Intelligent Service Robotics
https://doi.org/10.1007/s11370-019-00309-3

ORIGINAL RESEARCH

Fuzzy-backstepping control of quadruped robots

Shabnam Shakourzadeh[1] · **Mohammad Farrokhi**[1]

Received: 5 August 2017 / Accepted: 19 December 2019

Abstract
In this paper, fuzzy-backstepping control method is utilized to control quadruped robots. The backstepping control method is inherently a stable control method since its design procedure is based on the Lyapunov stability theory. However, for an acceptable performance of this controller, the feedback gains must be appropriately selected. Using fuzzy systems, these gains are adaptively tuned for best performance of the robot against uncertainties in system parameters, external disturbances, and measurement noises. Next, the stability of the robot is investigated using the Poincaré map stability criterion. This is because the Lyapunov method does not guarantee stability for the periodic systems such as legged robots. The proposed method is simulated on the highly nonlinear equations of a quadruped robot. The performance of the proposed method is compared with the backstepping control to show the importance of fine-tuning the parameters using fuzzy system. The simulating results for a nine-degree-of-freedom quadruped robot show very good performance of the proposed control method, especially against uncertainties in system parameters, external disturbances, and measurement noises, which are common in robots.

Keywords Quadruped robots · Backstepping control · Fuzzy systems · Poincaré map

1 Introduction

Control of the legged robots is an interesting and challenging research area that has attracted the attention of researchers in recent years [1–7]. It is well known that controlling the legged robots is more complex than the wheeled robots, because the legged robot must follow the desired path without becoming unstable. In addition, the controller design for stabilizing the motion is clearly necessary due to ability of the robot's walking on paths that contain obstacles, slopes, and uneven grounds [8]. However, there are challenges in designing the desired path and posture of every leg such that the stability of the robot is maintained by the controller at all times. Another issue is that the controller should be able to cope with uncertainties in the robot's model, external disturbances, and measurement noises. In this paper, the backstepping controller [9] in conjunction with the fuzzy systems is employed to fulfill these requirements with good performances. Backstepping controller has been used for controlling different type of robots. In [10], backstepping method for trajectory tracking of uncycle-type mobile robots was used which is based on lyapunov theorem. Also, in [11], backstepping method is used for trajectory tracking based on fuzzy sliding mode control for differential mobile robot. An adaptive backstepping controller is designed for n-degree-of-freedom manipulator robot based on combined state augmentation [12].

Wang et al. have used a three degree-of-freedom (DoF) model for each legs and the central pattern generator (CPG)-based gait control for the planar quadruped robots [13]. However, due to several DoFs in the quadruped robot, a proper control law requires many CPG units. Hence, many parameters need to be modulated in the CPG control network. Hence, determining the relationship between the parameters and the connection topology structure is a complex optimization problem.

In [14], the fuzzy control method and the adaptive resonance theory (ART) were used for the motion control of the quadruped robots in rough terrains. The fuzzy controller is also used in [15] for the TITAN-VIII robot, where an adaptive controller tunes the fuzzy IF–THEN rules according to the environments. For the stability, the zero moment point

✉ Shabnam Shakourzadeh
Sh.shakourzadeh@gmail.com

Mohammad Farrokhi
Farrokhi@iust.ac.ir

[1] Center of Excellence for Modelling and Control of Complex Systems, School of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran

Published online: 06 January 2020

(ZMP) criterion is employed. In [16], a robot with five links and four active joints is modeled. The dynamic model is a two-DoF nonholonomic system that is controlled by the invariant state-feedback method, which is a model-based method. However, since this method does not guarantee stability of the robot, they have used the backstepping method to control the robot against uncertainties in the system.

The backstepping controller is a nonlinear controller that can be used successfully for multi-input multi-output (MIMO) systems [17]. The main advantage of this controller is that it is inherently stable, because its design is based on the Lyapunov stability theory. However, one of the issues in the backstepping controller is determining its coefficients appropriately. In addition, these coefficients must be properly adapted to the changes in the system parameters. Since legged robots are prone to changes in the system parameters, which is unavoidable in real-life applications (e.g., carrying a load), fine-tuning the coefficients of the back stepping controller plays an important role. Otherwise, the robot may not perform well or even becomes unstable. In this manuscript, the fuzzy systems are employed to tune the coefficients of the backstepping controller adaptively. It will be shown that the proposed method is able to cope with uncertainties in robot's parameters, reduce measurement noises, and successfully rejects external disturbances. For stability check, the Poincaré map, which is suitable for periodic system such as the legged robots, is employed [18].

This paper is organized as follows. The mathematical modeling of the quadruped robot is introduced in Sect. 2. The path planning is introduced in Sect. 3. The controller design is presented in Sect. 4. Section 5 shows the stability of the closed-loop system using the Poincaré map. Section 6 provides the simulating results. Finally, Sect. 7 concludes the paper.

2 Modeling of robot

The robot considered in this manuscript is a quadruped robot with nine DoF moving in the sagittal plane. Due to the robot's complex dynamics, the gait generation for the quadruped robot is very important. The robot's movement must be performed in a way such that its stability is guaranteed. Moreover, the robot's legs must move in a predefined order. In addition, the distance between the swing leg and the support leg must obey some limitations. These issues are expressed in the followings in terms of some constraints for the robot.

The robot's legs are symmetrically located around the center of the body. Each leg has two DoF, and the body has one DoF (θ_9). Figure 1 shows the schematic diagram of this robot. The dynamic equations of the robot are derived based on this model in the followings. The kinematic equations of the robot are provided in "Appendix".

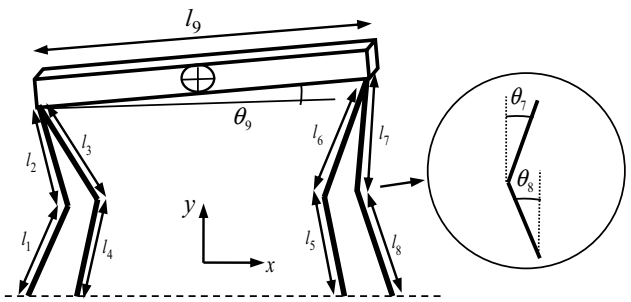

Fig. 1 Schematic diagram of a nine DoF quadruped robot moving in sagittal plane

If $(x_{\text{COM}}, y_{\text{COM}})$ and (x_{C_i}, y_{C_i}) are the coordinates of the center of mass (CoM) of the robot and the ith link of the robot in the x–y plane, respectively, then

$$x_{\text{COM}} = \frac{\sum_{i=1}^{9} m_i x_{C_i}}{\sum_{i=1}^{9} m_i}, \quad y_{\text{COM}} = \frac{\sum_{i=1}^{9} m_i y_{C_i}}{\sum_{i=1}^{9} m_i}. \tag{1}$$

where m_i is the mass of the ith link. The dynamic equation of the robot can be found using the well-known Lagrange method

$$L(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = K(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) - P(\boldsymbol{\theta}) \tag{2}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{9\times 1}$ and $\dot{\boldsymbol{\theta}} \in \mathbb{R}^{9\times 1}$ are the vectors of the angular position and velocity of the robot joints, respectively, and K and P are the total kinematic and potential energy of the robot, respectively, defined as

$$K = \sum_{i=1}^{9} \left[\frac{1}{2} m_i v_{C_i}^2 + \frac{1}{2} I_i \dot{\theta}_i^2\right], P = \sum_{i=1}^{9} \left[m_i g y_{C_i}\right] \tag{3}$$

in which g is the gravitational acceleration, I_i is the moment of inertia of the ith link, and v_{C_i} and y_{C_i} are the linear velocity and the y-coordinate of the center of mass (CoM) of the ith link, respectively.

The dynamic equations of the robot should be defined in two different phases: the swing phase and the support phase. During the swing phase, one leg of the robot moves forward, passing the opposite leg. On the other hand, during the support leg, all legs of the robot are on the ground.

For the swing phase, the Lagrange equation is defined as

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}_i}\right) - \frac{\partial K}{\partial \theta_i} + \frac{\partial P}{\partial \theta_i} = T_i \tag{4}$$

where T_i is the torque applied to the ith joint. Using (3), the dynamic equation of the robot in this phase can be obtained as

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{G}(\boldsymbol{\theta}) = \mathbf{T} \tag{5}$$

where $\mathbf{M}(\boldsymbol{\theta}) \in \mathbb{R}^{9\times9}$ is the matrix of the inertia terms, $\mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathbb{R}^{9\times9}$ is the matrix of the centrifuge and Coriolis terms, $\mathbf{G}(\boldsymbol{\theta}) \in \mathbb{R}^{9\times1}$ is the vector of the gravitational terms, and $\mathbf{T} \in \mathbb{R}^{9\times1}$ is the vector of the joint torques.

For the support phase, since all four legs of the robot are on the ground, the following holonomic constraints must be considered:

$$\boldsymbol{\Phi}(\boldsymbol{\theta}) = \begin{bmatrix} l_1c_1 + l_2c_2 - l_5c_5 - l_6c_6 - l_9s_9 \\ l_1c_1 + l_2c_2 - l_8c_8 - l_7c_7 - l_9s_9 \\ l_1s_1 + l_2s_2 + l_9c_9 + l_6s_6 + l_5s_5 - L_{13} \\ l_1s_1 + l_2s_2 + l_9c_9 + l_8s_8 + l_7s_7 - L_{14} \\ l_1s_1 + l_2s_s + l_3s_3 + l_4s_4 - L_{12} \\ l_1c_1 + l_2c_2 - l_4c_4 - l_3c_3 \end{bmatrix} = \mathbf{0} \tag{6}$$

where $c_i := \cos(\theta_i)$, $s_i := \sin(\theta_i)$, and L_{1i} are the horizontal position of the first leg from the ith leg.

For the swing phase, the following Lagrange equation is used:

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}_i}\right) - \frac{\partial K}{\partial \theta_i} + \frac{\partial P}{\partial \theta_i} = T_i + \left(\frac{\partial \boldsymbol{\Phi}}{\partial \theta_i}\right)^T \boldsymbol{\lambda} \tag{7}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. Hence, the dynamic equation of the robot in this phase can be written as

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{G}(\boldsymbol{\theta}) = \mathbf{J}^T(\boldsymbol{\theta})\boldsymbol{\lambda} + \mathbf{T} \tag{8}$$

where $\mathbf{J}(\boldsymbol{\theta}) = \partial\boldsymbol{\Phi}/\partial\boldsymbol{\theta} \in \mathbb{R}^{6\times9}$ is the Jacobian matrix.

At the end of the swing phase, the tip of the swing leg contacts the ground surface with an impact. Immediately after the impact, the vertical velocity of the tip of the swing leg becomes zero due to the collision with the ground. The dynamic equation governing this impact can be represented as follows [19]:

$$\dot{\boldsymbol{\theta}}^+_{\text{impact}} = \dot{\boldsymbol{\theta}}^- + \mathbf{M}^{-1}\mathbf{J}^{\mathbf{T}}\left[\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{\mathbf{T}}\right]^{-1}\left(-\mathbf{J}\dot{\boldsymbol{\theta}}^-\right) \tag{9}$$

where $\dot{\boldsymbol{\theta}}^+_{\text{impact}}$ and $\dot{\boldsymbol{\theta}}^-$ are the 9×1 vectors of the joint velocities immediately after and before the impact, respectively. For brevity, the matrices and vectors in this section are not given in this manuscript. They can be easily derived by the user.

Figure 2a–c shows the quadruped robot in the support phase, the swing phase, and the impact phase, respectively.

In order to make the model closer to the real quadruped robots, DC servomotors are also included in the model of the robot as follows [20]:

$$\begin{aligned} \boldsymbol{\tau}_e &= \mathbf{K}_T\mathbf{i}_a \\ \boldsymbol{\tau}_e &= \mathbf{J}_m\ddot{\boldsymbol{\theta}}_m + \mathbf{B}_m\dot{\boldsymbol{\theta}}_m + \boldsymbol{\tau}_m \\ \mathbf{v}_t &= \mathbf{R}_a\mathbf{i}_a + \mathbf{L}_a\dot{\mathbf{i}}_a + \mathbf{K}_E\dot{\boldsymbol{\theta}}_m \end{aligned} \tag{10}$$

where $\theta_m \in R^{9\times1}$ is the vector of the rotation angles (i.e., the angles of the joints), $\mathbf{v}_t \in R^{9\times1}$ is vector of the input voltages, $\mathbf{i}_a \in R^{9\times1}$ is the vector of the armature currents, $\mathbf{L}_a \in R^{9\times9}$ is the matrix of the stator inductances, $\boldsymbol{\tau}_e$ is vector of electromagnetic torques, $\boldsymbol{\tau}_m$ is vector of servomotors' torques, $\mathbf{R}_a \in R^{9\times9}$ is the matrix of the stator resistances, $\mathbf{J}_m \in R^{9\times9}$ is the matrix of the motor inertia,$\mathbf{K}_T \in R^{9\times9}$ is the matrix of

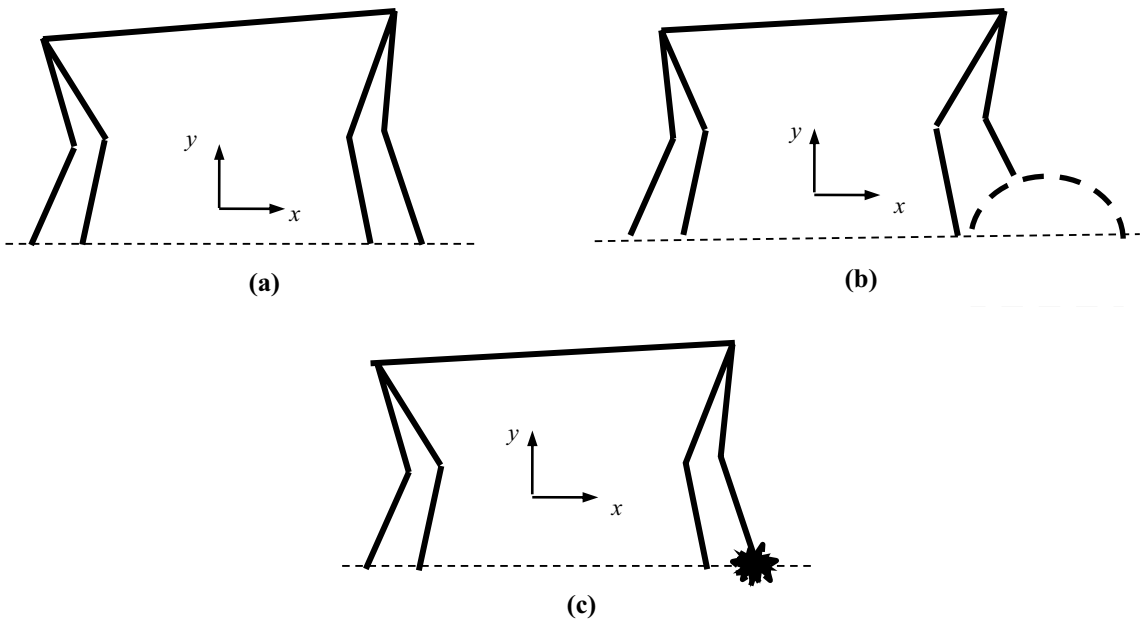

Fig. 2 Quadruped robot in: **a** support phase, **b** swing phase, **c** impact phase

torque constants, $\mathbf{K}_E \in R^{9\times9}$ is the matrix of the electrical power coefficients, $\mathbf{B}_m \in R^{9\times9}$ is the matrix of damping constants, and $w = \left\{ \frac{\theta_i}{\theta_{mi}} = \frac{\tau_{mi}}{\tau_i} \right\}$ is the gear ratio for the ith servomotor.

Since the inductance of the stator is usually very small, it can be ignored. In this case, the rotor currents can be expressed as

$$\mathbf{i}_a = \mathbf{R}_a^{-1}(\mathbf{v}_t - \mathbf{K}_E\dot{\boldsymbol{\theta}}_m). \tag{11}$$

Therefore, using (10) and (11), the motor dynamics can be written as follows:

$$\mathbf{J}_m\ddot{\boldsymbol{\theta}}_m + (\mathbf{B}_m + \mathbf{R}_a^{-1}\mathbf{K}_E\mathbf{K}_T)\boldsymbol{\theta}_m + \boldsymbol{\tau}_m = \mathbf{R}_a^{-1}\mathbf{K}_T\mathbf{v}_t. \tag{12}$$

Using the gear rations, $\boldsymbol{\theta}_m$ and $\boldsymbol{\tau}_m$ can be found and replaced in (12) as

$$\mathbf{J}_m\mathbf{W}^{-1}\ddot{\boldsymbol{\theta}}_m + (\mathbf{B}_m + \mathbf{R}_a^{-1}\mathbf{K}_E\mathbf{K}_T)\mathbf{R}^{-1}\boldsymbol{\theta}_m + \mathbf{W}\boldsymbol{\tau}_m = \mathbf{R}_a^{-1}\mathbf{K}_T\mathbf{v}_t. \tag{13}$$

Finding $\boldsymbol{\tau}_m$ from (13) and using the gear ration, $\boldsymbol{\tau}$ in (5) can be calculated. Then, the dynamic equations of the quadruped robot in the swing phase can be represented as follows:

$$\begin{aligned}\mathbf{J}_m\mathbf{W}^2\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + (\mathbf{B}_m + \mathbf{R}_a^{-1}\mathbf{K}_E\mathbf{K}_T)\boldsymbol{\theta} \\ + \mathbf{W}^2\mathbf{H}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \mathbf{W}^2\mathbf{G}(\boldsymbol{\theta}) = \mathbf{R}_a^{-1}\mathbf{K}_T\mathbf{v}_t\end{aligned} \tag{14}$$

And the dynamic equations of the supporting phase can be expressed as

$$\begin{aligned}(\mathbf{J}_m\mathbf{W}^2\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + (\mathbf{B}_m + \mathbf{R}_a^{-1}\mathbf{K}_E\mathbf{K}_T)\boldsymbol{\theta} + \mathbf{W}^2\mathbf{H}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \mathbf{W}^2\mathbf{G}(\boldsymbol{\theta}) \\ = \mathbf{W}^2\mathbf{J}^T(\boldsymbol{\theta})\boldsymbol{\lambda} + \mathbf{R}_a^{-1}\mathbf{K}_T\mathbf{v}_t\end{aligned} \tag{15}$$

The parameters of the servomotors that have been used in this paper are given in Table 1.

3 Path planning

For the robot motion, an appropriate path must be designed first. The method of polynomial functions is used here [19].

The following polynomial functions are considered for the robot's path:

$$x(t) = a_0 + a_1t + a_2t^2 + a_3t^3, \qquad 0 \le t \le T_s \tag{16}$$

$$y(t) = b_0 + b_1t + b_2t^2 + b_3t^3 + b_4t^4 + b_5t^5, \qquad 0 \le t \le T_s \tag{17}$$

where x and y are the coordinates of the desired path for the swing phase. The values of all coefficients can be found based on the following physical constraints of the robot:

$$y(0) = 0, \quad y(T_s) = 0 \tag{18}$$

$$\begin{aligned}\dot{y}(0) = 0, \quad \dot{y}(T_s) = 0 \\ \dot{x}(0) = 0, \quad \dot{x}(T_s) = 0\end{aligned} \tag{19}$$

$$y(T_m) = H_m, \quad \dot{y}(T_m) = 0 \tag{20}$$

$$x(0) = -\frac{S_L}{2}, \quad x(T_s) = \frac{S_L}{2} \tag{21}$$

where S_L is the step length, T_s is the time for one complete step, and H_m is the maximum height of the robot's leg that occurs at $t = T_m$. The desired path of the foot tip of the robot is shown in Fig. 3.

Since the robot's hip moves slower and has smaller variations, as compared to the legs, third-order polynomials are used for the hip as

$$\text{Swing Phase:} \quad \begin{cases} x_{h\ \text{swing}}(t) = c_0 + c_1t + c_2t^2 + c_3t^3, & 0 \le t \le T_s \\ y_{h\ \text{swing}}(t) = y_h(t), & 0 \le t \le T_s \end{cases} \tag{22}$$

$$\text{Support Phase:} \quad \begin{cases} x_{h\ \text{sup}} = d_0 + d_1t + d_2t^2 + d_3t^3, & 0 \le t \le T_{\text{sup}} \\ y_{h\ \text{sup}}(t) = y_h(t), & 0 \le t \le T_{\text{sup}} \end{cases} \tag{23}$$

where $x_{h\ \text{swing}}$ and $x_{h\ \text{sup}}$ are the swing and support movements of the robot's hip along the x-axis, respectively; similarly, $y_{h\ \text{swing}}$ and $y_{h\ \text{sup}}$ are the swing and support movements of the robot's hip along the y axis, respectively; T_{sup} is the period time of the support phase and T_s is the period time for taking one complete step (the swing phase).

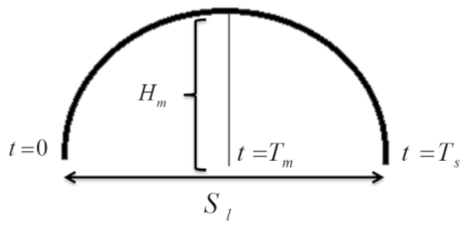

Fig. 3 Desired path of foot tip

Table 1 Servomotors parameters

Servomotors	R_a	K_E	K_T	B_m	J_m	W	V_t
1,…, 9	5.3	0.28	0.48	64×10^{-4}	0.51	1:10	48

To obtain the coefficients in (16) and (17), the following physical constraints are used:

1. Vertical motion of the hip

$$y_{h\ \mathrm{swing}}(t) = H_h,\ y_{h\ \mathrm{sup}}(t) = H_h \tag{24}$$

2. Repeatability of the gait

$$x_{h\ \mathrm{swing}}(0) = -S_{\mathrm{swing}_0},\ \dot{x}_{h\ \mathrm{swing}}(0) = v_{h_1} \tag{25}$$

$$x_{h\ \mathrm{sup}}(T_{\mathrm{sup}}) = \frac{S_L}{2} - S_{\mathrm{swing}_0},\ \dot{x}_{h\ \mathrm{sup}}(T_{\mathrm{sup}}) = v_{h_1} \tag{26}$$

3. Continuity of the gait

$$x_{h\ \mathrm{swing}}(T_s) = S_{\mathrm{sup}_0},\ \dot{x}_{h\ \mathrm{sup}}(T_s) = v_{h_2} \tag{27}$$

$$x_{h\ \mathrm{sup}}(0) = S_{\mathrm{sup}_0},\ \dot{x}_{h\ \mathrm{sup}}(0) = v_{h_2} \tag{28}$$

where v_{h_1} and v_{h_2} are the hip velocity at the beginning and at the end of each step in the swing phase and vice versa for the support phase, respectively, S_{swing_0} and S_{sup_0} are the positions of the hip at the beginning of the swing and support phases, respectively, H_h is the height of the hip, and T_{sup} is the period time of the support phase.

Using the kinematic equations of the robot ("Appendix"), the desired path for all joints of the robot for four gaits (i.e., every leg of the robot takes one step) can be obtained.

Initially, the robot is in the support phase. Then, the swing phase occurs. When the robot's leg touches the ground, the swing phase ends and the impact occurs, followed by the support phase. In this way, one step is taken. The order of robot's legs is as follows: left back, left front, right back, and right front.

It should be noted that in the backstepping control method, stability of the closed-loop system is insured because of being model-based method, when the disturbances and uncertainties are not taken into account. In order to improve these weaknesses of the backstepping control method, fuzzy systems are employed to adapt the feedback coefficients. Hence, the proposed method is robust against uncertainties in the robot's parameters and external disturbances since the design of the backstepping controller does not consider the uncertainties. Therefore, the proposed method can cope with uncertainties in the robot's parameters, is robust against external disturbances, and can perform well when measurement noises exist.

In the following sections, first the backstepping controller is explained followed by the fuzzy systems that are used to tune the parameters of the backstepping controller.

4.1 Backstepping approach

The backstepping control method is a Lyapunov function-based strategy that has been introduced in 1992 by Kokotovic [21]. The main idea of this method is to design a recursive controller such that the control laws are designed by considering the state variables as the virtual controls. This method has two main goals: 1) stabilization and 2) tracking.

In order to design this controller, consider the state-space variables of system (14) as:

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \end{bmatrix} \tag{29}$$

Therefore, the state-space equations of the system can be written as

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_2 \\ \left(\mathbf{J}_m \mathbf{W}^2 \mathbf{M}(\mathbf{x}_1)\right)^{-1} \left(\mathbf{R}_a^{-1} \mathbf{K}_T \mathbf{v}_t - \left(\mathbf{B}_m + \mathbf{R}_a^{-1} \mathbf{K}_E \mathbf{K}_T\right) \mathbf{x}_1 - \mathbf{W}^2 \mathbf{H}(\mathbf{x}_1, \mathbf{x}_2) - \mathbf{W}^2 \mathbf{G}(\mathbf{x}_1)\right) \end{bmatrix} \tag{30}$$

4 Fuzzy-backstepping controller

The fuzzy-backstepping controller is proposed in this paper for the tracking control of the quadruped robot. In the followings, first the backstepping method is described.

The reason for opting the backstepping controller is to obtain a stable walking of the robot, since this controller is inherently stable method due to the design method, which is based on Lyapunov stability theory. This method is very suitable for path tracing. Moreover, based on the complex dynamic equations of the robot, the goal is to develop a controller that is not model-based and does not require linearization. Hence, the backstepping control method is a proper choice that does not require linearization.

where the output of the system is defined by $\mathbf{y} = \mathbf{x}_1 = \boldsymbol{\theta}$. Next, consider the following virtual state variables:

$$\mathbf{z}_1 = \mathbf{x}_1 - \mathbf{x}_d \tag{31}$$

$$\mathbf{z}_2 = \mathbf{x}_2 - \boldsymbol{\alpha} \tag{32}$$

where $\mathbf{x}_d$ is the desired output. In this step, the aim is to design the virtual control law $\boldsymbol{\alpha}$ such that $\mathbf{z}_2$ converges to zero. In order to design the control law based on the Lyapunov method, consider the positive-definite Lyapunov function as

$$V_1 = \tfrac{1}{2}\mathbf{z}_1^T\mathbf{z}_1. \tag{33}$$

Substituting the derivative of (31) into the derivative of the Lyapunov function (33), it yields

$$\dot{V}_1 = \mathbf{z}_1^T\dot{\mathbf{z}}_1 = \mathbf{z}_1^T(\mathbf{z}_2 + \boldsymbol{\alpha} - \dot{\mathbf{x}}_d) \tag{34}$$

Now, $\boldsymbol{\alpha}$ must be specified such that the first-order system is stable. For this reason, consider

$$\boldsymbol{\alpha} = -k_1\mathbf{z}_1 + \dot{\mathbf{x}}_d \tag{35}$$

where k_1 is a positive constant. Therefore, (34) can be written as

$$\dot{V}_1 = -k_1\mathbf{z}_1^T\mathbf{z}_1 + \mathbf{z}_1^T\mathbf{z}_2. \tag{36}$$

When $\mathbf{z}_2 \to 0$, it yields $\dot{V}_1 = -k_1\mathbf{z}_1^T\mathbf{z}_1$, which guarantees asymptotic stability of the system. Next, substituting (30) into the derivative of (32) gives

$$\dot{\mathbf{z}}_2 = \dot{\mathbf{x}}_2 - \dot{\boldsymbol{\alpha}}_1 = (\mathbf{J}_m\mathbf{W}^2\mathbf{M}(\mathbf{x}_1))^{-1}\left(\mathbf{R}_a^{-1}\mathbf{K}_T\mathbf{v}_t - (\mathbf{B}_m + \mathbf{R}_a^{-1}\mathbf{K}_E\mathbf{K}_T)\mathbf{x}_1 - \mathbf{W}^2\mathbf{H}(\mathbf{x}_1, \mathbf{x}_2) - \mathbf{W}^2\mathbf{G}(\mathbf{x}_1)\right) - \dot{\boldsymbol{\alpha}} \tag{37}$$

Next, consider the second Lyapunov function as

$$V_2 = V_1 + \tfrac{1}{2}\mathbf{z}_2^T\mathbf{z}_2 \tag{38}$$

Substituting (36) and (37) into the derivative of (38) yields

$$\dot{V}_2 = \mathbf{z}_2^T\left[(\mathbf{J}_m\mathbf{W}^2\mathbf{M}(\mathbf{x}_1))^{-1}\left(\mathbf{R}_a^{-1}\mathbf{K}_T\mathbf{v}_t - (\mathbf{B}_m + \mathbf{R}_a^{-1}\mathbf{K}_E\mathbf{K}_T)\mathbf{x}_1 - \mathbf{W}^2\mathbf{H}(\mathbf{x}_1, \mathbf{x}_2) - \mathbf{W}^2\mathbf{G}(\mathbf{x}_1)\right) - \dot{\boldsymbol{\alpha}} - k_1\mathbf{z}_1^T\mathbf{z}_1\right] \tag{39}$$

The control law can be extracted from (39) as

$$\mathbf{v}_t = \mathbf{R}_a\mathbf{K}_T^{-1}[(\mathbf{B}_m + \mathbf{R}_a^{-1}\mathbf{K}_E\mathbf{K}_T)\mathbf{x}_1 + \mathbf{W}^2\mathbf{H}(\mathbf{x}_1, \mathbf{x}_2) + \mathbf{W}^2\mathbf{G}(\mathbf{x}_1) + \mathbf{J}_m\mathbf{W}^2\mathbf{M}(\mathbf{x}_1)(\dot{\boldsymbol{\alpha}} - k_2\mathbf{z}_2)] \tag{40}$$

where k_2 is a positive constant. Finally

$$\dot{V}_2 = -k_1\mathbf{z}_1^T\mathbf{z}_1 - k_2\mathbf{z}_2^T\mathbf{z}_2 \tag{41}$$

Since (36) and (41) are negative, the controller is globally asymptotically stable and trajectories tracking will be carried out.

One of the challenges in the backstepping controller is to select coefficients k_1 and k_2 appropriately; otherwise performance of the closed-loop system may not be acceptable. Selecting constant values for these coefficients may yield good response of the system. However, when there are uncertainties or changes in the system parameters, these coefficients must change accordingly. To obtain adaptive values for these coefficients, fuzzy systems are used in this manuscript. In other word, using appropriate fuzzy IF–THEN rules, the coefficients of the backstepping controller are adaptively changed such that the closed-loop system copes with uncertainties in the system parameters as well as with the external disturbances and measurement noises.

4.2 Fuzzy system

Fuzzy systems are knowledge- or rules-based systems. The main part of a fuzzy system is its rule base that is composed of IF–THEN statements. The crisp inputs are converted into fuzzy inputs by the fuzzifier to be used by the inference engine. In contrast, converting the fuzzy results from the inference engine into crisp values is performed by the defuzzifier. Indeed, the fuzzy inference engine decides how the rules in the knowledge base operate by using the fuzzy inputs from the fuzzifier.

In this paper, a fuzzy system is designed for adapting the coefficients k_1 and k_2 based on the operating point of the robot. Since fuzzy systems have only one output, two fuzzy systems are needed for k_1 and k_2, respectively. The quadruped robot has nine DoF. Hence, 18 fuzzy systems are needed for adjusting the backstepping coefficients. The inputs to the fuzzy systems are the tracking error and its derivative. The outputs are $k_1(t)$ and $k_2(t)$, respectively (Fig. 4). The fuzzy systems are composed of Mamdani product inference engine, singleton fuzzifier, center average defuzzifier, and triangular membership functions for the input variables. The output equation of these fuzzy systems can be written as [22]

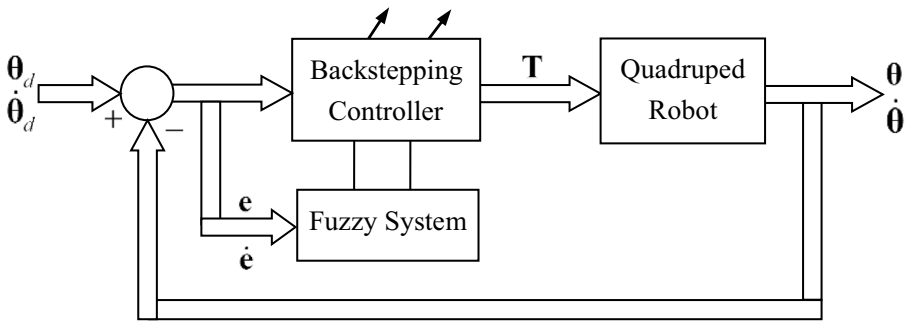

Fig. 4 Backstepping controller block diagram

$$k_{ji}(t) = f(e_i, \dot{e}_i) = \frac{\sum_{l=1}^{M} \bar{y}^l \left(\mu_{A^l}(e_i)\mu_{B^l}(\dot{e}_i)\right)}{\sum_{l=1}^{M} \left(\mu_{A^l}(e_i)\mu_{B^l}(\dot{e}_i)\right)}, \quad (j = 1, 2, \quad i = 1, \ldots, 9) \tag{42}$$

where $\mu_{A^l}(e_i)$ and $\mu_{B^l}(\dot{e}_i)$ are the membership function of the premise variables (e_i and $\dot{e}_i$) for the lth rule and the ith input, $\bar{y}^l$ is the center of the membership function of the consequent part of the fuzzy rules, and M is the number of fuzzy rules. The membership functions for the inputs (e_1 and $\dot{e}_1$) and the output ($k_{ji}(t)$) of the fuzzy systems are shown in Fig. 5. The maximum and the minimum values of the input variables (e_1 and $\dot{e}_1$) are obtained from the knowledge of the robot's behavior. The fuzzy IF–THEN rules for the backstepping coefficients $k_1(t)$ and $k_2(t)$ are given in Tables 2 and 3, respectively. These rules are obtained using some experiences from the operation of the robot. Table 4 shows the maximum values for these coefficients. These values are needed to denormalize the output of the fuzzy systems back to their actual values.

5 Stability analysis

The stability is one of the most important and difficult issues in the field of legged robots. These systems do not satisfy the Lipschitz condition, mainly because their motion is periodic. Moreover, the Lyapunov stability analysis does not provide global stability for periodic systems. The Poincaré return map is a useful method for stability analysis of the periodic systems. The advantage of this method is that it reduces the study of the periodic orbits to the study of equilibrium points (or fixed points). However, the most important issue in this

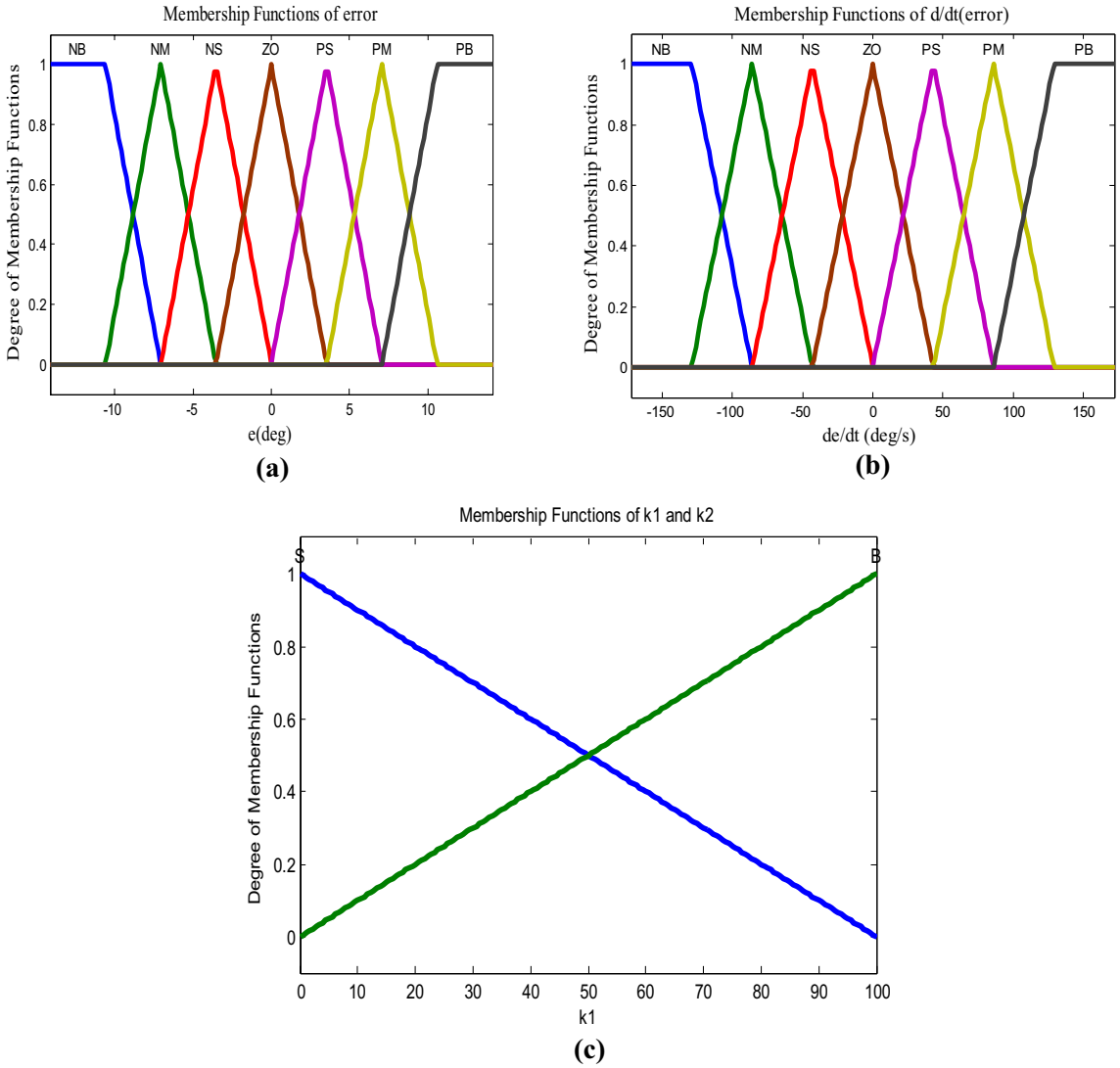

Fig. 5 Membership functions for inputs e (**a**) and $\dot{e}$ (**b**) and output (**c**) of fuzzy system

Table 2 Fuzzy rules for k_1

$e(t)$	$\dot{e}(t)$						
	NB	NM	NS	ZO	PS	PM	PB
NB	B	B	B	B	B	B	B
NM	S	B	B	B	B	B	S
NS	S	S	B	B	B	S	S
ZO	S	S	S	B	S	S	S
PS	S	S	B	B	B	S	S
PM	S	B	B	B	B	B	S
PB	B	B	B	B	B	B	B

Table 3 Fuzzy rules for k_2

$e(t)$	$\dot{e}(t)$						
	NB	NM	NS	ZO	PS	PM	PB
NB	S	S	S	S	S	S	S
NM	B	S	S	S	S	S	B
NS	B	B	S	S	S	B	B
ZO	B	B	B	S	B	B	B
PS	B	B	S	S	S	B	B
PM	B	S	S	S	S	S	B
PB	S	S	S	S	S	S	S

Table 4 Max value of backstepping coefficients

k_1	Max	k_2	Max
k_{11}	300	k_{21}	50
k_{12}	160	k_{22}	40
k_{13}	160	k_{23}	35
k_{14}	300	k_{24}	44
k_{15}	300	k_{25}	50
k_{16}	160	k_{26}	40
k_{17}	160	k_{27}	40
k_{18}	300	k_{28}	50
k_{19}	180	k_{29}	10

method is to find the Poincaré map, which is nonlinear and complicated. For typical nonlinear systems, it is almost impossible to obtain this nonlinear map in a closed form because it requires the solution of a nonlinear differential equation. Hence, numerical methods are used to compute the return map, find its fixed points, and estimate the eigenvalues for analyzing the stability of the closed-loop system. However, the numerical computations are usually time-consuming especially when they must be solved iteratively at every sampling time. Moreover, numerical computations are not insightful in the sense that it is often difficult to establish a direct relationship between the parameters that a designer can change in a system and the stability properties of the fixed point of the Poincaré map. One of the methods that is used for this problem is linearization of the Poincaré map around the fixed point and finding the Jacobian matrix. If the eigenvalues of this matrix are inside the unit circle, the stability of the robot is guaranteed.

5.1 Systems with impulse effect

Systems with impulse effects are used to model the inherently hybrid nature of the walking robots, which is explained in Sect. 2. Systems with impulse effects have a continuous phase described by differential equations and a discrete phase described by an instantaneous state-reset event. A general periodic time-invariant control system with impulse effect has the following form [23]:

$$\Gamma : \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} & \mathbf{x}^- \notin S \\ \mathbf{x}^+ = \boldsymbol{\Delta}(\mathbf{x}^-) & \mathbf{x}^- \in S \end{cases} \tag{43}$$

where $\mathbf{x} \in \chi$, and $\mathbf{x}^-$ and $\mathbf{x}^+$ are the left and right limits of the solution of the system (i.e., $\mathbf{x}^-$ and $\mathbf{x}^+$ are the state of the robot right before and after the impact, respectively), $\mathbf{u}$ is the control input, and S is the impact (or switching) surface, represented as

$$S = \left\{ \mathbf{x} \in \chi \,\middle|\, H(x) = 0, H_0(x) > 0 \right\} \tag{44}$$

in which $H : \forall \mathbf{x} \in S,\ S \neq \emptyset$ and $\frac{\partial H}{\partial x}(\mathbf{x}) \neq 0$. The solution of a system with impulse effects is denoted as $\phi(t, t_0, \mathbf{x_0})$ for $t > t_0$ and $\mathbf{x_0} \in X$.

For more details on application of the Poincaré map for periodic locomotion with impact effects, the reader may refer to [23, 24].

5.2 Quadruped robot dynamic as a system with impact effects

For stability analysis, the swing and impact dynamics are considered because the support phase is relatively more stable and hence does not incur stability challenges as compared to the other two phases. The swing phase dynamic was described in (14) as

$$\begin{aligned}&(\mathbf{J}_m\mathbf{W}^2\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + (\mathbf{B}_m + \mathbf{R}_a^{-1}\mathbf{K}_E\mathbf{K}_T)\boldsymbol{\theta}\\ &\quad + \mathbf{W}^2\mathbf{H}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \mathbf{W}^2\mathbf{G}(\boldsymbol{\theta}) = \mathbf{R}_a^{-1}\mathbf{K}_T\mathbf{v}_t\end{aligned} \tag{45}$$

where $\mathbf{v}_t$ can be calculated using (40). Hence, the state-space form of the swing phase is

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ (\mathbf{J}_m\mathbf{W}^2\mathbf{M}(\boldsymbol{\theta}))^{-1}\left(\mathbf{R}_a^{-1}\mathbf{K}_T\mathbf{v}_t - (\mathbf{B}_m + \mathbf{R}_a^{-1}\mathbf{K}_E\mathbf{K}_T)\mathbf{x}_1 - \mathbf{W}^2\mathbf{H}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) - \mathbf{W}^2\mathbf{G}(\boldsymbol{\theta})\right) \end{bmatrix} \tag{46}$$

where $\mathbf{x} = [\boldsymbol{\theta}^T \ \dot{\boldsymbol{\theta}}^T]^T$. Using these new variables and based on (9), the impact equation is

$$\dot{\mathbf{x}}^+ = \dot{\mathbf{x}}^- + \mathbf{M^{-1}J^T}\left[\mathbf{JM^{-1}J^T}\right]^{-1}(-\mathbf{J}\dot{\mathbf{x}}^-). \tag{47}$$

Therefore,

$$\mathbf{x}^+ = \begin{bmatrix} \boldsymbol{\Delta}(\mathbf{x}^-) \\ \boldsymbol{\Delta}(\mathbf{x}^-,\dot{\mathbf{x}}^-) \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \boldsymbol{\Delta}(\mathbf{x}^-,\dot{\mathbf{x}}^-) \end{bmatrix} \tag{48}$$

where $\boldsymbol{\Delta}(\mathbf{x}^-,\dot{\mathbf{x}}^-)$ is the right-hand side of (47). Note that $\boldsymbol{\Delta}(\mathbf{x}^-)$ is equal to one because it is assumed that the impact just affects the angular velocities and not the joint angles. The impact or the switching surface occurs when at the end of the swing phase the moving leg touches the ground, which means that the height of the tip of the swinging leg becomes zero. That is

$$S = \left\{ \mathbf{x} \middle| \, y_{\text{swing}}(\mathbf{x}) = 0, \ x_{\text{swing}}(\mathbf{x}) > 0 \right\} \tag{49}$$

where $[x_{\text{swing}}(\mathbf{x}), y_{\text{swing}}(\mathbf{x})]^T$ is the coordinate of the tip of the swinging leg.

5.3 Fixed point

In the sense of the Poincaré map, a periodic system is stable when it reaches a special point after some cycles with any initial condition. This point is called the "fixed point," because it is the same at the end of every cycle. As it was mentioned before, the Poincaré map has a simple idea. However, its implementation to nonlinear and complicated system (e.g., legged robots) is hindered by finding a closed-form solution to the nonlinear dynamic equations governing the system. One method is to linearize the Poincaré map around the operating point. For linearizing the Poincaré map, a fixed point must be found first. This can be done by linearizing the state equations just before the impact occurs [23].

5.4 Jacobian matrix

The restricted Poincaré map $\mathbf{P}(\mathbf{x})$ induces a discrete-time system as

$$\mathbf{x}^{k+1} = \mathbf{P}(\mathbf{x}^k) \tag{50}$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{x}^- \\ \dot{\mathbf{x}}^- + \mathbf{M}^{-1}\mathbf{J}^T[\mathbf{JM}^{-1}\mathbf{J}^T]^{-1}(-\mathbf{J}\dot{\mathbf{x}}^-) \end{bmatrix} \tag{51}$$

The linearization around the fixed point $\mathbf{x}^* = [(\mathbf{x}^*)^T \ (\dot{\mathbf{x}}^*)^T]^T = [\boldsymbol{\theta}^{-*} \ \boldsymbol{\theta}'^{-*}]^T$, which is the point that usually is obtained from a predetermined reference path and is generally chosen at the moment right before the impact, determines the stability of the closed-loop system. Let us define $\delta\mathbf{x}^k = \mathbf{x}^k - \mathbf{x}^*$. The Poincaré map linearized around the fixed point $\mathbf{x}^*$ yields the following linear system:

$$\delta\mathbf{x}^{k+1} = \mathbf{A}\delta\mathbf{x}^k \tag{52}$$

where the $\mathbf{A}$ is an 18×18 square matrix that is called the Jacobian of the Poincaré map whose columns are the fix point vectors (position and velocity of 9 joints of the robot) and is computed as follows:

$$\mathbf{A} = \left[\mathbf{A}_1 \ \mathbf{A}_2 \ \cdots \ \mathbf{A}_{18}\right]_{18\times 18} \tag{53}$$

where

$$\mathbf{A}_i = \frac{\mathbf{P}(\mathbf{x}^* + \boldsymbol{\Delta}\mathbf{x}_i) - \mathbf{P}(\mathbf{x}^* - \boldsymbol{\Delta}\mathbf{x}_i)}{2\Delta\mathbf{x}_i} \quad i = 1, \ldots, 18 \tag{54}$$

in which $\Delta x_i = \Delta\theta_i$ for $i = 1, \ldots, 9$ and $\Delta x_i = \Delta\dot{\theta}_i$ for $i = 10, \ldots, 18$; Δx_i are small perturbations introduced to calculate the linearized model. Hence, (54) should be interpreted as the scalar perturbation used in computing Δx_i. Generally, the calculation of the Jacobian matrix is sensitive to the amplitude of perturbation Δx_i. A fixed point of the restricted Poincaré map is locally exponentially stable if and only if the eigenvalues of $\mathbf{A}$ have magnitude strictly less than one [25].

6 Simulations

In this section, the simulation results of the proposed fuzzy-backstepping controller for the quadruped robot are presented. For comparison and investigating the effect of the fuzzy system on updating the parameters of the backstepping control to cope with uncertainties, external disturbances and noises, both methods have been applied to the quadruped robot.

The physical parameters of the quadruped robot are given in Table 5.

Figure 6 shows adaptation of k_{18} and k_{28} for forward right leg of the robot. Other coefficients have similar changes during the motion of the robot. The tracking of the desired paths for θ_3,θ_4, θ_5, and θ_6 is shown in Fig. 7. As this figure shows, the tracking errors in both fuzzy-backstepping and backstepping method are very small and both have almost the same performance in tracking desired path without considering the uncertainties, disturbances, and noises. Also, the angles of the robot are changing smoothly. The horizontal position of the CoM and the displacement of the legs along the x-axis are shown in Fig. 8a, b, respectively. The horizontal position of the CoM starts from 0.4 m initially and increases continuously as the robot moves forward. This means that the robot's movement is very smooth and continuous. Figure 8b shows the cycle of motion of four legs of the robot. In every step, one leg (the swing leg) moves forward, while the other legs (the support legs) are on the ground. Figure 9 shows the angular velocity of the joints. This figure also confirms soft movement of the robot that requires less energy. Figure 10 illustrates the walking of the robot in 3D for better visualization.

Table 5 Parameters of quadruped robot

Mass of the link (kg)		Length of the link (m)	
m_1	3	l_1	0.7
m_2	7	l_2	0.5
m_3	3	l_3	0.5
m_4	7	l_4	0.7
m_5	3	l_5	0.7
m_6	7	l_6	0.5
m_7	7	l_7	0.5
m_8	7	l_8	0.7
m_9	10	l_9	1

The phase planes of the first and fourth joints of the robot are shown in Fig. 11. As this figure shows, after passing the transient state (that is shown with dashed line), the paths are periodic. Other joints of the robot have similar periodic paths in their phase plane. Hence, the linearized Poincaré map in (52) can be used to check the stability of the closed-loop system. The eigenvalues of matrix **A** using the proposed fuzzy-backstepping controller are as follows:

$$\lambda_1 = -0.696,\ \lambda_{2,3} = 0.283 \pm j0.133,\ \lambda_4 = 0.088,$$
$$\lambda_{5,6} = -0.020 \pm j0.020,\ \lambda_7 = j0.011$$
$$\lambda_8 = 0.013,\ \lambda_{9,10} = 0.002 \pm j0.002,\ \lambda_{11} = 0.001,$$
$$\lambda_{12} = 0.500,\ \lambda_{13} = 0.500,\ \lambda_{14} = 0.500$$
$$\lambda_{15,16} = 0.500 \pm j0.000,\ \lambda_{17,18} = 0.500$$

These eigenvalues are shown in Fig. 12. As can be seen, all eigenvalues of the linearized Poincaré map are located well within the unit circle, which indicates a stable walking of the quadruped robot.

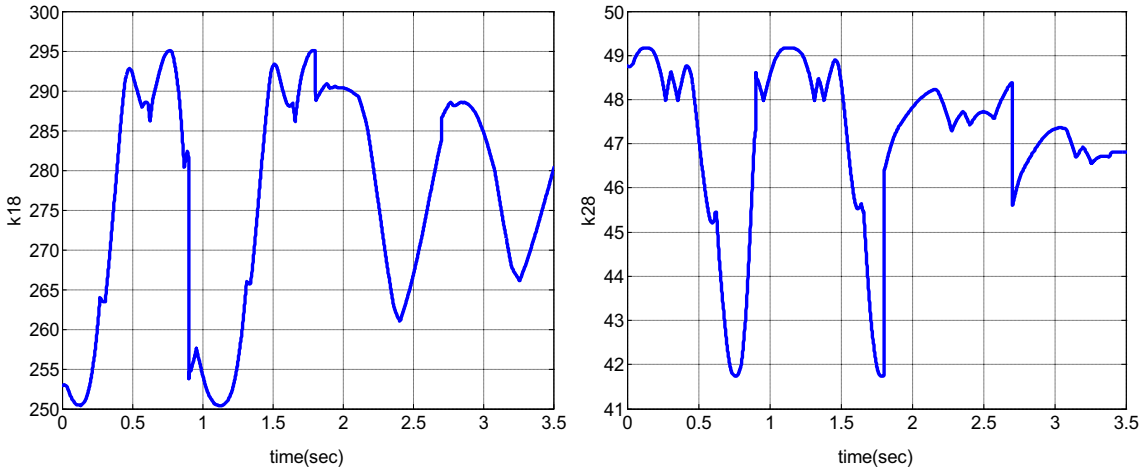

Fig. 6 Adaptation of coefficients k_{18} and k_{28} (forward right leg)

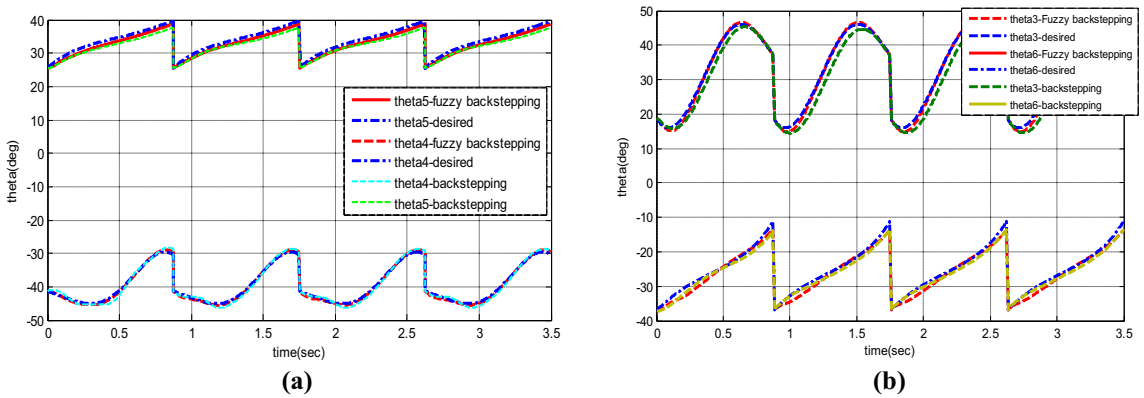

Fig. 7 Tracking of desired path for **a** θ_4 and θ_5, **b** θ_3 and θ_6

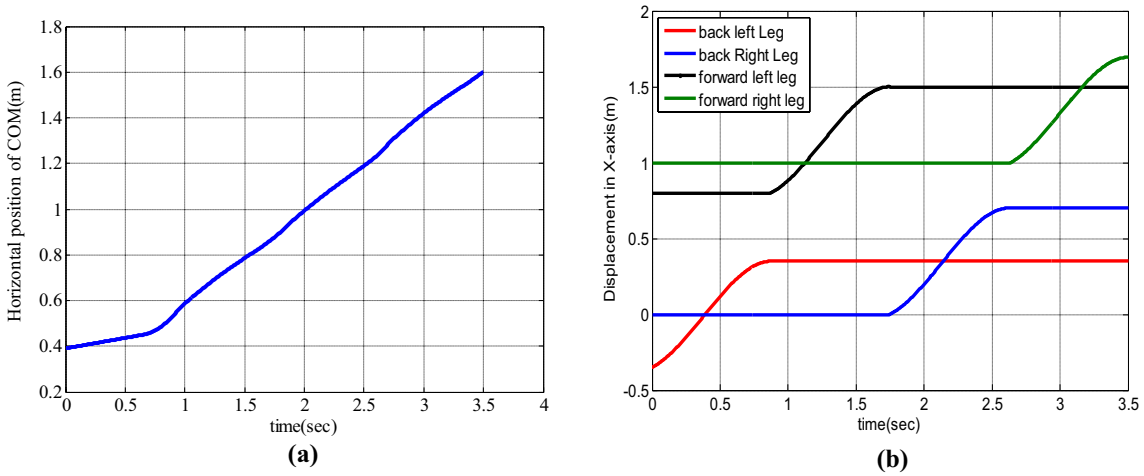

Fig. 8 **a** horizontal of position of CoM, **b** displacement of legs in x-axis

6.1 Effects of uncertainties, external disturbances, and measurement noises

The effects of external disturbances, uncertainties in the robot's parameters, and the measurement noises are investigated in the followings.

Figure 13 shows performance of the proposed method when the body mass of the robot is increased by 40% at $t = 3$ sec.. This is the maximum amount of body mass uncertainty that the proposed controller can handle. The red line in Fig. 13 shows the stability region of the robot. As this figure shows, the ZMP of the robot is within the stability region, yielding a stable walk. Figure 14 confirms that by increasing the body mass to 42%, some eigenvalues of matrix **A** fall outside the unite circle, which is equivalent to an unstable walk of the robot. In this case, the ZMP also yields an unstable walk of the robot, but for brevity it is not shown.

Next, an external disturbance with the amplitude of 200NM for 0.15 s is applied to the upper body of the robot (Fig. 15). Although this external disturbance causes the ZMP of the robot to go out of the stable region for a short time, the controller is able to regain stability of the robot by adapting the backstepping coefficients. For brevity, only one plot for each case is shown. Figure 16 shows that applying 210 NM as the external disturbance makes the robot unstable. Figure 17a, b illustrates the 3D visualization of the robot's walk for Figs. 14 and 16, respectively. The robot with red lines shows the moment that the robot becomes unstable.

Measurement noises inherently exist in all practical applications. Figure 18a, b shows performance of the proposed method when a white noise with 8% SNR is applied to the

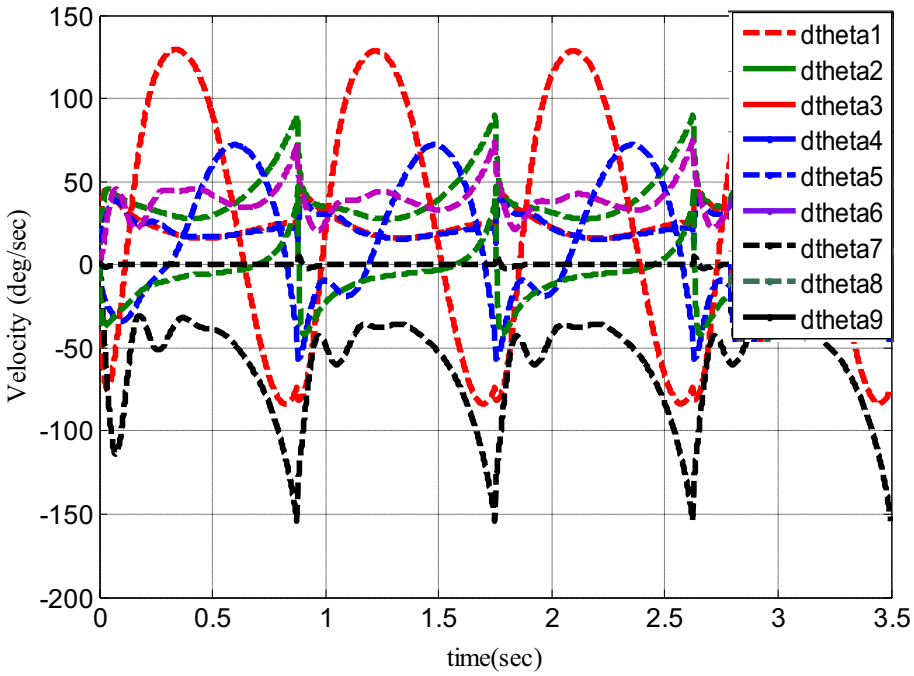

Fig. 9 Angular velocities

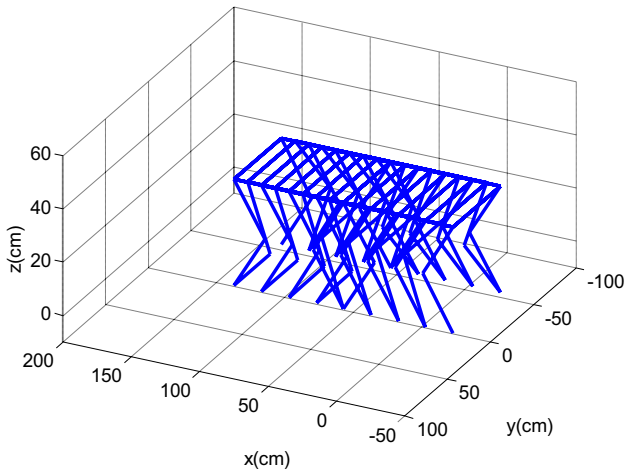

Fig. 10 A cycle of robot's walking in 3D

body of the robot. As these figures show, the fuzzy-backstepping controller not only provides stable walking of the robot but also reduces the measurement noise considerably.

In order to show the performance of the backstepping-alone controller, the followings are applied to the robot. First, the mass of robot's body is increased by 10%. Second, an external disturbance of 50 Nm in 2 s is applied to the body of the robot. Third, measurement noise with 8% SNR is applied to the body of the robot. Figure 19 shows the results. As this figure indicates, the robot cannot follow the desired path and becomes unstable for small uncertainties in the mass of the robot, minor external disturbances, and moderate measurement noise. This figure clearly indicates how vital the adaptation of the feedback gains in the backstepping controller is for good performance of the quadruped robot.

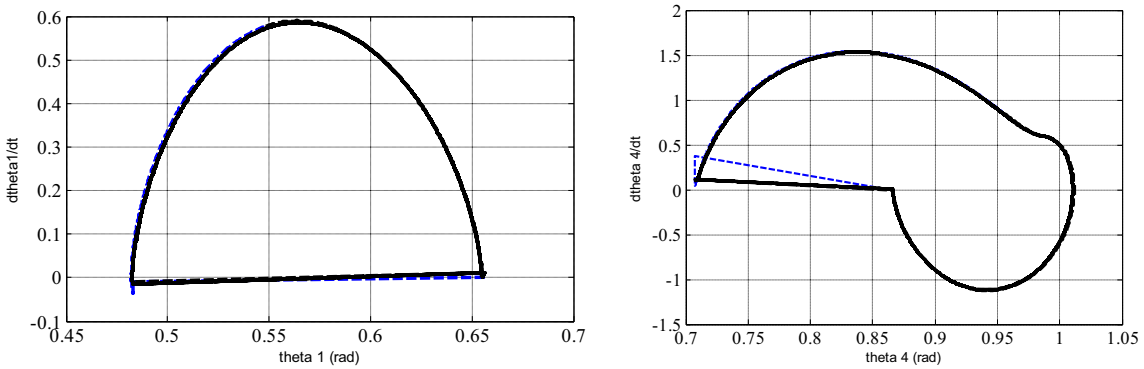

Fig. 11 Phase plane of the first and fourth joints

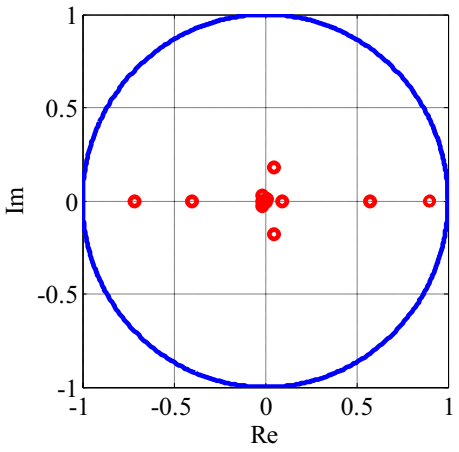

Fig. 12 Eigenvalues of matrix **A**

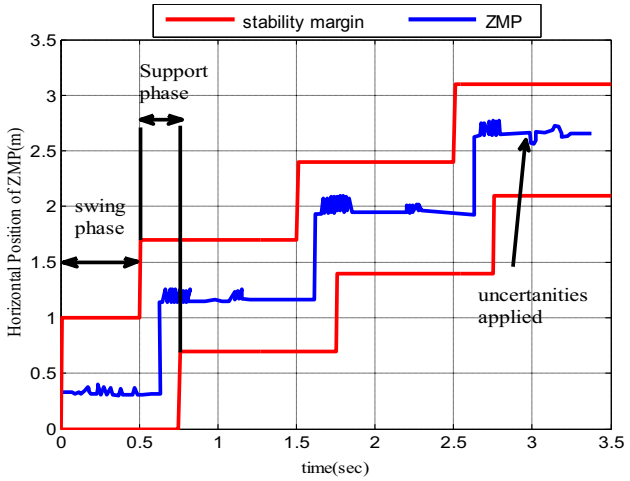

Fig. 13 ZMP of robot when 40% increase in the body mass is applied

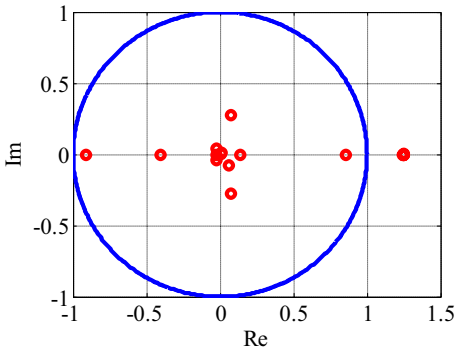

Fig. 14 Eigenvalues of a when 42% increase in the body mass is applied

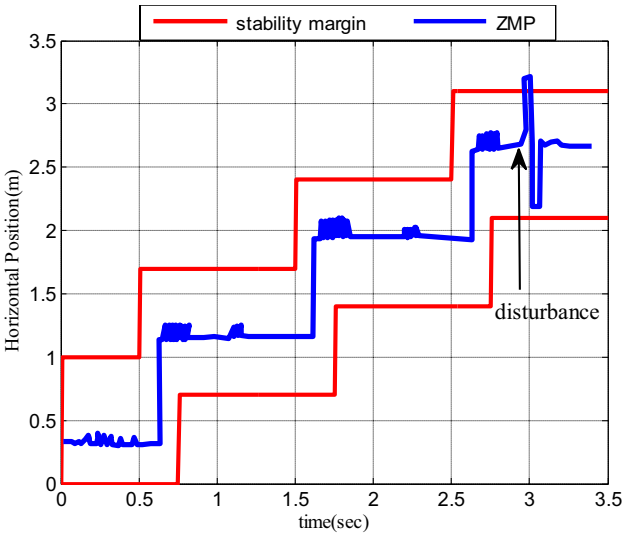

Fig. 15 ZMP of robot when 200 NM external disturbance is applied

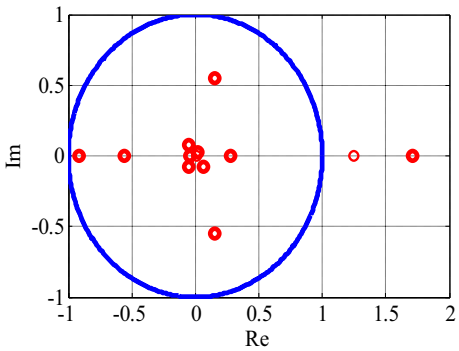

Fig. 16 Eigenvalues of a when 210 NM external disturbance is applied

7 Conclusion

In this paper, fuzzy-backstepping control method was proposed to control the quadruped robot's walking in the sagittal plane. Although the backstepping controller is essentially a stable method, fine-tuning its coefficients is not easy, especially when the controller is applied to highly nonlinear MIMO systems with strong interactions between states, such as quadruped robots. Hence, fuzzy systems were employed to define these coefficients adaptively. Another advantage of this adaptive method is that it can cope very well against uncertainties in the robot's

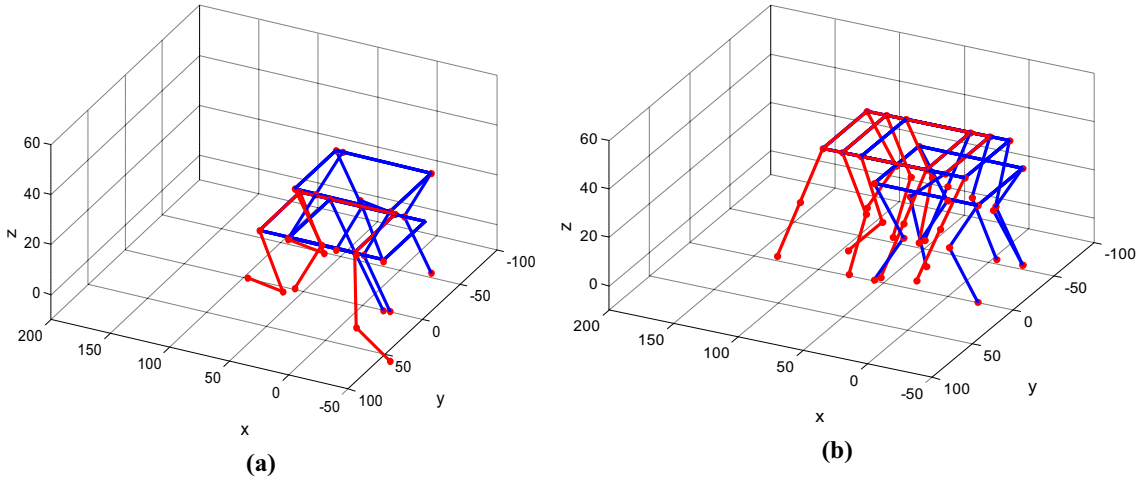

Fig. 17 3D plot of robot's walking when **a** 42% increase in the body mass is applied, **b** 210 NM external disturbance in applied

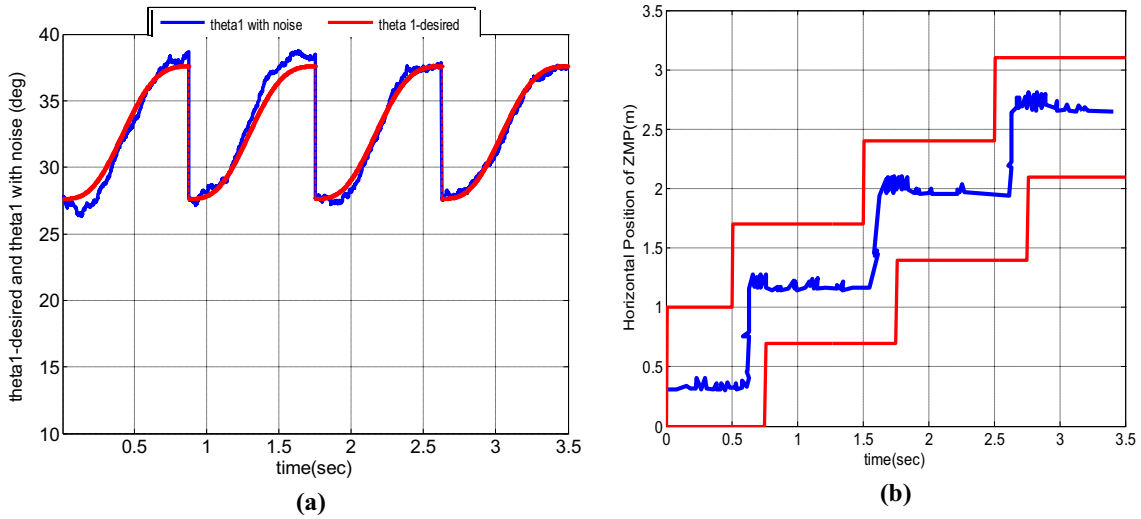

Fig. 18 **a** Tracking of rear joint ankle (θ_1) and **b** horizontal position of ZMP, when measurement noise is added

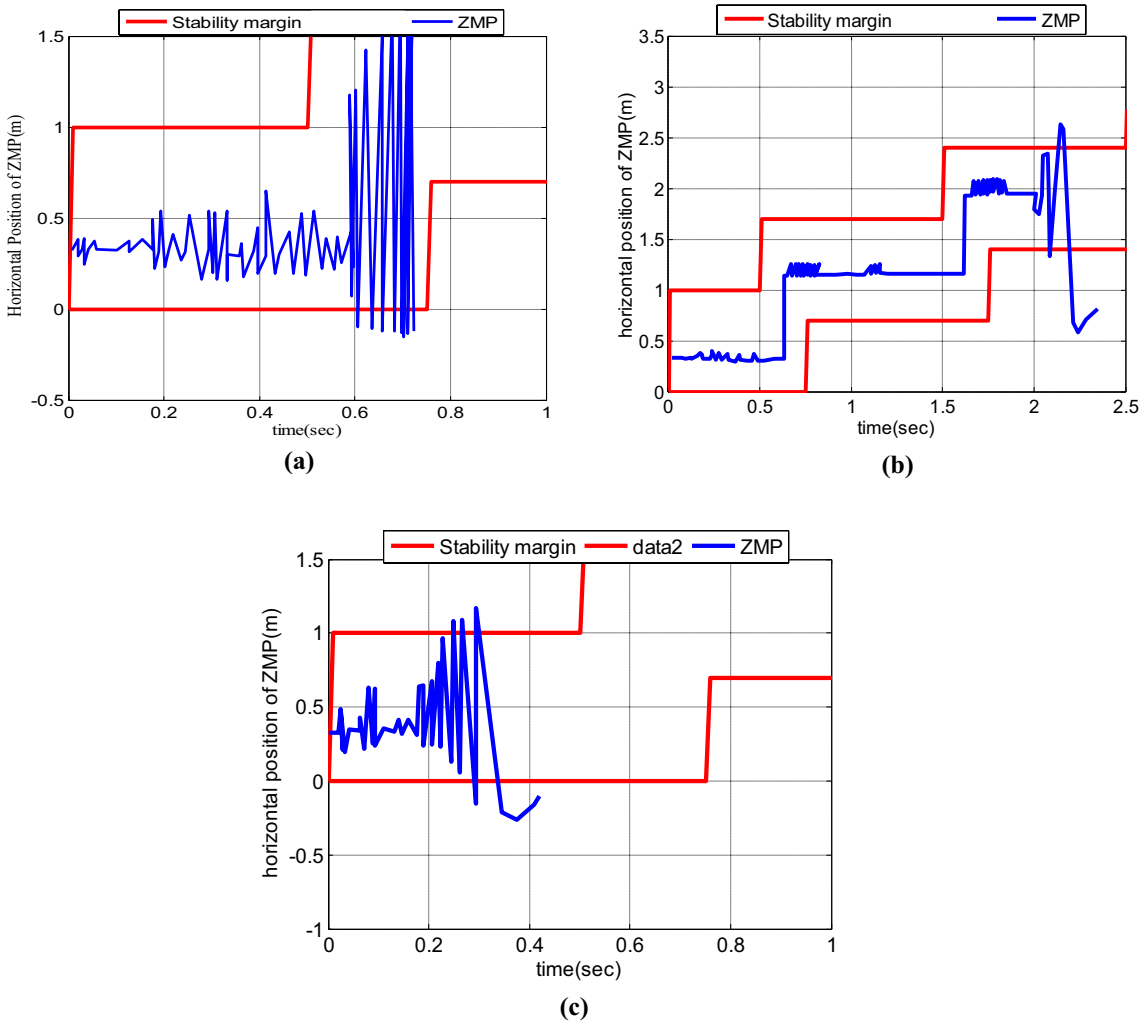

Fig. 19 ZMP of quadruped robot when backstepping-alone controller is used, **a** 10% increase in body mass, **b** external disturbance of 50 NM, **c** 8% SNR measurement noise

parameters, rejecting large external disturbances, and handling moderate measurement noises. Even though the backstepping controller is designed based on the Lyapunov stability theory, its stability for periodic systems with impact effect is not guaranteed. For this reason, the Poincaré map stability criterion was used to check the stability of the proposed method.

Appendix

The kinematic equations of the quadruped robot are given below:

$$x_{C_1} = d_1 \sin\theta_1,\ y_{C_1} = d_1 \cos\theta_1$$
$$x_{C_2} = l_1 \sin\theta_1 + d_2 \sin\theta_2,\ y_{C_2} = l_1 \cos\theta_1 + d_2 \cos\theta_2$$
$$x_{C_3} = l_1 \sin\theta_1 + l_2 \sin\theta_2 + (l_3 - d_3)\sin\theta_3,$$
$$y_{C_3} = l_1 \cos\theta_1 + l_2 \cos\theta_2 - (l_3 - d_3)\cos\theta_3$$
$$x_{C4} = l_1 \sin\theta_1 + l_2 \sin\theta_2 + l_3 \sin\theta_3 + (l_4 - d_4)\sin\theta_4,$$
$$y_{C_4} = l_1 \cos\theta_1 + l_2 \cos\theta_2 - l_3 \cos\theta_3 - (l_4 - d_4)\cos\theta_4$$
$$x_{C_5} = d_5 \sin\theta_5,\ y_{C_5} = d_5 \cos\theta_5$$

$$x_{C_6} = l_5 \sin\theta_5 + d_6 \sin\theta_6, \quad y_{C_6} = l_5 \cos\theta_5 + d_6 \cos\theta_6$$
$$x_{C_7} = l_5 \sin\theta_5 + l_6 \sin\theta_6 + (l_7 - d_7)\sin\theta_7,$$
$$y_{C_7} = l_5 \cos\theta_5 + l_6 \cos\theta_6 - (l_7 - d_7)\cos\theta_7$$
$$x_{C_8} = l_5 \sin\theta_5 + l_6 \sin\theta_6 + l_7 \sin\theta_7 + (l_8 - d_8)\sin\theta_8,$$
$$y_{C_8} = l_5 \cos\theta_5 + l_6 \cos\theta_6 - l_7 \cos\theta_7 - (l_8 - d_8)\cos\theta_8$$
$$x_{C_9} = l_1 \sin\theta_1 + l_2 \sin\theta_2 + l_9 \cos\theta_9,$$
$$y_{C_9} = l_1 \cos\theta_1 + l_2 \cos\theta_2 + l_9 \sin\theta_9$$

where l_i and $d_i (i = 1, \ldots, 9)$ are the length and the half of the length of the ith link, respectively, and x_{c_i} and y_{c_i} are the x–y coordinate of the center of mass of the ith link, respectively.

References

1. Hoffmann M, Simanek J (2017) The merits of passive compliant joints in legged locomotion: fast learning, superior energy efficiency and versatile sensing in a quadruped robot. J Bionic Eng 14(1):1–14
2. Bououdenand S, Abdessemed F (2014) Walking control for a planar biped robot using 0-flat normal form. Robot Auton Syst 62(1):68–80
3. Karydis K, Poulakakis I, Tanner HG (2017) A navigation and control strategy for miniature legged robots. IEEE Trans Robot 33(1):214–219
4. Koo IM, Lee TD, Moon YH, Koo J, Park S, Choi HR (2015) Biologically inspired gait transition control for a quadruped walking robot. Auton Robots 39(2):169–182
5. Na B, Choi H, Kong K (2015) Design of a direct- driven linear actuator for high-speed quadruped robot, Cheetaroid-I. IEEE/ASME Trans Mechatron 20(2):924–933
6. Shakourzadeh Sh, Farrokhi M (2016) Optimal gait generation for quadruped robots using mesh adaptive direct search. Int J Math Model Numer Optim 7(1):66–82
7. Tran DT, Koo IM, Lee YH, Moon H, Park S, Koo JC, Choi HR (2014) Central pattern generator based reflexive control of quadruped walking robots using a recurrect neural network. Robot Auton Syst 62(10):1497–1516
8. Santos PGD, Garcia E, Estremera J (2008) Quadruped locomotion, 1st edn. Springer, Madrid
9. Zhou J, Wen C (2008) Adaptive backstepping control of uncertain systems, 1st edn. Springer, Berlin
10. Ibari B, Benchikh L, Reda A, Ahmed-Foitih Z (2016) Backstepping approach for autonomous mobile robot trajectory tracking. Indones J Electr Eng Comput Sci IAES 3:478–485
11. Wu H, Karkoub M, Hwang C (2015) Mixed fuzzy sliding-mode tracking with backstepping formation control for multi-nonholonomic mobile robots subject to uncertainties. J Intell Robot Syst 79(1):73–86
12. Nikdel N, Badamchizadeh MA, Azimirad V (2017) Adaptive backstepping control for an n-degree of freedom robotic manipulator based combined state augmentation. Robot Comput Integr Manuf 44:129–143
13. Wang LJ, Chao W, Tong G (2014) Central pattern generator based gait control for planar quadruped robots. J Shanghai Jiaotong Univ (Sci) 19(1):1–10
14. Chen X, Watanabe K, Kiguchi K, Izumi K (2002) An art-based fuzzy controller for the adaptive navigation of a quadruped robot. IEEE/ASME Trans Mechatron 7(3):318–328
15. Marhefka DW, Orin DE, Schmiedeler JP, Waldron KJ (2003) Intelligent control of quadruped gallops. IEEE Trans Mechatron 8(4):446–456
16. Kazemi H, Majd V, Moghaddam M (2013) Modeling and robust backstepping control of an underactuated quadruped robot in bounding motion. Robatica 31(3):423–439
17. Shaocheng T, Changying L, Yongming L (2009) Fuzzy adaptive observer backstepping control for MIMO nonlinear systems. Fuzzy Sets Syst 160(19):2755–2775
18. Chen MY, Lin CS (1995) Measurement of robustness for biped locomotion using linearized Poincaré map. In: IEEE international conference on system, man, and cybernetics: intelligent systems for the 21th century, Vancouver, Canada
19. Mu X (2004) Dynamics and motion regulation of a five-link biped robot walking in the sagittal plane. Ph.D. thesis, The University of Manitoba, Department of Mechanical and Manufacturing Engineering, Canada
20. Lewis FL, Abdallah CT, Dawson DN (2004) Control of robot manipulators: theory and practice. Marcel Dekker Inc, New York
21. Kokotovic PV (1992) The joy of feedback: nonlinear and adaptive. IEEE Control Syst Mag 12(3):7–17
22. Wang LX (1996) A course in fuzzy systems and control, 2nd edn. Wiley, Hoboken
23. Grizzle JW, Abba G, Plestan F (2001) Asymptotically stable walking for biped robots: analysis via systems with impulse effects. IEEE Trans Autom Control 46(1):51–64
24. Morris B, Grizzle JW (2009) Hybrid invariant manifolds in systems with impulse effects with application to periodic locomotion in bipedal robots. IEEE Trans Autom Control 54(8):1751–1764
25. Westervelt E, Grizzle J, Chevallereau C, Choi J, Morris B (2007) Feedback control of dynamic bipedal robot locomotion (control and automation). CRC Press, Boca Raton