} = p_{35} \sin(\theta_3 - \theta_5), \quad h_{36} = p_{36} \sin(\theta_3 - \theta_6)$$

$$h_{41} = -h_{14}, \quad h_{42} = h_{24}, \quad h_{43} = h_{34}, \quad h_{44} = 0$$

$$h_{45} = -p_{45} \sin(\theta_4 - \theta_5), \quad h_{46} = -p_{46} \sin(\theta_4 + \theta_6)$$

$$h_{51} = -h_{15}, \quad h_{52} = h_{25}, \quad h_{53} = h_{35}, \quad h_{56} = -h_{45}, \quad h_{55} = 0$$

$$h_{56} = -p_{56}\sin(\theta_5 + \theta_6)$$

$$h_{61} = h_{16}, \quad h_{62} = -h_{26}, \quad h_{63} = -h_{36}, \quad h_{64} = h_{46},$$

$$h_{65} = h_{56}, \quad h_{66} = 0.$$

where

$$p_{12} = m_2L_{2c}L_1 + (m_3 + m_4 + m_5 + m_6 + m)L_1L_2$$

$$p_{13} = m_3L_1L_{3c} + (m_4 + m_5 + m_6 + m)L_1L_3$$

$$p_{14} = m_4L_1L_{4c} - (m_4 + m_5 + m_6)L_1L_4$$

$$p_{15} = m_5L_1L_{5c} - (m_5 + m_6)L_1L_5$$

$$p_{16} = m_6L_1L_{6c}$$

$$p_{23} = m_3L_2L_{3c} + (m_4 + m_5 + m_6 + m)L_2L_3$$

$$p_{24} = m_4L_2L_{4c} - (m_4 + m_5 + m_6)L_2L_4$$

$$p_{25} = m_5L_2L_{5c} - (m_5 + m_6)L_2L_5$$

$$p_{26} = m_6L_2L_{6c}$$

$$p_{34} = m_4L_3L_{4c} - (m_4 + m_5 + m_6)L_3L_4$$

$$p_{35} = m_5L_3L_{5c} - (m_5 + m_6)L_3L_5$$

$$p_{36} = m_6L_3L_{6c}$$

$$p_{45} = -m_5L_4L_{5c} + (m_5 + m_6)L_4L_5$$

$$p_{46} = -m_6L_4L_{6c}$$

$$p_{56} = -m_6L_5L_{6c}$$

The parameters of gravity vector  $G(\theta)$  can be computed as follows:

$$G_1 = [m_1L_{1c} + (m_2 + m_3 + m_4 + m_5 + m_6 + m)L_1]g\cos(\theta_1)$$

$$G_2 = [m_2L_{2c} + (m_3 + m_4 + m_5 + m_6 + m)L_2]g\cos(\theta_2)$$

$$G_3 = [m_3L_{3c} + (m_4 + m_5 + m_6 + m)L_3]g\cos(\theta_3)$$

$$G_4 = -[m_4(L_4 - L_{4c}) + (m_5 + m_6)L_4]g\cos(\theta_4)$$

$$G_5 = -[m_5(L_5 - L_{5c}) + m_6L_5]g\sin(\theta_5)$$

$$G_6 = (L_{6c} + L_6)m_6g\sin(\theta_6)$$

$g$  is acceleration of gravity.

It is assumed that  $U = [u_1 \ u_2 \ u_3 \ u_4 \ u_5 \ u_6]$  is the vector of the driving torques of the six rotating joints of quadruped robot which are generated from the servo motors, thus, the following equation can be written as:

$$T = SU^T \quad (16)$$

where  $S \in \mathbb{R}^{6 \times 6}$  is a transformation matrix which can be given as:

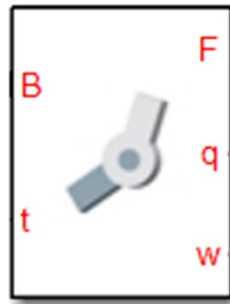
$$S = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

### 3 Control System Design

In this section, control systems are designed and implemented as will be discussed in the following subsections. These two controlling systems are the PID controller and the Fuzzy logic controller. In the designed system, there are four legs in which each leg has three links. Therefore, there will be twelve links in total. Each link is driven from a separate servo motor. The structure of each group of links is identical to others. Hence, we will only introduce the control system for one link and the others are exactly the same architecture. The block diagram of each link has two inputs and three outputs. The inputs are the 'B' which represents the port that connects the current link with the previous one. The second input is 't' which represents the control signal supplied to the torque of a servo motor. The outputs are F, q and w, F is the connection point between the current link and the next one. The q and w are the angle and speed of a current link, respectively. Figure 4 below shows the block diagram of a link in a leg.

Based on the operation of the DC servo motor, the speed and torque of the DC servo motor can be controlled by sending electrical pulses of variable width. For each servo motor, there is a minimum pulse and a maximum pulse and any of which has a repetition rate. The width of the sent electrical pulses changes the value of the applied voltage accordingly where the speed of a servo is dependent. Furthermore, as the voltage changes, the flowing current will be changed which leads to the torque being changed as well.

**Fig. 4** Block diagram of a link in a leg



Overall, there will be three block diagrams in our design for each leg, thus, twelve block diagrams in the entire system. Consequently, this requires twelve controllers to achieve feasible movement of a quadruped robot.

### 3.1 PID Controller

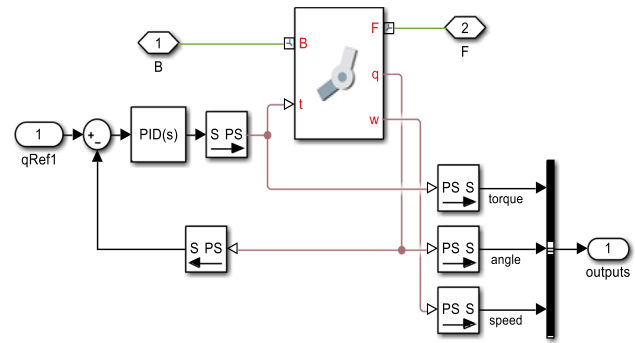
The PID controller is widely used in many industrial applications [28]. The structure of the PID controller consists of three parameters, i.e., proportional gain ( $k_p$ ), integral gain ( $k_i$ ) and derivative gain ( $k_d$ ), any of which can be tuned to obtain an optimal value based on specifications of a given system. The main equation that governs the operation of such a controller is illustrated below.

$$u_{PID}(t) = k_p e(t) + k_i \int e(t)dt + k_d \dot{e}(t) \tag{17}$$

where  $e(t)$  is an error signal between two signals, i.e., reference and actual values. The reference value is supplied signal based on the desired input of a plant and the actual value is the real output response of that plant.

The parameters of the PID controller play a crucial role in changing an overall performance and a time response of the system and it can significantly influence the operation conditions either positively or negatively. In order to reach the best response possible, the  $k_p$ ,  $k_i$  and  $k_d$  need to be tuned optimally. The architecture of a PID controller in connection with only one link in a leg is given in Fig. 5 below.

To obtain optimal performance of PID controllers, we need to optimize their parameters using optimization algorithms. Recently, many optimization algorithms are developed such as Genetic Algorithm (GA) [29], Particle Swarm Optimization (PSO) algorithm [30], Artificial Bee Colony (ABC) algorithm [31] and Ant Colony Optimization (ACO) [32, 33], any of which has its own structure and method in order to approach an optimal solution for a given optimization algorithm. The ACO algorithm has been developed as a meta-heuristic algorithm that can address high combinatorial optimization problems. It has unique advantages such as the construction graph used in the ACO algorithm that can make the search space represented less redundant. In addition, it has fewer parameters and strong robustness [34].



**Fig. 5** PID controller for one link in a leg

The principle operation of ACO is based on the behavior of ants which can be represented by individuals that can find a suitable solution for an optimization problem. The best solution can be determined globally by the cooperation of all ants together.

In this paper, the ACO algorithm will be utilized to tune the parameters of the twelve PID controllers in the offline mode. Therefore, a parameter matrix (PM) can be created which has three columns and  $N$  rows as shown below:

$$PM = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ \vdots & \vdots & \vdots \\ P_{N1} & P_{N2} & P_{N3} \end{bmatrix} = \begin{bmatrix} k_{p1} & k_{i1} & k_{d1} \\ k_{p2} & k_{i2} & k_{d2} \\ \vdots & \vdots & \vdots \\ k_{pN} & k_{iN} & k_{dN} \end{bmatrix} \tag{18}$$

where each row of  $PM$  matrix represents the candidates of  $k_p$ ,  $k_i$  and  $k_d$ , respectively;  $N$  is the total number of candidates for any parameters. For reducing the time of optimization search, the maximum and minimum values are assumed known. The values of candidates' elements of  $k_p$ ,  $k_i$  and  $k_d$  are equally distributed between these values (maximum and minimum values). The problem is to find the optimal value for each parameter that minimizes the cost function.

Figure 6 illustrates the graphical representation for selecting the best parameters of the PID controller using the ACO algorithm. The three columns represent the tunable parameters of a PID controller. The selection of the node in any column “by ant” is based on pheromone trails between nodes of each parameter. The connection matrix between the columns is called the pheromone matrix  $\tau \in \mathbb{R}^{3 \times N}$ . Each element of the pheromone matrix is denoted by  $\tau_{ij}$  for  $i = 1, 2, 3$  and  $j = 1, 2, \dots, N$ . The selection of the specific node from any column “by ant” is achieved according to the state transition rule as illustrated in algorithm no. 1.

Algorithm no.1 given below, summaries the tuning procedures of ant colony optimization algorithm to select optimal parameters for a PID. The algorithm starts by defining the inputs and outputs and then providing tuning processing based on the behavior of the ACO algorithm to obtain optimal



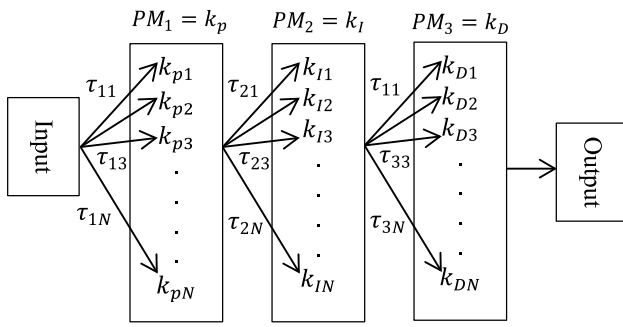


Fig. 6 Graphical representation of ACO for selection process

parameters of PID controller. The numbers of iterations that are needed to find the best parameters of the PID controller can be reduced by introducing knowledge in the range of parameters of the PID controller (maximum and minimum values). The learning tours of the ACO algorithm are stopped when the cost function is equal to or less than a small value  $\epsilon$ .

**Algorithm no.1:** ACO algorithm for PID controllers

**Inputs:** Initial candidates  $k_p, k_i$  and  $k_d$ .,  $\tau_{ij}(0) = \tau_0$ , Set  $q \in [0,1]$ ,  $q_0 \in [0,1]$  and  $q_1 \in [0,1]$ ,  $P_{min}^i, P_{max}^i, \tau \in \mathbb{R}^{3 \times N}$ ,  $\epsilon$

**Output:** optimized parameters  $k_p, k_i$  and  $k_d$ .

```

1 Loop
2   for i=1 to 3
3     for j=1 to N
4        $PM_i \in [P_{min}^i, P_{max}^i], P_{i1} = P_{min}^i, P_{iN} = P_{max}^i$ ,
5        $P_{ij} = P_{i(j-1)} + \frac{P_{max}^i - P_{min}^i}{N-1}$ 
6     end
7   end
8   while
9      $j = \begin{cases} \arg \max_{y \in j} \{\tau_{iy}(t)\} & \text{if } q \leq q_0 \\ j_1 & \text{if } q_0 < q < q_1 \\ j_2 & \text{if } q \geq q_1 \end{cases}$ 
10     $e(t) = r(t) - c(t)$ 
11     $u_{PID}(t) = k_p e(t) + k_i \int e(t) dt + k_d \dot{e}(t)$ 
12     $Cost \text{ function } (CF) = \sum_{f=0}^{iter_{max}} e_f^2$ 
13    if  $CF \leq \epsilon$ 
14      then,  $j = [k_p, k_i, k_d]$ 
15    else go to 9
16  end
17 end
18 end
    
```

In other words, we firstly initialized the elements of the parameter matrix (PM) where  $\tau_{ij}(0) = \tau_0$ , secondly, the selection of the values of candidates for each control parameter has been accomplished by using Step no. 4 in the Algorithm no. 1 above. Thirdly, the choice of the PID controller’s parameters has been achieved by using Step no. 9 in the same algorithm. Fourthly, the control signal of the PID controller is computed by using Eq. (17) given earlier which is applied to the plant. Then, the value of the cost function needs to be calculated by Step no. 12 as in the above algorithm. Finally, the value of the cost function is compared with the set value  $\epsilon$ , consequently, if the cost function  $(CF) \leq \epsilon$ , then the optimal values are reached, otherwise, another set of PID controller’s parameters will be chosen based on Step no. 9 and the following will re-occurring until the main condition of the cost function is satisfied. It seen that, the cost function is based on a mean squared error.

**3.2 Fuzzy Controller**

Fuzzy control has successfully proved its suitability and effectiveness to solve complex problems [35–38]. The principle operation of Fuzzy control systems is based on multiple stages, i.e., fuzzification, inference engine, and defuzzification. The implementation of those stages is based on selecting appropriate membership functions and inference engine rules which are specifically designed to a particular problem. In this work, twelve fuzzy controllers are designed (one for each joint) to control the gait of a quadruped robot. Each fuzzy control system has two inputs (an error and a change of error) and one output (control signal). The input membership functions are assumed symmetrical about the y-axis. For each input variable, five equally spaced triangular membership functions are selected. Other types of membership functions have been tested such as Gaussian functions and Bell functions. However, it is found that the triangular membership functions are more suitable than the others. It is assumed that the parameters of the input membership functions are inadaptable. The universe of discourse of error membership functions is ranged from ‘–20’ to ‘20’, while the universe of discourse of change of error membership functions is ranged from ‘–5’ to ‘5’. The input membership functions are labelled as Negative Big (NB), Negative Small (NS), Zero (ZE), Positive Small (PS), and Positive Big (PB). While the distribution of seven output triangular membership functions on the output universe of discourse is initialized randomly. Table 1 shows the rules relationships between error and change of error from one side and the output ‘q’ from the other side. The centres of output membership functions are assumed adaptable. Figure 7 below shows the block diagram of a Fuzzy controller for one link in a leg.

As explained earlier regarding the tuning of PID controllers using the ACO algorithm, the latter is used to tune the parameters of the centres of membership functions the fuzzy controllers. The algorithm no. 2 demonstrates the steps to reach optimal values of centres of output membership functions. Likewise, the algorithm starts by identifying the inputs and the outputs, ranging boundaries of the minimum and maximum values. The control signal generated from the fuzzy controller is computed from the equation below:

$$u(x) = \frac{\sum_i u_i(x) \times \delta_{\text{premise}(i)}(x)}{\sum_i \delta_{\text{premise}(i)}(x)} \tag{19}$$

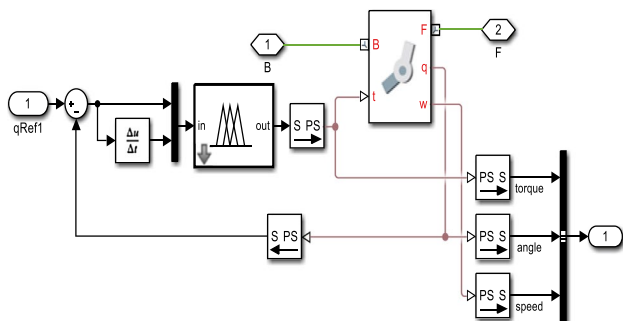
where  $u_i(x)$  is the output value of  $i$ th rule,  $\delta_{\text{premise}(i)}(x)$  is the value of the premise. It is computed from the following equation.

$$\delta_{\text{premise}(i)}(x) = \min\{\delta_{A_{i1}}(x_1), \delta_{A_{i2}}(x_2), \dots, \delta_{A_{in}}(x_n)\} \in [0, 1], \forall x_i \in X, \text{ where } i = 1, \dots, R \tag{20}$$

where:  $\delta_{A_{ik}}(x_k)$  is the input membership function of  $k$ th linguistic value; and  $X$  is the universe of discourse of the input linguistic variables.

**Table 1** The rules inference for each fuzzy controller

Output (q)	Change in error ( $\dot{e}$ )				
	NB	NS	ZE	PS	PB
Error ( $e$ )					
NB	NS	NS	NS	ZE	PB
NM	NS	NB	ZE	PS	PB
NS	NB	ZE	PS	PS	PS
ZE	NS	NB	PS	PB	PB
PS	NS	NS	PB	PB	PB



**Fig. 7** Fuzzy controller for one link in a leg

**Algorithm 2:** ACO algorithm for Fuzzy Controllers

**Inputs:** Initial candidates center of membership functions.,  $\tau_{ij}(0) = \tau_0$ , set  $q \in [0,1]$ ,  $q_0 \in [0,1]$  and  $q_1 \in [0,1]$ ,  $P_{min}^i, P_{max}^i, \tau \in \mathbb{R}^{3 \times N}, \varepsilon$

**Output:** optimized the center of membership functions.

```

1 Loop
2 for i=1 to 7 (centers of output membership functions)
3   for j=1 to N (N of columns)
4      $PM_i \in [P_{min}^i, P_{max}^i], P_{i1} = P_{min}^i, P_{iN} = P_{max}^i,$ 
5      $P_{ij} = P_{i(j-1)} + \frac{P_{max}^i - P_{min}^i}{N-1}$ 
6   end
7 end
8 while
9    $j = \begin{cases} \arg \max_{y \in j} \{\tau_{iy}(t)\} & \text{if } q \leq q_0 \\ j_1 & \text{if } q_0 < q < q_1 \\ j_2 & \text{if } q \geq q_1 \end{cases}$ 
10   $e(t) = r(t) - c(t)$ 
11   $u(x) = \frac{\sum_i u_i(x) \times \delta_{\text{premise}(i)}(x)}{\sum_i \delta_{\text{premise}(i)}(x)}$ 
12   $CF = \sum_{f=0}^{iter_{max}} e_f^2$ 
13  if  $FF \leq \varepsilon$ 
14    then, j= Centre of membership functions
15  else go to 9
16 end
17 end
18 end
    
```

**4 Lyapunov Stability Test for Fuzzy System**

To prove that the controlled system stays stable and has an acceptable response, the testing of the stability of the proposed controller should be achieved to confirm the effectiveness and suitability of the proposed controller even different significant disturbances are applied to the controlled system. Therefore, in this section, the Lyapunov stability criterion [39] is used to test the stability of the fuzzy control system. The Lyapunov stability criterion states that the nonlinear system is stable if the following conditions are satisfied [40]:

1.  $\mu(x) > 0$  for all values of states  $x \neq 0$ .
2.  $\dot{\mu}(x) \leq 0$  for all values of states  $x$ .



where  $\mu(x)$  is a Lyapunov function which is assumed as:

$$\mu(x) = x^T \rho x \tag{21}$$

where  $\rho \in \mathbb{R}^{n \times n}$  is assumed as a symmetric positive definite matrix. Depending on this assumption, the first condition of the Lyapunov stability criterion is satisfied. The first derivative of the Lyapunov function is written as:

$$\dot{\mu}(x) = \dot{x}^T \rho x + x^T \rho \dot{x} \tag{22}$$

The state-space equations of any nonlinear system can be written as [1]:

$$\dot{x} = A(x) + B(x)u(x) \tag{23}$$

where:  $x \in \mathbb{R}^{n \times 1}$  is the state variable;  $A(x) \in \mathbb{R}^{n \times 1}$  is non-linear function vector;  $B(x) \in \mathbb{R}^{n \times 1}$  is nonlinear function vector;  $u(x)$  is the control signal and  $n$  is the number of the state variables.

If the fuzzy system is proposed as a controller for a nonlinear system, the control signal is given by Eq. (18).

Therefore, Eq. (21) can be rewritten as:

$$\therefore \dot{\mu}(x) = \Gamma(x) + \Phi(x)u(x) \tag{24}$$

where the terms of the equation above are given by:  $\Gamma(x) = A(x)^T \rho x + x^T \rho A(x)$  and  $\Phi(x) = B(x)^T \rho x + x^T \rho B(x)$

The vector  $\Phi(x)$  is defined into three regions:

1. Positive region:  $\Phi(x) > 0$  or it is described as the following set:  $\Phi_{+ve} = \{x \in X | \Phi(x) > 0\}$ .
2. Negative region:  $\Phi(x) < 0$  or it is described as the following set:  $\Phi_{-ve} = \{x \in X | \Phi(x) < 0\}$ .
3. Zero region:  $\Phi(x) = 0$  or it is as the following set:  $\Phi_{ze} = \{x \in X | \Phi(x) = 0\}$ .

If the Lyapunov function exists, then the following points should be satisfied:

1.  $\Gamma(x) \leq 0, \forall x \in \Phi_{ze}$ .
2.  $u_i(x) \leq -\frac{\Gamma(x)}{\Phi(x)}$  for  $x \in X_i^a \cap \Phi_{+ve}$  and  $u_i(x) \geq -\frac{\Gamma(x)}{\Phi(x)}$  for  $x \in X_i^a \cap \Phi_{-ve}$  for  $i = 1, \dots, R$ .
3. For any initial state  $x_0 \in X$ , the set  $\{x \in X | \dot{\mu}(x) = 0\}$ .

Now, the second condition of the Lyapunov stability criterion should be satisfied for all regions of vector  $\Phi(x)$ .

1. If  $\Phi(x)$  is positive, for any arbitrary initial state  $x_0 \in X$ , then:

$$u_i(x_0) \leq -\frac{\Gamma(x_0)}{\Phi(x_0)} \Rightarrow u(x_0) = \frac{\sum_i u_i(x_0) \times \delta_{premise(i)}(x_0)}{\sum_i \delta_{premise(i)}(x_0)} \leq \frac{-\Gamma(x_0) \sum_i \delta_{premise(i)}(x_0)}{\sum_i \delta_{premise(i)}(x_0)} = -\frac{\Gamma(x_0)}{\Phi(x_0)}$$

$$\therefore \dot{\mu}(x_0) = \Gamma(x_0) + \Phi(x_0)u(x_0) \leq \Gamma(x_0) + \Phi(x_0) \left( -\frac{\Gamma(x_0)}{\Phi(x_0)} \right) = 0$$

$\therefore \dot{\mu}(x) \leq 0 \Rightarrow$  the second condition of the Lyapunov stability criterion is satisfied.

2. If  $\Phi(x)$  is negative, for any arbitrary initial state  $x_0 \in X$ , then:

$$u_i(x_0) \geq -\frac{\Gamma(x_0)}{\Phi(x_0)} \Rightarrow u(x_0) = \frac{\sum_i u_i(x_0) \times \delta_{premise(i)}(x_0)}{\sum_i \delta_{premise(i)}(x_0)} \geq \frac{-\Gamma(x_0) \sum_i \delta_{premise(i)}(x_0)}{\sum_i \delta_{premise(i)}(x_0)} = -\frac{\Gamma(x_0)}{\Phi(x_0)}$$

$$\therefore \dot{\mu}(x_0) = \Gamma(x_0) + \Phi(x_0)u(x_0) \leq \Gamma(x_0) + \Phi(x_0) \left( -\frac{\Gamma(x_0)}{\Phi(x_0)} \right) = 0$$

$\therefore \dot{\mu}(x) \leq 0 \Rightarrow$  the second condition of the Lyapunov stability criterion is satisfied.

3. If  $\Phi(x)$  is zero, for any arbitrary initial state  $x_0 \in X$ , then:

$$\dot{\mu}(x_0) = \Gamma(x_0) + \Phi(x_0)u(x_0) = \Phi(x_0) \leq 0$$

$\therefore \dot{\mu}(x) \leq 0 \Rightarrow$  the second condition of the Lyapunov stability criterion is satisfied.

It can be seen that, the Lyapunov condition for the stability requirement is satisfied. Therefore, the proposed fuzzy controller is globally asymptotically stable.

## 5 Simulation Results

The simulation results are conducted to investigate the performance of the designed controllers and validate their effectiveness in guiding a quadruped robot to reach a given destination without interruptions. Three case studies are considered, i.e., firstly, an open-loop system where no controllers are used in the robot movement. Secondly, a closed-loop system using PID controllers and thirdly, a closed-loop system using fuzzy controllers as demonstrated in the following subsections.

### 1. Case study 1: open-loop system

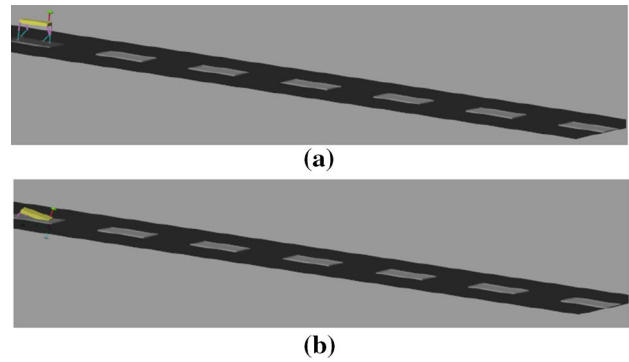
In this case, no controllers are used in the looping system and it is required by the quadruped robot to achieve a feasible movement linearly on a road. The reference movement in the input is based on chosen angles. The robot should be able to complete its motion stably with being collapsed. However, this could not be approachable as the robot has lost its stability and collapsed immediately after the simulation starts. Figure 8a shows the quadruped robot standing at the initial position before running the simulation. When the operation is activated, the robot is collapsed without elapsing any distance as depicted in Fig. 8b.

### 2. Case study 2: closed loop using PID controller

From the previous results, it was observed that there is a need to design a more efficient controller to stabilize the movement of the quadruped robot and move it smoothly and feasibly to its destination without breakdown. In response to such a situation, we endeavor to introduce PID controllers. As there are twelve servomotors in the robotic mechanism, therefore, twelve PID controllers are needed to control the movement of the quadruped robot parallelly across all the three links in the four legs. In order to achieve the design of the twelve PID controllers, an ACO algorithm is utilized to tune the parameters of each PID controller to reach optimal performance. The required data for the ACO algorithm is provided as in Table 2 below, which illustrates the number of iterations, ants, and nodes. After completing the tuning process, the optimal parameters of each PID controller are obtained as given in Table 3.

The simulation results using PID controllers have demonstrated some improvements over the previous case study as shown in Fig. 9 below. The quadruped robot commences his movement successfully starting from Fig. 9a and has reached a selected position after a few meters as shown in Fig. 9b. However, the quadruped robot has not been able to reach the expected target as it is halted at the middle distance as can be noted in Fig. 9c.

The control efforts for the proposed PID controller are provided to demonstrate its effectiveness. Figure 10 shows the control efforts of the front left leg for three links, i.e., ankle, hip, and knee. The time responses of the control efforts illustrate that their values reach high magnitudes about 350 Nm. Such values mean that the PID control exerts high efforts to drive the quadruped robot. In Figs. 11, 12 and 13 demonstrate the control efforts for the other three legs. The simulation results show that the control efforts are quite high.



**Fig. 8** Case study 1 **a** quadruped robot stands at its initial posture, **b** quadruped robot has just attempted to move but collapsed immediately at the same position, this is without any controller

**Table 2** The parameter of the ACO algorithm for PID controllers

Parameter	Value
No. of iteration	120
No. of ants	300
No. of nodes	700

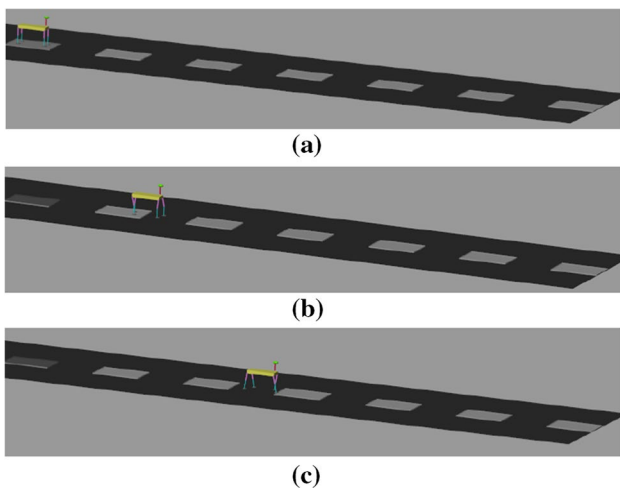
### 3. Case study 3: closed loop using fuzzy controller

As provided earlier, there are still some drawbacks in the overall operation of the quadruped robot. In order to obtain the best performance possible, we propose novel Fuzzy controllers to control the motion of the quadruped robot and stabilize its movement until it reaches the required position. The best-chosen parameters for the ACO algorithm are provided in Table 4. These parameters are the number of iterations, ants and nodes which represent the main elements in implementing the ACO algorithm. Accordingly, the five centres of areas are obtained as given in Table 5 for all the twelve Fuzzy controllers. From Fig. 14a–c, it is observable that the quadruped robot has stably and feasibly managed to travel along the given distance without being halted at any position in its route while moving. Based on the obtained results, it has been confirmed that all the aforementioned implications have been successfully eliminated and the proposed Fuzzy controllers have improved the overall performance significantly.

Similarly, the control efforts for the proposed Fuzzy controllers are obtained to demonstrate its performance as shown in Figs. 15, 16, 17 and 18. The control efforts are provided for every three links per leg which means twelve control signals are delivered to the system. It is noticeable that the control efforts for Fuzzy controllers have been decreased significantly in comparison with the PID controllers. This, in turn, provides a Fuzzy controller with great advantage as this is a key improvement in addition to the stabilized movement along the given path.

**Table 3** Optimal parameters of PID controllers

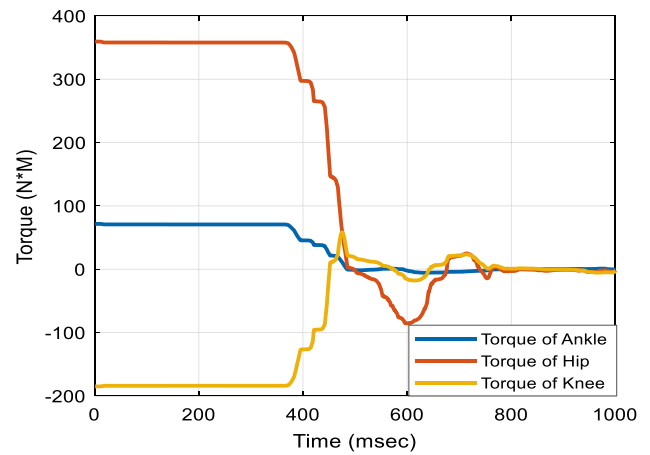
PID controller based on ACO	$K_p$	$K_i$	$K_d$
Controller #1	50.1254	2.5487	3.6457
Controller #2	60.3548	1.2459	3.6548
Controller #3	64.2648	2.2457	2.3648
Controller #4	55.3467	4.0215	1.2486
Controller #5	77.2548	2.6541	0.9875
Controller #6	82.6457	1.5642	1.0245
Controller #7	55.3467	4.0215	1.2486
Controller #8	77.2548	2.6541	0.9875
Controller #9	82.6457	1.5642	1.0245
Controller #10	50.1254	2.5487	3.6457
Controller #11	60.3548	1.2459	3.6548
Controller #12	64.2648	2.2457	2.3648



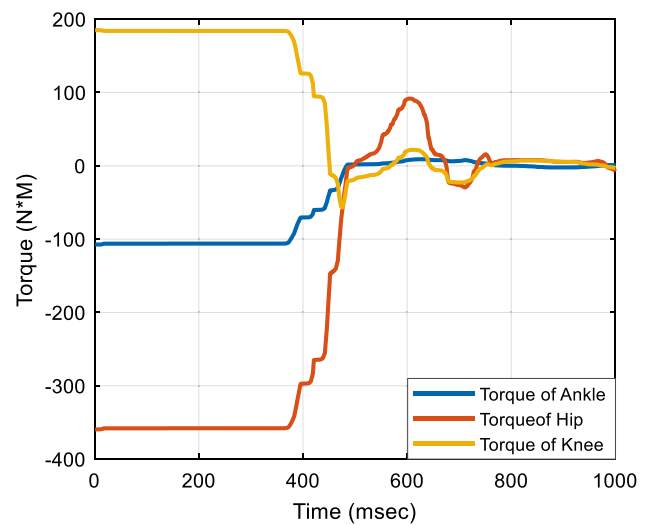
**Fig. 9** Case study 2 **a** quadruped robot stands at its initial posture, **b** quadruped robot has moved a few metres and **c** when the quadruped robot has proceeded further but this is the final reached position based on PID controllers

### 6 Comparison of the Proposed Controllers

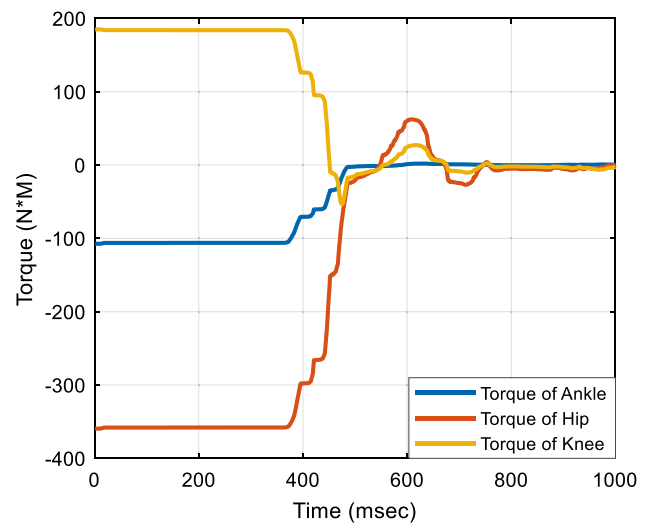
In this section, we introduce some comparisons to investigate the suitability and effectiveness of the proposed controllers, i.e., PID and Fuzzy controllers in controlling the quadruped robot to move stability over the given path and reach its destination. The comparisons are based on the total travelled distances in relationship to the elapsed time for the whole path and the values of the control effort applied in the designed controllers. As can be clearly seen in Fig. 19, the Fuzzy controllers have led the quadruped robot to reach its final destination. Therefore, the total distance is higher compared to the other two case studies. In the case of no



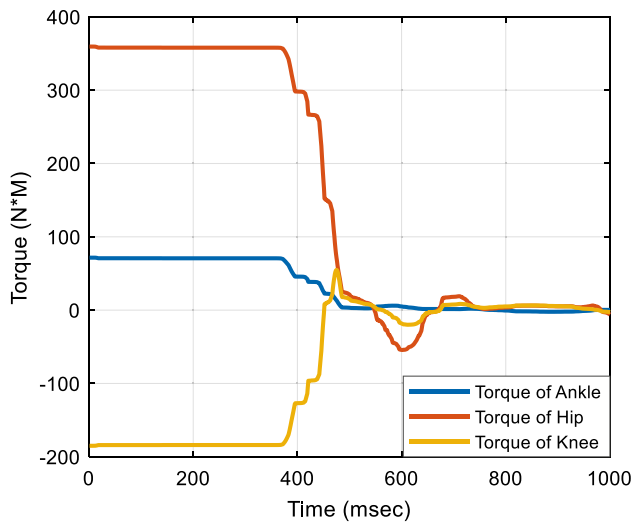
**Fig. 10** Control efforts of PID controller for front left leg



**Fig. 11** Control efforts of PID controller for front right leg



**Fig. 12** Control efforts of PID controller for rear right leg



**Fig. 13** Control efforts of PID controller for rear left leg

**Table 4** The parameter of the ACO algorithm for Fuzzy controllers

Parameter	Value
No. of iteration	10
No. of ants	50
No. of nodes	60

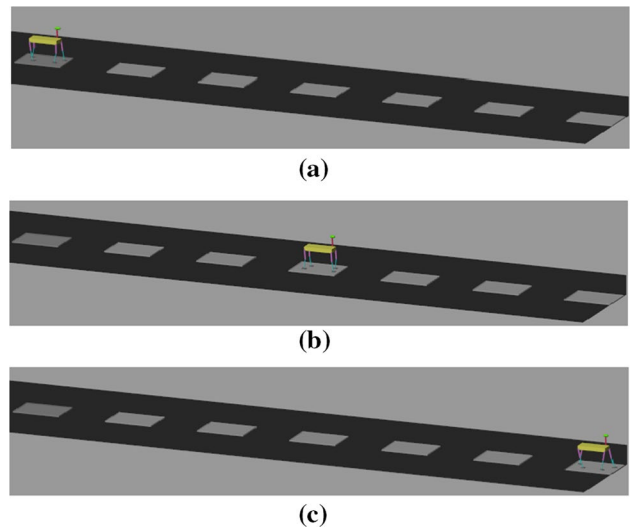
controller is applied, the distance is zero as the quadruped robot has not travelled any noticeable distance. Only, a few meters have been travelled when the PID controllers are utilized. However, this is not the target as the quadruped robot is unable to reach the final required position.

The execution time is made to be fixed in all conducted case studies and it equals to 35 s. Regarding the scalability of the proposed algorithms, it has been observed that the performance of the Fuzzy controller presents significant adaptability of the designed system in comparison with the performance of the PID controller.

Furthermore, control efforts have been compared between Fuzzy and PID controllers to evaluate the significance of improvements between both types of controllers. The comparisons are only conducted for the front right leg; thus, three comparisons are needed for each link. In Fig. 20, the control efforts for the ankle joint of both Fuzzy and PID controllers are provided. They clearly demonstrate the effectiveness of the Fuzzy controller in achieving the target while minimizing the control efforts remarkably. Likewise, in Fig. 21, it is observable that the control effort of the Fuzzy controller for the hip joint is reduced. Finally, the last comparison is obtained as depicted in Fig. 22 for the knee joint which also shows a feasible response of the control efforts for the Fuzzy controller over the PID controller. These simulation results demonstrate and validate the privileges of the designed Fuzzy controller over the PID controller.

**Table 5** Optimal centres of area for Fuzzy controllers

Fuzzy controller based on ACO	Centre				
	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
Controller #1	-5.5	-3.8	-1.01	2.789	6.74
Controller #2	-8.1	-3.4	0.97	2.45	6.45
Controller #3	-9.45	-5.234	0.95	3.02	8.36
Controller #4	-7.5	-3.3	0.59	3.54	6.34
Controller #5	-10.3	-4.12	0.015	3.25	11.23
Controller #6	-12.2	-6.36	0.701	4.16	8.23
Controller #7	-7.49	-3.37	0.59	3.54	6.348
Controller #8	-10.3	-4.12	0.015	3.25	11.233
Controller #9	-12.2	-6.36	0.701	4.16	8.23
Controller #10	-5.51	-3.84	-1.01	2.78	6.74
Controller #11	-8.12	-3.45	0.975	2.45	6.45
Controller #12	-9.45	-5.23	0.952	3.02	8.369



**Fig. 14** Case study 3 **a** quadruped robot stands at its initial posture, **b** quadruped robot reaches approximately the middle distance and **c** when the quadruped robot has successfully reached the target based on Fuzzy controllers

## 7 Conclusion

In this paper, we proposed the modelling of the quadruped robot based on three movable links per leg. This will introduce the robot with great advantages to move feasibly and smoothly. The architecture of this design has been implemented for each individual link and combined together to obtain the whole system. The latter has been tested without applying any controller and it is noticed that the robot is not capable of performing any movement. Hence, we proposed two novel controllers, i.e., PID and Fuzzy controllers which there are twelve controllers for each servomotor in each link.

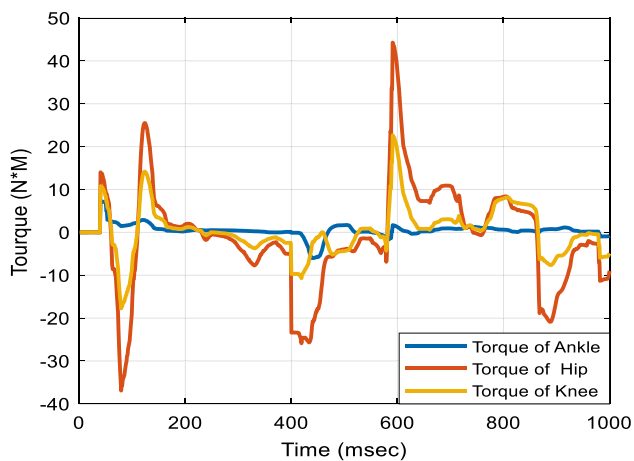


Fig. 15 Control signal of Fuzzy controller for front left leg

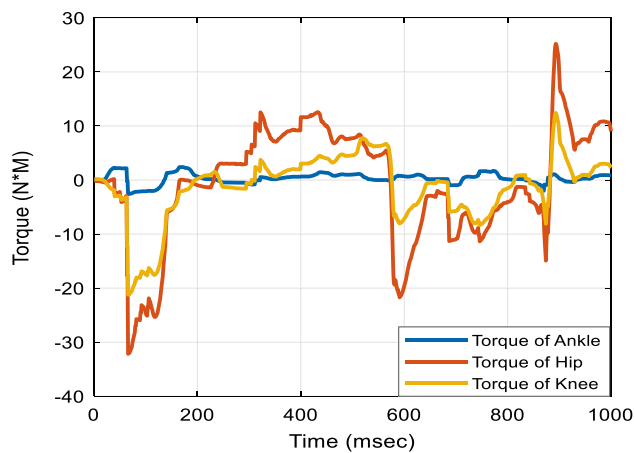


Fig. 18 Control signal of Fuzzy controller for front right leg

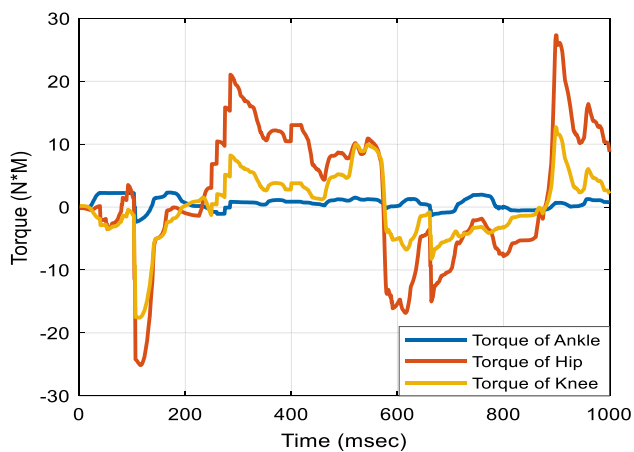


Fig. 16 Control signal of Fuzzy controller for rear left leg

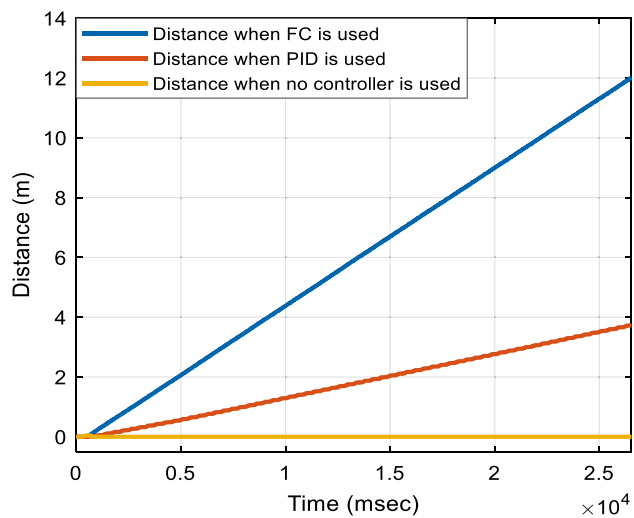


Fig. 19 Travelled distance for the three case-studies

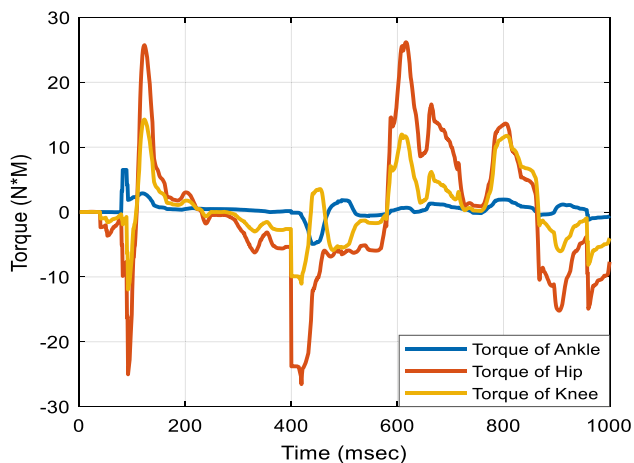


Fig. 17 Control signal of Fuzzy controller for rear right leg

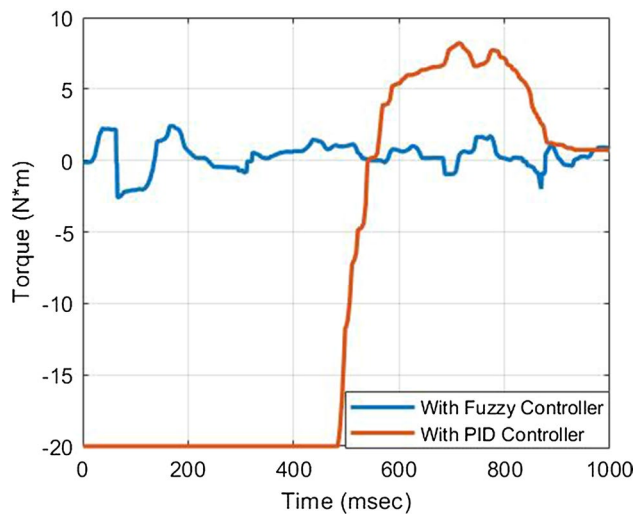


Fig. 20 Control efforts of the ankle joint

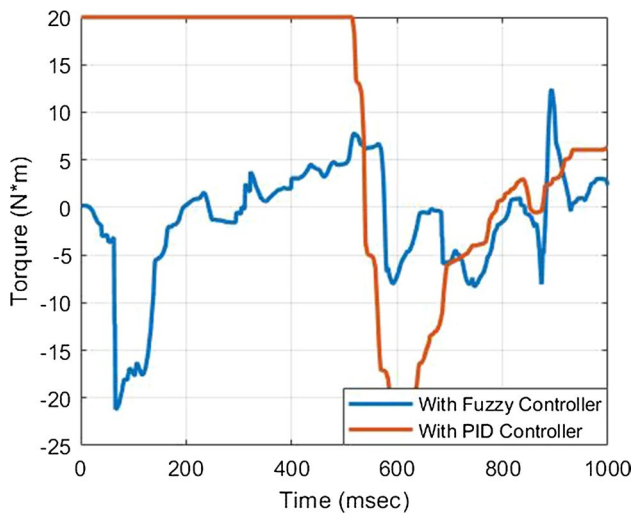


Fig. 21 Control efforts of the hip joint

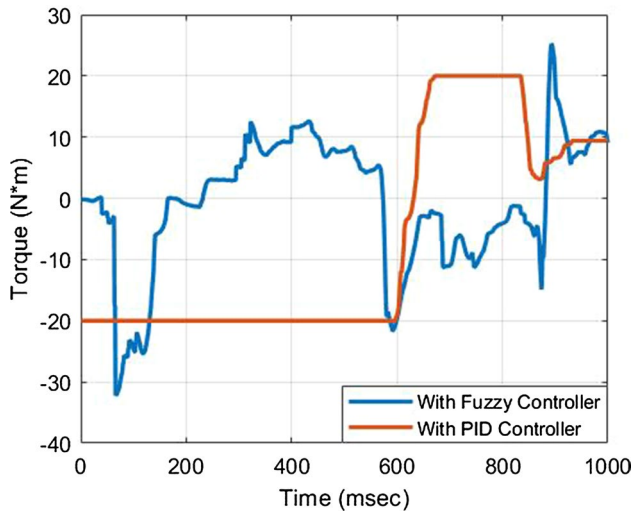


Fig. 22 Control efforts of the knee joint

The PID controller has demonstrated a relatively practical start but has not been able to take the quadruped robot to the final position. Instead, the Fuzzy controller is capable of achieving the required of our design and has shown effectiveness and suitability after the performance has been validated. The parameters of both PID and Fuzzy controllers have been tuned using an ant colony optimization algorithm which has optimized the performance of each designed controller by obtaining its optimal parameters.

To investigate the overall performance of the proposed controllers and the whole designed system, two main comparisons are introduced based on the travelled distance against the elapsed time and the control efforts. The simulation results have shown high effectiveness and robustness

of the proposed Fuzzy controllers in comparison to PID controllers. Moreover, the stability of the Fuzzy controllers has been approved by using the Lyapunov stability method.

In future work, the proposed three links per leg architecture will be constructed using a 3D printer. Hence, real-time experiments will be conducted to validate the proposed control systems practical and then some comparisons will be done between the theoretical and practical experiments thoroughly.

## Compliance with Ethical Standards

**Conflict of interest** The authors declare that there is no conflict of interest. The authors alone are responsible for the content and writing of this article.

## References

1. Chwa D (2010) Tracking control of differential-drive wheeled mobile robots using a backstepping-like feedback linearization. *IEEE Trans Syst Man Cybern Part A Syst Hum* 40(6):1285–1295
2. Wang H, Li B, Liu J, Yang Y, Zhang Y (2011) Dynamic modeling and analysis of Wheel Skid steered Mobile Robots with the different angular velocities of four wheels. In: *The 30th Chinese control conference*, pp 3919–3924
3. Sun T, Xiang X, Su W, Wu H, Song Y (2017) A transformable wheel-legged mobile robot: design, analysis and experiment. *Robot Auton Syst* 98(September):30–41
4. Li X, Gao H, Li J, Wang Y, Guo Y (2019) Hierarchically planning static gait for quadruped robot walking on rough terrain. *J Robot* 2019:1–12
5. Kumar GK, Pathak PM (2013) Dynamic modelling an simulation of a four legged jumping robot with compliant legs. *Robot Auton Syst* 61(3):221–228
6. Peula JM, Urdiales C, Herrero I, Sánchez-Tato I, Sandoval F (2009) Pure reactive behavior learning using case based reasoning for a vision based 4-legged robot. *Robot Auton Syst* 57(6–7):688–699
7. Aiyama Y, Hara TYM, Ota J, Arai T (1999) Cooperative transportation by two four-legged robots with implicit communication. *Robot Auton Syst* 29(1):13–19
8. Martins-Filho LS, Prajoux R (2000) Locomotion control of a four-legged robot embedding real-time reasoning in the force distribution. *Robot Auton Syst* 32(4):219–235
9. Fukui T, Fujisawa H, Otaka K, Fukuoka Y (2019) Autonomous gait transition and galloping over unperceived obstacles of a quadruped robot with CPG modulated by vestibular feedback. *Robot Auton Syst* 111:1–19
10. Lee JH, Park JH (2019) Time-dependent genetic algorithm and its application to quadruped's locomotion. *Robot Auton Syst* 112:60–71
11. Chow CK, Jacobson DH (1972) Further studies of human locomotion: postural stability and control. *Math Biosci* 15(1–2):93–108
12. Hemami H, Golliday CL Jr (1977) The inverted pendulum and biped stability. *Math Biosci* 34(1–2):95–110
13. Furusho J, Masubuchi M (1986) Control of a dynamical biped locomotion system for steady walking. *J Dyn Syst Meas Control* 108(2):111–118



14. Hirose S (1984) A study of design and control of a quadruped walking vehicle. *Int J Robot Res* 3(2):113–133
15. Liu M, Xu F, Jia K, Yang Q, Tang C (2016) A stable walking strategy of quadruped robot based on foot trajectory planning. In: 3rd international conference on information science and control engineering, pp 799–803
16. Agrawal SP, Dagale H, Mohan N, Umanand L (2016) IONS: a quadruped robot for multi-terrain applications. *Int J Mater Mech Manuf* 4(1):84–88
17. Sprlak R, Chlebis P (2014) A study on locomotions of quadruped robot. In: International conference on advanced engineering—theory and application, pp 125–133
18. Atique MMU, Sarker MRI, Ahad MAR (2018) Development of an 8DOF quadruped robot and implementation of Inverse Kinematics using Denavit–Hartenberg convention. *Heliyon* 4(12):2405–8440
19. Moosavian SAA, Khorram M, Zamani A, Abedini H (2011) PD regulated sliding mode control of a quadruped robot. In: IEEE international conference on mechatronics and automation. IEEE, Beijing, pp 2061–2066
20. Gor MMS, Pathak PM, Samantaray AK, Yang JM, Kwak SW (2015) Control of compliant legged quadruped robots in the workspace. *Simulation* 91(2):103–125
21. Sun L, Meng M, Chen W, Liang H, Mei T (2007) Design of quadruped robot based neural network. In: International symposium on neural networks, pp 843–851
22. Li K, Wen R (2017) Robust control of a walking robot system and controller design. In: 13th global congress on manufacturing and management, pp 947–955
23. Bhatti J, Hale M, Irvani P, Plummer A, Sahinkaya N (2017) Adaptive height controller for an agile hopping robot. *Robot Auton Syst* 98:126–134
24. Liu Q, Chen X, Han B, Luo Z, Luo X (2018) Learning control of quadruped robot galloping. *J Bionic Eng* 15(2):329–340
25. Potts AS, Da Cruz JJ (2010) Kinematics analysis of a quadruped robot. In: 5th IFAC symposium on mechatronic systems, pp 261–266
26. Gor M, Pathak P, Samantaray A (2013) Dynamic modeling and simulation of compliant legged quadruped robot. In: The 1st international and 16th national conference on machines and mechanisms, pp 7–16
27. Zeng X, Zhang S, Zhang H, Li X, Zhou H, Fu Y (2019) Leg trajectory planning for quadruped robots with high-speed trot gait. *Appl Sci* 9(7):1508
28. McMillan GK (2012) Industrial applications of PID control. In: Vilanova R, Visioli A (eds) PID control in the third millennium. *Advances in industrial control*. Springer, London, pp 415–461
29. Meena DC, Devanshu A (2017) Genetic algorithm tuned PID controller for process control. In: International conference on inventive systems and control
30. Solihin MI, Tack LF, Kean ML (2011) Tuning of PID controller using particle swarm optimization. In: international conference on advanced science, engineering and information technology, pp 458–461
31. Ali RS, Aldair AA, Almousawi AK (2014) Design an optimal PID controller using artificial bee colony and genetic algorithm for autonomous mobile robot. *Int J Comput Appl* 100(16):8–16
32. Hsiao Y-T, Chuang C-L, Chien C-C (2004) Ant colony optimization for designing of PID controllers. In: International conference on robotics and automation
33. Lakshmi Narayana KV, Kumar VN, Dhivya M, Prejila Raj R (2015) Application of ant colony optimization in tuning a PID controller to a conical tank. *Indian J Sci Technol* 8(S2):217
34. Ning J, Zhang C, Sun P, Feng Y (2018) Comparative study of ant colony algorithms for multi-objective optimization. *Information* 10(1):1–19
35. Lee CCC (1990) Fuzzy logic in control systems: fuzzy logic controller, Part II. *IEEE Trans Syst Man Cybern* 20(2):404–418
36. Almayyahi A, Wang W, Hussein A, Birch P (2017) Motion control design for unmanned ground vehicle in dynamic environment using intelligent controller. *Int J Intell Comput Cybern* 10(4):530–548
37. Aldair AA, Alsaedee EB, Abdalla TY (2019) Design of ABCF control scheme for full vehicle nonlinear active suspension system with passenger seat. *Iran J Sci Technol Trans Electr Eng* 43:289–302
38. Al-Mayyahi A, Ali RS, Thejel RH (2018) Designing driving and control circuits of four-phase variable reluctance stepper motor using fuzzy logic control. *Electr Eng* 100(2):695–709
39. Pukdeboon C (2011) A review of fundamentals of Lyapunov theory. *J Appl Sci* 10(2):55–61
40. Ogata K (1995) *Discrete time control systems*, 2nd edn. Prentice Hall International Inc, Upper Saddle River

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Ammar A. Aldair** was born in Basrah, Iraq. He received his B.Sc. Degree in Electrical Engineering from University of Basrah, Iraq in 2000. In 2003, he received his M.Sc. Degree in Control and Systems Engineering from University of Basrah, Iraq. In 2012, he received his PhD Degree in Control and Systems Engineering from University of Sussex, UK. From 2003–2008, he was a Lecturer in Electrical Department, University of Basrah, Iraq. He teaches many subjects such as: Mathematics, Logic Systems, Electrical Circuits, Electronic Circuits, Control Systems and Advance Control systems. He is currently a Prof Assistance at the University of Basrah. His current research interests in Design of Intelligent Control Systems.



**Auda Al-Mayyahi** received his B.Sc. and M.Sc. degrees in Electrical Engineering from University of Basrah, Iraq in 2003 and 2007, respectively. Additionally, he received his Ph.D. degree in industrial automation from University of Sussex, UK in 2018. Currently, he is a lecturer at University of Basrah in which he has taught different topics within the area of electrical engineering. His research interests include Robotics, Automation and Intelligent Control Systems.



**Weiji Wang** received MEng and DEng from University of Nanjing Aeronautics and Astronautics, China in 1980s, and DPil from University of Oxford, UK in 1994. Currently he works as a senior lecturer at University of Sussex, UK. His research interests include signal processing, system control, automotive engineering, and aeronautical engineering.