

Article

Fuzzy Logic-Based Controller for Bipedal Robot

Phan Bui Khoi *  and Hong Nguyen Xuan 

School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi 100000, Vietnam; hong_nx@hicas.com.vn

* Correspondence: khoi.phanbui@hust.edu.vn; Tel.: +84-(0)-24-3868-0469

Abstract: In this paper, the problem of controlling a human-like bipedal robot while walking is studied. The control method commonly applied when controlling robots in general and bipedal robots in particular, was based on a dynamical model. This led to the need to accurately define the dynamical model of the robot. The activities of bipedal robots to replace humans, serve humans, or interact with humans are diverse and ever-changing. Accurate determination of the dynamical model of the robot is difficult because it is difficult to fully and accurately determine the dynamical quantities in the differential equations of motion of the robot. Additionally, another difficulty is that because the robot's operation is always changing, the dynamical quantities also change. There have been a number of works applying fuzzy logic-based controllers and neural networks to control bipedal robots. These methods can overcome to some extent the uncertainties mentioned above. However, it is a challenge to build appropriate rule systems that ensure the control quality as well as the controller's ability to perform easily and flexibly. In this paper, a method for building a fuzzy rule system suitable for bipedal robot control is proposed. The design of the motion trajectory for the robot according to the human gait and the analysis of dynamical factors affecting the equilibrium condition and the tracking trajectory were performed to provide informational data as well as parameters. Based on that, a fuzzy rule system and fuzzy controller was proposed and built, allowing a determination of the control force/moment without relying on the dynamical model of the robot. For evaluation, an exact controller based on the assumption of an accurate dynamical model, which was a two-feedback loop controller based on integrated inverse dynamics with proportional integral derivative, is also proposed. To confirm the validity of the proposed fuzzy rule system and fuzzy controller, computation and numerical simulation were performed for both types of controllers. Comparison of numerical simulation results showed that the fuzzy rule system and the fuzzy controller worked well. The proposed fuzzy rule system is simple and easy to apply.

Keywords: bipedal robot; fuzzy logic; fuzzy control; fuzzy rule; robot kinematics; robot dynamics; robot's motion trajectory



Citation: Khoi, P.B.; Nguyen Xuan, H. Fuzzy Logic-Based Controller for Bipedal Robot. *Appl. Sci.* **2021**, *11*, 11945. <https://doi.org/10.3390/app112411945>

Academic Editor: Adel Razek

Received: 22 November 2021

Accepted: 13 December 2021

Published: 15 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A bipedal robot has a structure with many degrees of freedom. An important feature of bipedal robots is poor stability because the robot is supported by only two feet on the ground. Bipedal robots are capable of performing a variety of tasks to replace humans, serve humans, or coordinate operations with humans.

Because of such structural characteristics and applicability, bipedal robots are of wide research interest [1–8].

The first important problem is the dynamics of bipedal robots. Although there have been many studies on the problem of dynamics [8–16], because the structure is very diverse for each field of application, this is still an open problem. Normally, Newton–Euler equations and Lagrange equations of the 2nd kind are widely applied when investigating the dynamics of bipedal robots. As shown in the following section, the research applied Lagrange equations of the 2nd kind.

An important feature of the dynamics problem is the relationship between motion and the applied force, which ensures the robot's manipulative motion and at the same time ensures that the robot is in balance, i.e., maintaining the desired posture and state without falling.

This is closely related to the problem of designing the robot's motion trajectory and controlling the robot.

There are many studies on motion trajectory planning for bipedal robots based on different approaches [17–22], but due to the robot structure being positioned only by two feet on the ground with poor stability, a class of motion trajectory design problems was implemented taking into account the dynamic characteristics. Accordingly, the motion trajectory was designed with dynamic characteristics following the principle of ZMP point (Zero Moment Point) [23–33].

Basing the design of the motion trajectory and the control a bipedal robot on the principle of the ZMP point has been widely applied. However, ensuring that the ZMP point is always within the support polygon limits the flexibility of the robot and increases the complexity and computational load when controlling the robot.

Actual observation of human activity shows that there are many times or periods when the ZMP point is outside the support polygon, but human activity is still stable and ensures dynamic balance. This is explained first of all by the process of falling, which will have to take place over a period of time. When the time to change the position of the robot is small enough compared with the time of falling, the robot will not fall; the posture and equilibrium will change insignificantly. A second point is that humans have the ability to adjust the interaction to ensure operation while maintaining a state of balance.

In order for a robot to mimic human-like interaction, the robot's structure must first have a form close to that of a human. In addition, the ability to adjust the interaction flexibly thanks to the motors that drive at the joints must be based on the same rules as humans. Crisp controllers based on dynamical models with closed equations will find it difficult to meet this requirement.

Meanwhile, controllers based on fuzzy logic imitating human natural reasoning have been studied and applied in many control problems in general and in the control of bipedal robots—for example, a fuzzy logic-based controller for robot in mechanical processing [34,35], a fuzzy-based-admittance controller for safe natural human–robot interaction [36], and a fuzzy logic-based bipedal robot control [37–39]. Moreover, the integration of fuzzy logic with intelligent algorithms is also a research direction that is being applied to control bipedal robots [40–42].

With robots applied in mechanical processing requiring high accuracy, the research [34,35] has shown that in the construction of a fuzzy rule system as well as in the determination of an appropriate physical value domain, a fuzzy controller gives accurate and reliable results. Although the control of a bipedal robot does not require high precision, as in machining, the difficulty lies in the fact that the movement of the robot is more flexible, the operation is not clearly defined as a mechanical engineering process. The authors of [37] show that fuzzy logic can be used to compensate for the error of the computed torque controller. In the work presented in [38], it was shown that the combination of fuzzy logic and ant colony algorithm could be used to eliminate the noise that occurs when a bipedal robot moves. This is also a new direction on the method of integrating intelligent algorithms that this paper mentions at the end. However, the model studied in [38] was far from a real model, and the motion trajectory was not fully and clearly performed. While the control force/moment is a quantity that needs to be determined to ensure the law of motion of the robot, it depends on the dynamic quantities of the robot. Likewise, the robot model in [39] was also far from a real model in dynamic parameters, and motion trajectory was not performed fully and clearly.

In this paper, the difference from the previous works is that first of all, a fuzzy system and fuzzy inference were built, allowing the determination of control force/moment without using the dynamic model of the robot. The proposed dynamic model with kinematic

and dynamic parameters was close to that of a real human. The control of the robot was closely related to the robot's gait as well as to the dynamic factors and the equilibrium condition. Therefore, the paper presents a comprehensive overview of the work, including the motion trajectory planning and dynamic analysis that bring the control problem closer to reality.

In Section 2 of the paper, the method of designing the motion trajectory of the robot is presented based on observation and the interpolation method so that it has a human gait. The design of motion trajectories taking into account the dynamic characteristics is considered in Section 3 to investigate some parameters that can improve the dynamic characteristics. In this way, it is possible to get the robot's motion trajectory close to the human gait. At the same time, the ZMP is either inside the support polygon or, if it is outside the support polygon, but the ZMP is still within a distance and time that allow the fuzzy controller to still ensure robot control to follow the trajectory and maintain balance. Section 4 presents crisp controllers based on the dynamical model of the robot. Section 5 presents the proposed fuzzy rule system for a bipedal robot along with the method of composing the fuzzy rule system, fuzzification, fuzzy inference, and defuzzification applied to the fuzzy controller for the bipedal robot. Section 6 presents the simulation results of the controllers. Section 7 concludes with the reliability and efficiency of the fuzzy controller based on the established rule system and considers research and development.

2. Kinematics of Robot and Motion Trajectory Planning

A human-like bipedal robot can have main parts that include a head, body, two arms, and two legs as shown in Figure 1a. Arms and legs are made up of several links, connected by joints, and a joint can have from 1 up to 3 degrees of freedom of motion. The head, arms, and legs are connected to the robot body by joints from 1 to 3 degrees of freedom. With such a structure, the robot can perform movements as well as manipulations ranging from simple to complex spatial actions. The goal of the paper was to present the robot's walking control and not to mention the dynamic effects of arm movement. Therefore, mechanically, the head and arms were integrated with the robot body as a solid object and are collectively referred to as the robot body when surveying and calculating from now on. The robot can move on flat horizontal ground, or on ground that is sloping and rough, convex, and complex. A simple motion of the robot is a straight, horizontal motion. This motion is called a motion that is in the sagittal plane [21]. Figure 1a–c shows the robot moving in the sagittal S plane. With this motion, the survey model of the robot can lead to a simple model in which the parts of the legs that are connected to each other and to the robot body are revolute joints. Revolute joints that connect the robot body to the thighs of the legs are called hip joints. The knee joints connect the thighs to the calves, the ankle joints connect the calves to the feet. All revolute joints have a rotation axis perpendicular to the sagittal plane. With such structure and motion in the sagittal plane, the survey model of the robot is returned to the flat model, as shown in Figure 1b,c. Figure 1b represents when the robot is stationary in the vertical state, Figure 1c demonstrates the robot in walking position. Because the paper focuses on studying control algorithms based on fuzzy logic and building a fuzzy rule system, this flat model was used.

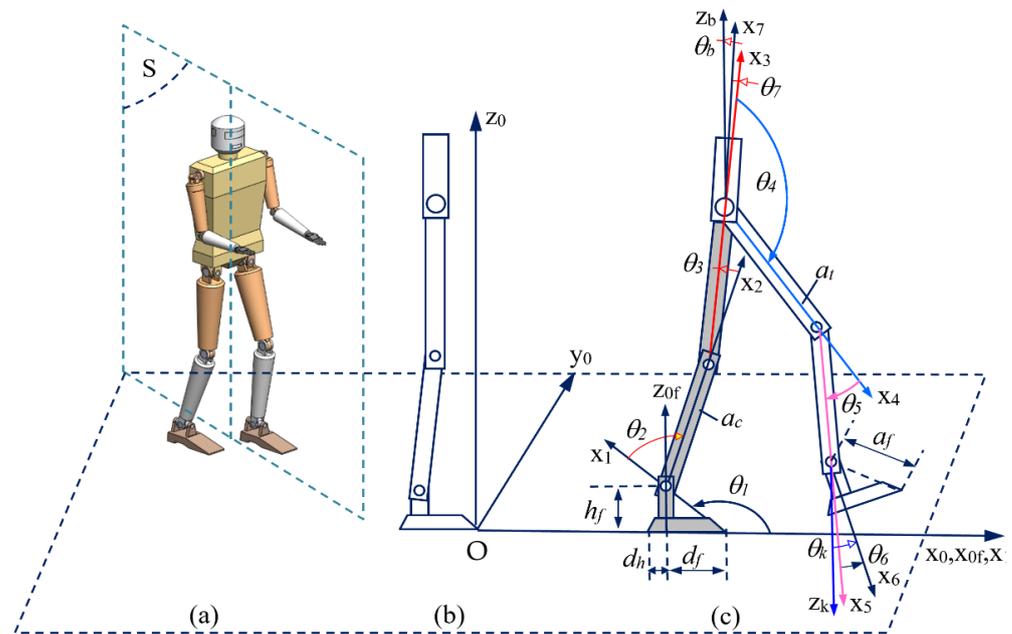


Figure 1. Bipedal robot model: (a) human-like bipedal robot; (b,c) robot’s kinematic scheme and coordinate systems.

2.1. Kinematic Equations of Robot

When a robot moves, one leg will act as the supporting leg, the other leg will swing in the direction of travel to take a step. These will be called supporting (leg, foot) and swinging (leg, foot), respectively. To set up the kinematic equations of the robot, the coordinates systems and kinematic parameters were derived as shown in Figure 1c, in which the coordinate system $x_0y_0z_0$ was attached to the background, which was the base coordinate system. The $x_{of}y_{of}z_{of}$ coordinate system was also attached to the background, which was used to represent the robot position at each step. The $x_1y_1z_1$ coordinate system was attached to the foot of the supporting leg, having its origin at the toe of the foot. The $x_7y_7z_7$ coordinate system was attached to the robot body, whose origin was at the hip joint.

The angles $\theta_2, \dots, \theta_7$ are the joint coordinates, where θ_7 is joint coordinate between the thigh, axis x_3 , and body, axis x_7 . These coordinates, together with θ_1 , i.e., $\theta_1, \theta_2, \dots, \theta_7$, are independent generalized coordinates, defining the robot’s state and motion. On the other hand, in space, the factors that characterize the robot’s state are the position of the robot body and the positions of the supporting foot and the swinging foot. At each step, the coordinate system $x_{of}y_{of}z_{of}$ was determined with respect to the coordinate system $x_0y_0z_0$. The robot body position is represented by the hip joint coordinates x_b, y_b , and angle θ_b between the x_7 axis, which is the axis mounted along the robot body, relative to the vertical, axis z_b . The position of the supporting foot is determined by the coordinates of the supporting foot x_1, y_1 , and angle θ_1 . The coordinates of the ankle, x_k, y_k , and θ_k , are used to represent the swinging foot position. Where, θ_k is the angle between the swinging foot, x_6 axis, and the vertical, axis z_k .

In the calculation, the coordinates x_{of}, y_{of}, z_{of} are determined by the coordinate system $x_{of}y_{of}z_{of}$ with respect to the base coordinate system $x_0y_0z_0$; the coordinates $x_b, y_b, \theta_b, x_1, y_1, \theta_1, x_k, y_k, \theta_k$ are determined in each step and are represented with respect to the $x_{of}y_{of}z_{of}$ coordinate system.

In general, when the robot is operating, from the operation request, the robot’s attitude is determined by the position of the robot’s body, supporting foot, and the swinging foot. Therefore, the coordinates locating the above components are called operational (task) coordinates and are denoted as follows:

$$p = [p_1, p_2, \dots, p_{12}]^T = [x_{of}, y_{of}, z_{of}, x_b, y_b, \theta_b, x_1, y_1, \theta_1, x_k, y_k, \theta_k]^T \tag{1}$$

The independent generalized coordinates are denoted as follows:

$$q = [q_1, q_2, \dots, q_7]^T = [\theta_1, \theta_2, \dots, \theta_7]^T \tag{2}$$

In addition to the time-varying parameters, which are the coordinates mentioned above, there are also geometric parameters of the robot, including the robot’s foot dimensions h_f, d_f, d_h (Figure 1c), and the length of the legs calves and thighs are a_c and a_t , respectively.

The operational coordinates (1) are the parameters for designing the motion trajectory, while the independent coordinates (2) are the basis for controlling the robot’s motion. However, only six generalized coordinates, which are the joint coordinates $\theta_2, \dots, \theta_7$ are driven and controlled, while the number of independent generalized coordinates is seven. Thus, the bipedal robot is an underactuated system. The angle θ_1 is a parameter related to the stable equilibrium of the robot. The guarantee of θ_1 angle value for the purpose of stable balance of the robot is not performed directly by the driving motor but depends on the dynamic state of the robot as well as the control algorithm applied on the joints driven by the motor. Therefore, θ_1 is both a parameter related to the design problem of the motion trajectory and one of the independent coordinates related to the problem of controlling the robot’s motion. The following sections will continue to present issues related to this problem. The angles θ_6, θ_7 represent the posture of the robot body and swinging feet, respectively, in stationary as well as in moving states.

Using the homogeneous coordinate transformation transfer matrix method, the robot kinematic equations are set up as follows:

$$\begin{cases} a_t \cos \theta_{13} + a_c \cos \theta_{12} + a_f \cos \theta_1 + d_f - x_b = 0; \cos \theta_{13} = \cos(\theta_1 + \theta_2 + \theta_3); \cos \theta_{12} = \cos(\theta_1 + \theta_2); \\ a_t \sin \theta_{13} + a_c \sin \theta_{12} + a_f \sin \theta_1 - y_b = 0; \sin \theta_{13} = \sin(\theta_1 + \theta_2 + \theta_3); \sin \theta_{12} = \sin(\theta_1 + \theta_2); \end{cases} \tag{3}$$

$$\begin{cases} a_c \cos \theta_{15} + a_t \cos \theta_{14} + a_t \cos \theta_{13} + a_c \cos \theta_{12} + a_f \cos \theta_1 + d_f - x_k = 0; \cos \theta_{15} = \cos(\theta_1 + \dots + \theta_5); \cos \theta_{14} = \cos(\theta_1 + \dots + \theta_4); \\ a_c \sin \theta_{15} + a_t \sin \theta_{14} + a_t \sin \theta_{13} + a_c \sin \theta_{12} + a_f \sin \theta_1 - y_k = 0; \sin \theta_{15} = \sin(\theta_1 + \dots + \theta_5); \sin \theta_{14} = \sin(\theta_1 + \dots + \theta_4); \end{cases} \tag{4}$$

From the kinematic structure diagram, it is possible to derive

$$\theta_6 = \theta_k - (\theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5) - \frac{\pi}{2} \tag{5}$$

$$\theta_7 = \theta_b - (\theta_1 + \theta_2 + \theta_3) + \frac{\pi}{2} \tag{6}$$

With the position and state of the robot at each step being determined by the operational coordinates (1), Equations (3)–(6) allow solving the generalized coordinates (2), in which the joint coordinates $\theta_2, \dots, \theta_7$ are driven, thus creating the basis for controlling robot movements. In contrast, the generalized coordinates (2) are identified by sensors allowing the solution of Equations (3)–(6) to find the robot’s posture at each step taken.

Therefore, the systems of kinematic Equations (3)–(6) are used when designing motion trajectories for the robot.

2.2. Robot Motion Trajectory Planning

The bipedal robot can perform human-replacement operations, it can serve humans as a normal robot, and it can cooperate by manipulating objects with humans in human environments. Therefore, the robot’s activities will be very diverse in general, and with walking in particular, there are many different status postures.

In terms of the stages of movement in each step of the robot, there can be three stages as follows:

- The stage in which one foot lands on the ground, while the other swings in the air to take a step in the direction of travel. This phase is called the single phase of the robot’s walk.

- The stage in which both feet rest on the ground and the robot body system moves in the direction of motion. This phase is called the double phase of the robot's walk.
- The stage in which both feet leave the ground. This phase is called the aerial phase of the robot's walk.

Usually, the stage corresponding to the aerial phase occurs only when moving very quickly, such as when running or jumping, and occurs at the time of switching from the supporting foot to the swinging foot and vice versa. The stage corresponding to the double phase can also occur at the time of switching from the supporting foot to the swinging foot and vice versa. Therefore, the duration of the double phase is usually small and even very small if the motion is not too slow.

As mentioned, the goal of this paper focuses on a robot controller based on fuzzy logic. To confirm the validity of the fuzzy controller and the fuzzy rule system, the robot's walk that is close to that of a human is investigated. Therefore, it is possible to ignore the double phase and the aerial phase because the duration of these phases is very small.

In terms of speed of movement, walking can be accomplished with increasing velocity, constant velocity, or decreasing velocity.

From observations, it has been shown that the human gait is related to posture and state of the body, as well as to that of the robot's feet. Normally, from the beginning of a movement with increasing velocity and reaching a constant velocity, a robot body tends to lean toward the motion (forward). This can be seen when observing runners. When walking at a decreasing speed, the body tends to lean in the opposite direction (backward). The supporting foot can have two states—that is, the foot is in full contact with the platform or transitions from full contact and gradual decrease to point contact at the toe before leaving the platform. The swinging foot can have three states when landing—that is, landing with the toe of the foot, landing with the heel of the foot, or landing with the entire surface of the foot.

The above states on the one hand depend on the individual human, while on the other hand, they are related to dynamic conditions, such as ensuring dynamic balance, non-slip conditions, and optimal energy.

Therefore, when investigating with the bipedal robot model, the parameters representing these states, as denoted above by (1), were used for designing the motion trajectory as well as for controlling the motion of the bipedal robot.

Modeling and simulation methods are often applied to planning motion trajectories for robots in general and bipedal robots in particular. Inputs for modeling and simulation of bipedal robots can be obtained by cameras and sensors attached to the robot body and at the joints when taking a step. In this way, it is possible to obtain the entire motion trajectory of the robot so that it is close to the human gait. A simpler way is to observe human movement and record a number of points per step and to then use interpolation to obtain the full motion trajectory for each step. In the paper, the second method was used.

Figure 2 depicts the robot's steps, in which lines 1, 3, 5, 7, and 9 represent the movement of the ankle joint, and lines 2, 4, 6, 8, and 10 represent the movement of the hip joint. The movement of the robot body was recorded at trace 11. Robot movements were simulated and saved at the link [43]. The length of each step of the foot, also of the ankle joint, is denoted by $2L$, and that of the hip joint is denoted by L . Particularly for the starting step from the resting state, which with the initial position is vertical, the movement of the ankle joint is L . Similarly, the final step back to rest, in an upright position, also has lengths L for the movement of the ankle joint.

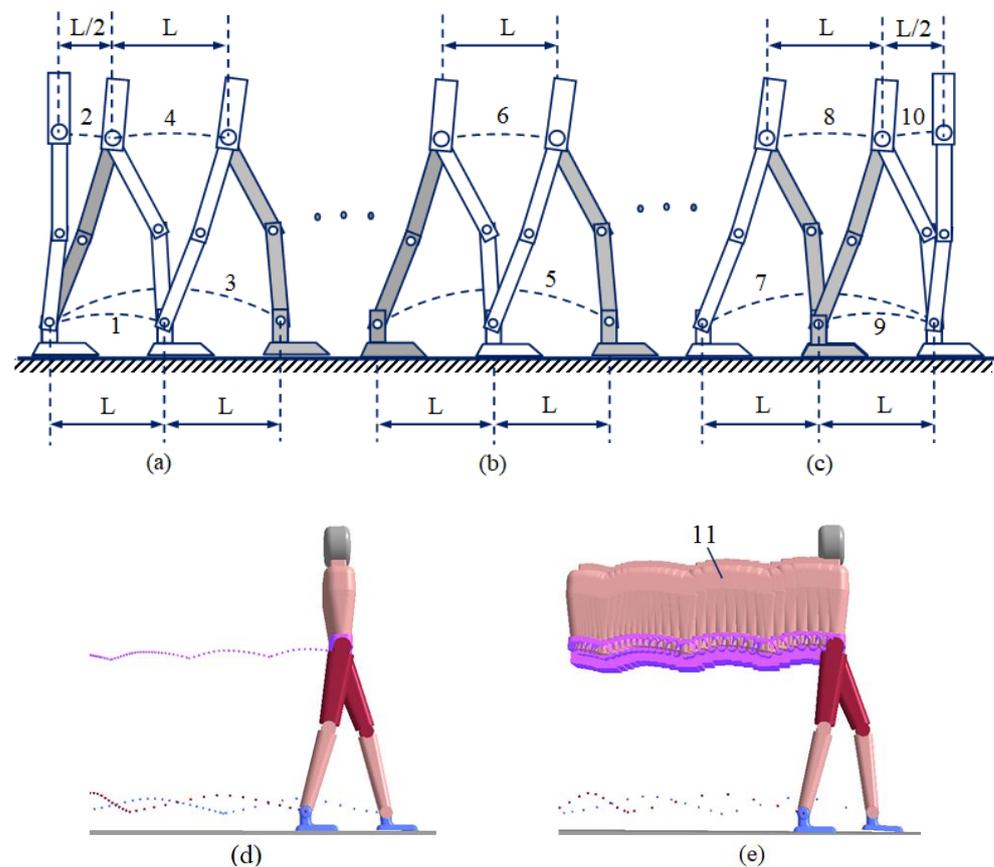


Figure 2. The walking steps of bipedal robot: (a) the starting steps; (b) the steady steps; (c) the final steps; (d) the robot's gait; (e) posture of robot body when moving: <https://youtu.be/8A3849PUCyM> (accessed on 21 September 2021).

To create a database for the problem of dynamics and control, some numerical calculations and simulations of the motion trajectory of the robot were performed. A proposed robot model with kinematic and dynamic parameters is listed in Table 1. The process of starting the robot usually moves at an increasing speed, while the motion slows down to a state of rest. In the steady-state of motion, the horizontal velocity of the robot body was assumed to be constant.

Calculations and simulations were performed with the typical steps for the above three processes as follows:

- The first step of the acceleration process, called the “starting step” from now on;
- The step of the steady motion process, called the “steady step”;
- The last step of deceleration to rest, called the “final step”.

The kinematic parameters in Table 1 were used to calculate and design motion trajectories for the robot corresponding to these three steps, and the results are shown in Figures 3–8.

Table 1. The kinematic, dynamic, and motion trajectory parameters of the proposed robot model.

The Geometrical, Kinematical, and Motion Trajectory Parameters					
Geometrical Parameters	Values	Dimensional	Kinematical Parameters	Values	Dimensional
foot size h_f	0.08	m	normal step length	$2L = 0.76$	m
foot size \bar{d}_f	0.132	m	starting step length	$L = 0.38$	m
foot size \bar{d}_h	0.06	m	final step length	$L = 0.38$	m
thigh length a_t	0.38	m	steady velocity	$V_c = 4275$	m/h (hour)
calf length a_c	0.38	m	average speed of starting step	$V_s = 1487$	m/h
body height h_b	0.45	m	average speed of final step	$V_f = 1368$	m/h

The dynamical parameters										
Links	Mass (kg)	Link's center of mass coordinates (m)			Inertia tensor of links (kg·m ²)					
		x_{ci}	y_{ci}	z_{ci}	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{xz}	I_{yz}
1	1.22803	−0.06792	0.02140	0	0.00138	0.24390	0.0055	0.00129	0	0
2	3.25665	−0.18377	0	0	0.00122	0.04695	0.04714	0	0	0
3	5.86021	−0.16274	0	0	0.00279	0.10417	0.10467	0	0	0
4	5.86021	−0.21726	0	0	0.00279	0.10417	0.10467	0	0	0
5	3.25665	−0.19623	0	0	0.00122	0.04695	0.04714	0	0	0
6	1.22803	0.05350	0	0	0.00490	0.00111	0.0055	0.000851	0	0
7	38.8	0.36870	0	0	0.24390	2.21146	2.02948	0	0	0

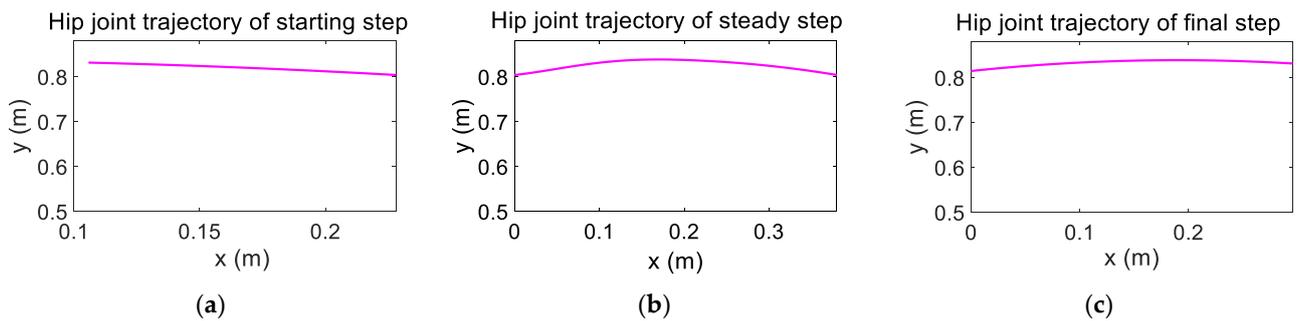


Figure 3. The motion trajectory of the hip joint in the starting (a), steady (b), and final (c) steps.

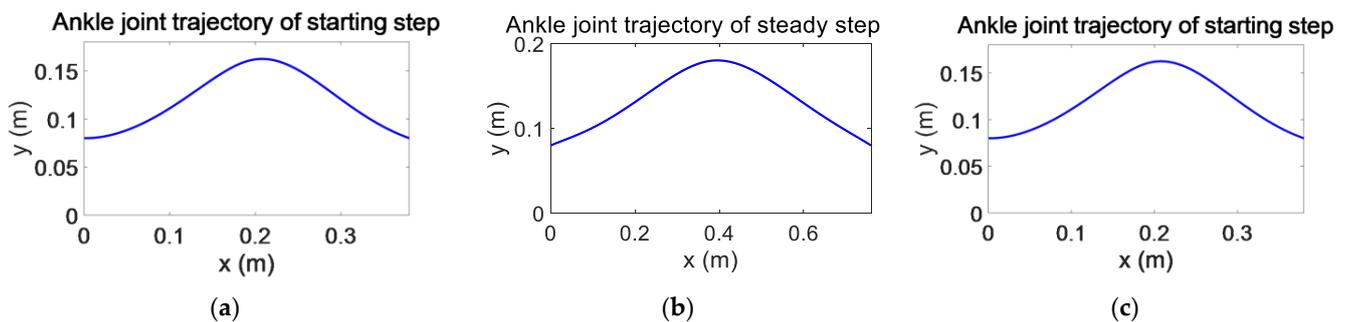


Figure 4. The motion trajectory of ankle joint in the starting (a), steady (b), and final (c) steps.

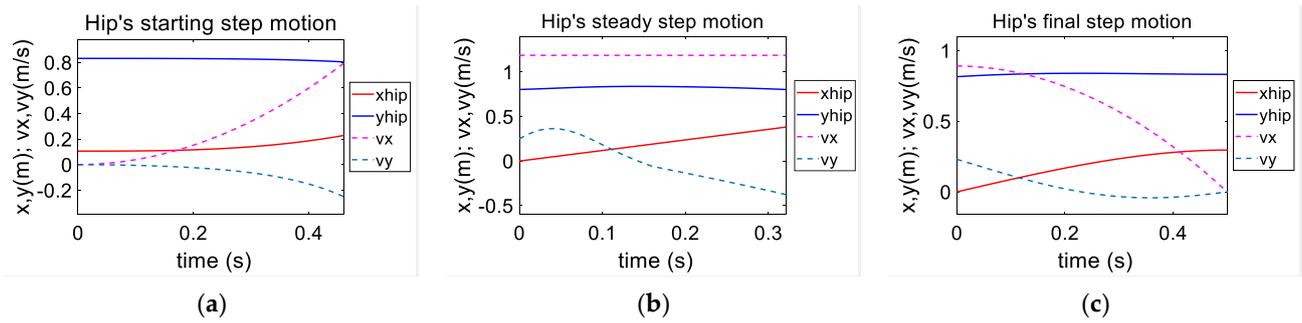


Figure 5. The motion of the hip joint in the starting (a), steady (b), and final (c) steps.

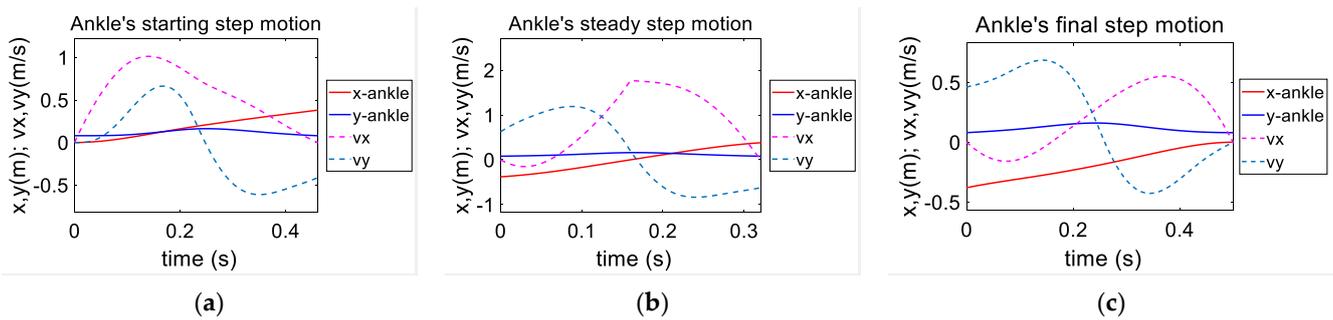


Figure 6. The motion of the ankle joint in the starting (a), steady (b), and final (c) steps.

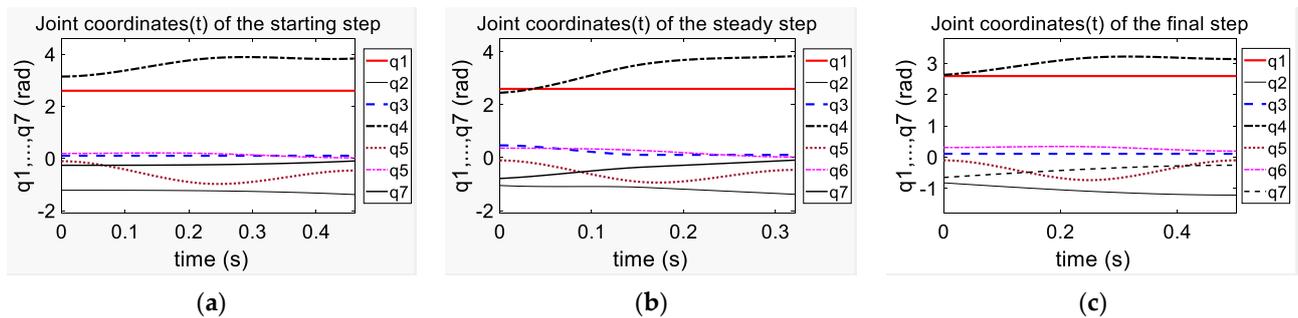


Figure 7. The motion of robot link: the joint coordinates (t) in the starting (a), steady (b), and final (c) steps.

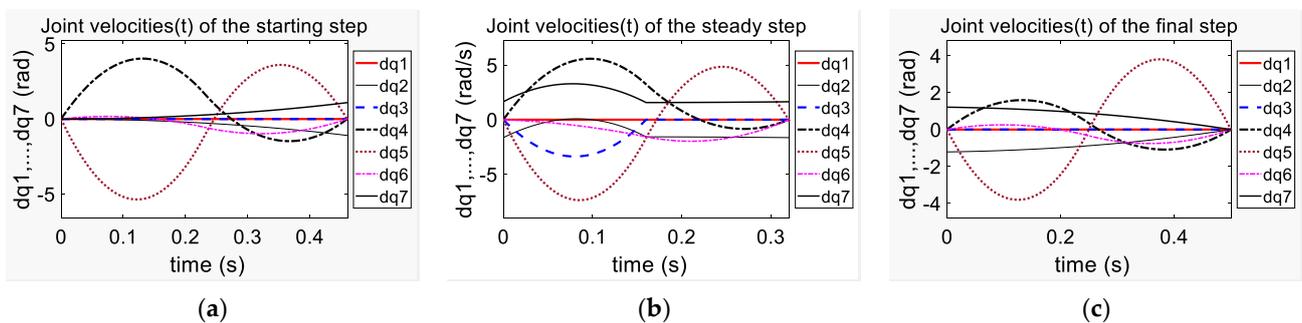


Figure 8. The motion of robot link: the joint velocities (t) in the starting (a), steady (b), and final (c) steps.

Figure 3a–c shows the motion trajectory of the hip joint in the starting, steady, and final steps, respectively. Similarly, the movement of the ankle joint is shown in Figure 4a–c.

Figure 5a–c shows the position and velocity of motion over time of the hip joint at the starting, steady, and final steps, respectively. Similarly, the movement over time of the ankle joint is shown in Figure 6a–c.

With the calculated motion of the hip and ankle joints, applying Equations (3)–(6) as presented in Section 2.1, the relative motion of the robot’s links was determined by the joint coordinates, according to the time, and were calculated and shown in Figures 7 and 8.

Figure 7a–c shows the position of the links and the joint coordinate value in time at the starting, steady, and final steps, respectively. Figure 8a–c shows the relative velocity of the links in time at the starting, steady, and final steps, respectively.

The trajectories of the robots in the form of joint coordinates and their 1st and 2nd derivatives with respect to time were saved in “pq^(k).txt”, “vq^(k).txt”, “aq^(k).txt” files. Here, the index $k = 1, 2, 3$ corresponds to the starting step, the steady step, and the final steps. These data files were used to simulate the controllers in Section 6.

The analysis of human gait and the selection of kinematic parameter values for motion trajectory design is related to the dynamics of the robot, and are further analyzed in the following sections.

3. Dynamics of Robot

In the paper, the Lagrange equation of the 2nd kind was applied to establish the system of differential equations of motion of a bipedal mobile robot, as follows:

$$M(q)\ddot{q} + \psi(q, \dot{q}) + G(q) + Q(q) = U \tag{7}$$

Here:

$M(q)$ is the mass matrix of size (7×7) , whose elements are functions of the generalized coordinates q , determined as follows:

$$\begin{cases} M(q)_{(7 \times 7)} = \sum_{i=1}^7 (J_{Ti}^T m_i J_{Ti} + {}^{ci}J_{Ri}^T {}^{ci}\Theta_{ci} {}^{ci}J_{Ri}) \\ J_{Ti} = \frac{\partial r_{ci}}{\partial q} \\ J_{Ri} = \frac{\partial {}^{ci}\omega_{ci}}{\partial \dot{q}} \end{cases} \tag{8}$$

where C_i is the center of mass of link i , r_{ci} is the location vector of C_i with respect to the coordinate system $x_{of}y_{of}z_{of}$; ${}^{ci}\omega_{ci}$ is the angular velocity of link i represented in the coordinate system of link i ; m_i is the mass of link i ; ${}^{ci}\Theta_{ci}$ is the inertia tensor of link i with respect to the center of mass C_i , represented in the i -th coordinate system, which has its origin at the center of mass of link i .

$\psi(q, \dot{q})$ (7×1) is the vector of size (7×1) , whose elements are generalized forces of Coriolis and centrifugal inertial forces, which are calculated through the elements of matrix $M(q)$ by the formula:

$$\begin{cases} \psi_j = \sum_{k,l=1}^7 (k, l; j) \dot{q}_k \dot{q}_l; \\ (k, l; j) = \frac{1}{2} \left(\frac{\partial m_{kj}}{\partial q_l} + \frac{\partial m_{lj}}{\partial q_k} - \frac{\partial m_{kl}}{\partial q_j} \right) \end{cases} \tag{9}$$

where $(k, l; j)$ is the Christoffel notation with three indexes of the first kind; m_{kl} ($k, l = 1, \dots, 7$) are the elements of the matrix $M(q)$.

$G(q)$ (7×1) is the generalized force vector of the conservative forces acting on the robot.

$$G(q)_{(7 \times 1)} = [G_1, G_2, \dots, G_7]^T; \quad G_j = \frac{\partial \Pi}{\partial q_j} \tag{10}$$

Π is the sum of the potential energy of the robot.

$Q(q)$ (7×1) is the vector of the generalized forces of the non-conservative forces acting on the robot. The article investigates the robot’s motion, without interacting with the environment. If the robot is carrying a load, the load will cause a conservative force to appear. Therefore, non-conservative forces are assumed to include only frictional and disturbance forces at the joints.

Some dynamical factors directly related to the motion of a bipedal robot are the non-slip condition and the dynamic balance condition of the robot. The non-slip condition mainly depends on the friction between the platform and the supporting foot, assuming that the friction is large enough to ensure that the non-slip condition is always satisfied. Regarding the dynamic equilibrium condition, there have been many studies based on the application of ZMP [23–33] that have been discovered and proposed by MIOMIR VUKOBRATOVIC [23,24]. Accordingly, if a robot moves in a trajectory so that the ZMP point is always inside the supporting polygon, the dynamic equilibrium condition will be guaranteed—that is, the robot will not fall. This method has been widely applied for nearly half a century and provides good results in controlling a moving robot, following the designed trajectory based on the principle of the ZMP point.

However, is it necessary to fully comply with the ZMP point principle when planning the motion trajectory of a bipedal robot? Such a question is raised because it is a rather difficult and complicated job. In addition, a robot's mobility is limited. When a robot's motion velocity is significantly greater than the speed that can occur when falling, then the robot "did not fall" before it entered a new state. This is easy to see when observing a person running. Intuitively and through preliminary calculations, the robot's ability to drop increases when the ZMP point is outside and further away from the boundary of the support polygon.

Figure 9 shows the possible states of the robot when the ZMP point is outside the support polygon, where Figure 9a shows that the contact of the supporting foot with the ground remains almost unchanged until the swinging foot begins to make contact with the ground. Figure 9b shows the change in contact of the supporting foot with the ground when the swinging foot begins to make contact with the ground.

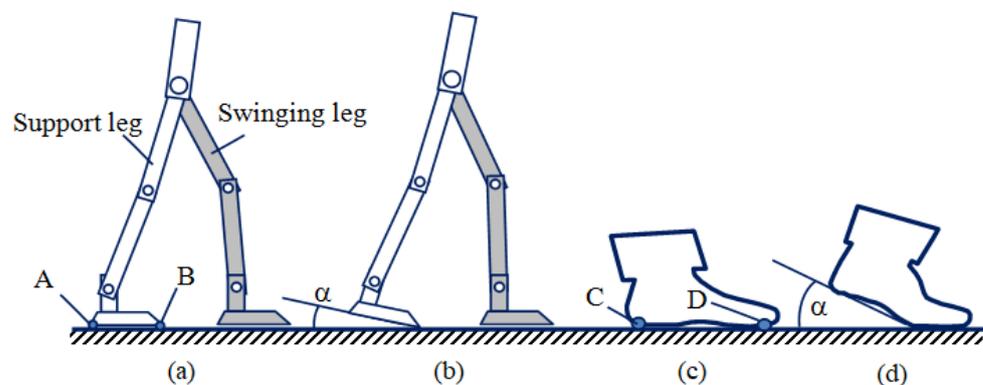


Figure 9. The robot state when the ZMP point is outside the support polygon.

These cases are similar to the walking of the human foot shown in Figure 9c,d, respectively.

In practice, there is often a change in contact between the supporting foot and the ground. This is explained because the ZMP point is outside the supporting polygon, so according to the principle of the ZMP point, there will be a phenomenon of flipping around point A, or B, with the robot model. With the human foot model, the flip point can be either C or D. However, due to the time of step's transition is very short, the α angle value for the case as shown in Figure 9a,c is very small, can be considered as zero.

In this article, we investigate a number of factors that can affect the robot's ability to flip around the fulcrum, which are the toe of the foot (point A), or the heel (B), as shown in Figure 9.

First of all, it is the θ_b angle between the robot body and the vertical. As mentioned above, when observing human movement in the process of increasing speed or steady speed, the body often leans toward the motion, while leaning in the opposite direction to the direction of motion when the motion slows down to a stationary state. This is the basis for the selection of θ_b values.

Another parameter to be considered is the position of the center of mass of the whole bipedal robot system. Intuitive and preliminary calculations show that the lower the center of mass position, the closer the ZMP point is to the support polygon domain or the closer that it is to being within the support polygon domain. However, the position of the center of mass is always changing in the moving process, and the calculation is quite complicated, so here another parameter was chosen, which was the θ_3 joint coordinate of the supporting leg. Because if θ_3 was large, then the center of mass would be low and vice versa. Figure 10 shows some robot gaits with different levels of lowering of the center of mass. The relative horizontal position of the center of mass relative to the position of the supporting foot also affects the robot's ability to resist tipping.

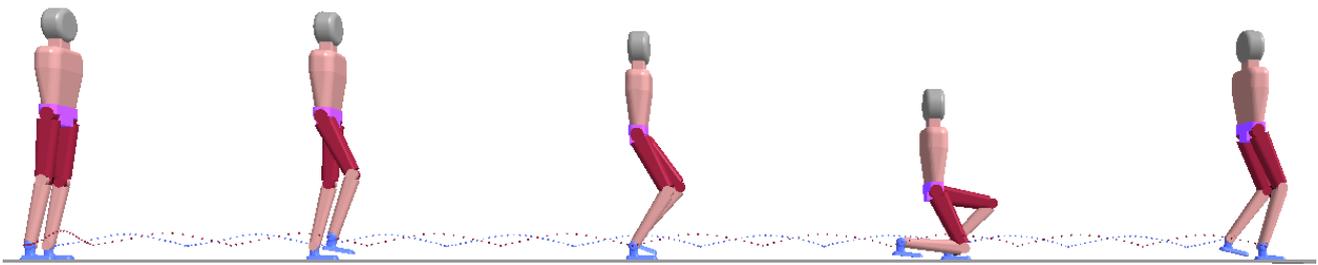


Figure 10. Some robot gaits with different levels of lowering of the center of mass.

By changing the parameters θ_b , θ_3 , and the law of coordinated motion for the supporting leg and swinging leg, some motion trajectories of the robot were calculated and designed and are presented in Section 2.2. These motion trajectories were tested through numerical simulation of dynamics and ensured that the robot did not fall and that it moved according to the designed trajectory. This is explained because the parameters θ_b , θ_3 , and the law of coordinated motion of the robot legs, on the one hand, affected the position of the ZMP point, and on the other hand changed the dynamic quantities that mainly affected inertia force/momentum and driving moments. It should be emphasized that the driving moments on the one hand ensured that the robot's movement was according to the set trajectory, and on the other hand that it also affected the robot's dynamic equilibrium.

Therefore, studying a control algorithm for the robot to follow its trajectory and ensure the condition of not flipping is one of the key issues of the paper and is presented in Sections 4 and 5. To facilitate the presentation and distinguish it from the terms "equilibrium" in general and "dynamic equilibrium" of bipedal robots, in cases in which the robot does not fall and moves along the designed trajectory even though the ZMP point is outside the supporting polygon, it is called "relative equilibrium".

4. The IDPD, IDPID Controller for Bipedal Robot

As shown, the bipedal robot was an underactuated system having seven generalized coordinates, of which six were controllable joint coordinates. The non-driven coordinate θ_1 was related to the robot's relative equilibrium condition. The relative equilibrium condition was guaranteed if θ_1 remained unchanged or changed within a range such that the robot was still moving along the designed trajectory. The robot lost its relative equilibrium if θ_1 changed greatly, leading to the robot falling or completely deviating from the designed trajectory. Thus, the underactuated system of the biped robot was different from other underactuated systems. That is, when the uncontrollable coordinates deviated greatly, the controlled coordinates also became out of control. As preliminarily mentioned above, among the factors affecting the relative equilibrium condition of bipedal robots, the moments that drive the controlled joints are also directly affected. The problem was to find control algorithms with which the driving moments were calculated to ensure both the motion laws of the controlled joints and at the same time contribute to ensuring the relative equilibrium conditions. The most common and popular controller in the industry is a controller based on two closed feedback loops consisting of an inner feedback loop, to

calculate Inverse Dynamics, and an outer loop that applies Proportional Derivative control law (Inverse Dynamics + PD), or Proportional Integral Derivative (Inverse Dynamics + PID). These controllers are also commonly applied in robotics engineering. For the convenience of presentation, we abbreviate the controller Inverse Dynamics + PD or Inverse Dynamics + PID as IDPD or IDPID, respectively. With the motion trajectory of the bipedal robot designed to satisfy the ZMP point principle or the relative equilibrium condition as stated, applying the control law IDPD, IDPID was convenient and easy. However, with these controllers, the driving moments were calculated based on the equations, and so it was not flexible to adjust the driving torque to contribute to increasing the robot’s ability to ensure the relative equilibrium condition.

In his natural walk, man adjusts himself based on natural reflexes trained from natural “inferences” from the first steps.

The fuzzy logic-based controller (hereinafter referred to as the fuzzy controller) was based on natural human reasoning, so it had the flexibility to adjust the interactions reasonably. Specifically, the fuzzy controller was built by a reasonable fuzzy rule system based on natural human inference, allowing it to adjust the driving torques to ensure the control goal. The use of a fuzzy controller for a bipedal robot is presented in Section 5.

Below, the IDPD and IDPID controllers that were applied to bipedal robots are presented. In the simulation calculations, first of all, the assumption that the dynamic model is accurately determined and that the motion trajectory is designed to ensure the relative equilibrium condition was applied to the IDPD controller. Next, in the dynamical model of the robot, it was assumed that there were noisy or uncertain forces, such as friction and/or damping at the joints. In this case, the IDPD controller needed to have additional noise cancellation capability, so with the addition of the integral component in the control law, the IDPD controller became the IDPID controller and was applied for simulation calculations.

4.1. IDPD Controller

Assuming that the robot dynamical model was fully and accurately determined and that the motion trajectory was designed to ensure the relative equilibrium condition, ignoring noise and external forces, the dynamic model applied to the IDPD controller is the differential equations of motion of the robot (7). Applying the control law IDPD, the system of equations to determine the control force/moment is shown as follows:

$$U_{(7 \times 1)} = M(q)_{(7 \times 7)} u_{(7 \times 1)} + \psi(q, \dot{q})_{(7 \times 1)} + G(q)_{(7 \times 1)} + Q(q)_{(7 \times 1)}; \quad U_1 = 0 \quad (11)$$

Here, u is determined based on the proportional derivative control law (PD) as follows:

$$u_{(7 \times 1)} = \ddot{q}_{d(7 \times 1)} + K_v_{(7 \times 7)} \dot{e}_{(7 \times 1)} + K_p_{(7 \times 7)} e_{(7 \times 1)} \quad (12)$$

$$\begin{cases} e_{(7 \times 1)} = [e_1, e_2, \dots, e_7]^T; & e_i = q_{di} - q_i \\ \dot{e}_{(7 \times 1)} = [\dot{e}_1, \dot{e}_2, \dots, \dot{e}_7]^T; & \dot{e}_i = \dot{q}_{di} - \dot{q}_i \end{cases} \quad (13)$$

$$\begin{cases} K_{P(7 \times 7)} = \text{diag}\{k_{p1}, k_{p2}, \dots, k_{p7}\}; & k_{pi} > 0, (i = 2, \dots, 7), k_{p1} = 0 \\ K_{V(7 \times 7)} = \text{diag}\{k_{v1}, k_{v2}, \dots, k_{v7}\}; & k_{vi} > 0 (i = 2, \dots, 7), k_{v1} = 0 \end{cases} \quad (14)$$

In which:

$q_{di}, \dot{q}_{di}, \ddot{q}_{di} (i = 1, \dots, 7)$ are the desired designed joint coordinates and their time derivatives, respectively; $q_i, \dot{q}_i, \ddot{q}_i (i = 1, \dots, 7)$ are the joint coordinates obtained by the control and their first and second derivatives with respect to time, respectively; k_{pi}, k_{vi} are the proportional and derivative gain coefficients, respectively; K_P, K_V are diagonal matrices. Since the coordinates q_1 are not controlled, $U_1 = 0$; for simplicity, we set $k_{p1} = 0, k_{v1} = 0$.

Substituting (12)–(14) into (11) to find out the system of canonical equations of the controller is shown as follows:

$$\ddot{e}_{(7 \times 1)} + K_V_{(7 \times 7)} \dot{e}_{(7 \times 1)} + K_P_{(7 \times 7)} e_{(7 \times 1)} = 0 \quad (15)$$

4.2. IDPID Controller

In the case of disturbance and unknown forces, it was assumed to be disturbance force and damping (frictional) force, as below:

$$Q_d = k_d [0, \cos q_2, \sin q_3, \cos 2q_4, \sin 2q_5, \cos q_6, \sin q_7]^T \tag{16}$$

$$Q_b = k_b [0, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6, \dot{q}_7]^T \tag{17}$$

Here Q_d, Q_b are vectors whose elements are the disturbance force and friction at the joints, respectively.

Applying the IDPID controller, the system of equations for determining the control force/torque (12) is rewritten as follows:

$$u_{(7 \times 1)} = \ddot{q}_{d(7 \times 1)} + K_V_{(7 \times 7)} \dot{e}_{(7 \times 1)} + K_P e_{(7 \times 1)} + K_I_{(7 \times 7)} \int_0^t e_{(7 \times 1)} d\tau \tag{18}$$

$$K_I_{(7 \times 7)} = \text{diag}\{k_{I1}, k_{I2}, \dots, k_{I7}\}; k_{Ii} > 0 (i = 2, \dots, 7), k_{I1} = 0 \tag{19}$$

The numerical simulation calculations of these controllers are presented in Section 6.

5. The Fuzzy Controller for Bipedal Robot

Normally, a controller has three main components—namely, input signal block, control algorithm block based on control law, and output signal block, which is the control quantity. In addition, there is a component of the fuzzy controller that has an invisible connection to it and plays a decisive role: the expert knowledge base. With a crisp controller, such as a robot controller, the input signal block is usually the motion trajectory required by the robot. The algorithm block calculates the control quantity based on the dynamical model of the system (transfer function, motion differential equation, equation of state, etc.)—for example, the IDPD, IDPID algorithms, as mentioned above. The output signal block is a controlled quantity defined from the algorithm block, such as the motor driving force and torque, and it can be converted in machine language, such as amperage, etc.

In the fuzzification block of fuzzy controller, the process of converting the physical values of the controlled variables into linguistic values was performed.

The fuzzy controller’s algorithm performed calculations from fuzzy inferences based on fuzzy rules.

In the output block, defuzzification occurred, and the language values of the control quantities were converted to physical values.

The knowledge base determined all the processes in the blocks of the fuzzy controller. Figure 11 shows the general structure of a control system for a bipedal robot, in which the control block is a fuzzy controller. In turn, the general structure of the fuzzy controller is shown in Figure 12. The controller’s algorithm block is the central block, called the inference mechanism, which performed fuzzy inference based on the fuzzy rule system.

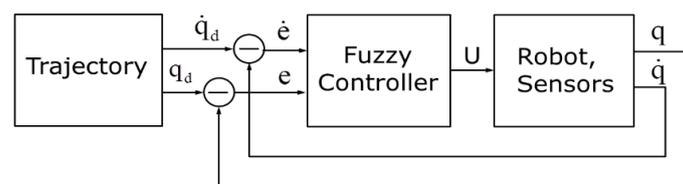


Figure 11. The general structure of a control system for a bipedal robot, in which the control block is a fuzzy controller.

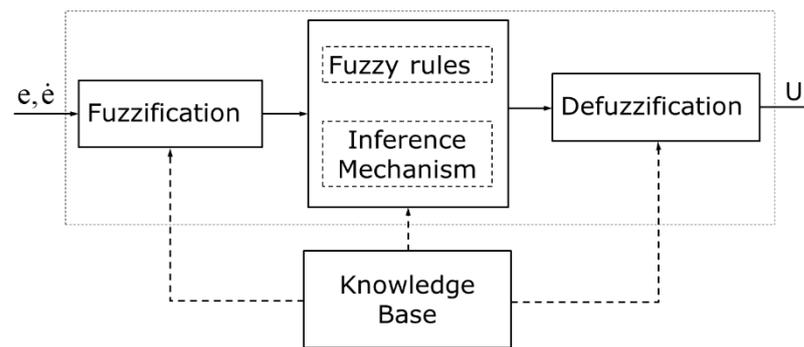


Figure 12. The general structure of a fuzzy controller of a bipedal robot.

The problem of fuzzy logic-based controllers is that there is no common rule or method for constructing control rules, as well as no rule or method for fuzzification and defuzzification. The processes of fuzzification, defuzzification, and building fuzzy rules for the controller are based on expert knowledge and depend entirely on the designer. Fortunately, fuzzy rules can be built and integrated from very simple basic fuzzy rules, in simple natural languages.

In the paper, the steps of fuzzification and building a fuzzy rule system, along with applying an inference method and defuzzification corresponding to the blocks in Figure 12, are presented below.

5.1. Fuzzification

5.1.1. The Input, Output Signals, and Physical Value Domains

The input signals were the position error $e(t)$ and velocity error $\dot{e}(t)$ of the joints generated from comparing quantities that define the robot's motion trajectory represented by the joint coordinates q_d and their derivative \dot{q}_d , with response values of the corresponding quantities, q, \dot{q} . These quantities were determined according to Equation (13). The output signal of the controller was the value of driving moments $U(t)$ at the driven joints, which ensured the motion trajectory of the robot—i.e., with the errors $e(t), \dot{e}(t)$ as small as possible, relative to the desired value—and ensures the relative equilibrium.

The driving moments $U(t)$ depended on the motion requirements of the robot as well as the geometrical, kinematical, and dynamical parameters of the robot. For crisp controllers in general, and IDPD and IDPID controllers, the computation of these moments is based on a dynamical model, which is a system of differential equations of motion, and can have kinematic constraints by errors of position $e(t)$ and velocity $\dot{e}(t)$. With the fuzzy controller applied in this paper, the calculation of $U(t)$ did not use the dynamical model but only relied on a fuzzy rule system built based on expert knowledge and on the errors $e(t), \dot{e}(t)$. To accomplish that, one of the first requirements was to determine the physical value domain of the control moments $U(t)$ as well as the physical value domain where the errors $e(t)$ and $\dot{e}(t)$ could occur. Using the minimum and maximum limit values, the physical value domains of errors $e(t), \dot{e}(t)$, and driving torque $U(t)$ are denoted as follows:

$$\begin{cases} e_i = [e_{imin}, e_{imax}] (rad), & i = 1, \dots, 7 \\ \dot{e}_i = [\dot{e}_{imin}, \dot{e}_{imax}] (rad.s^{-1}), & i = 1, \dots, 7 \\ U_i = [U_{imin}, U_{imax}] (Nm), & i = 1, \dots, 7 \end{cases} \quad (20)$$

5.1.2. Linguistic Variables and Scale Mapping of Input, Output Data

The linguistic variables used corresponded to the input and output signals, which are the errors of position $e(t)$ and velocity $\dot{e}(t)$ and the driving moments of the joints $U(t)$. Although the input signals of the fuzzy controller were not position and velocity but were their errors, for convenience of presentation, their linguistic variables were sometimes, for short, called linguistic variables of joint position and linguistic variables of joint velocity.

Linguistic variables have both physical and linguistic values. The physical values belonged to the physical value domain of linguistic variables and were determined by Equation (20). Let X be the universe of discourse of linguistic variables.

In the fuzzy controller, the input data of fuzzy operations and fuzzy inference were fuzzy sets. Therefore, the input data (physical values) of linguistic variables needed to be converted into suitable linguistic values (fuzzy sets).

The modeling of fuzzy sets from uncertain data posed the problem of information quantification. Thus, the discretization (quantization) of a universe of discourse into a certain number of segments was performed. The scale mapping of the input and output data was applied, which in order to scale the input and output crisp values, transformed the value domain of the input and output variables into the partitioned corresponding universe of discourse. Each segment of a universe of discourse was named as a generic element and formed a discrete universe of discourse. Then, in the discrete universe of discourse, the fuzzy sets were defined. The number of quantization levels has an important influence on the control performance and should therefore be large enough to give an appropriate approximation. With the fuzzy controller for the bipedal robot, the universes of discourse were discretized into five levels, denoted by $X_i (i = 1, \dots, 5)$. Thus, there were five corresponding fuzzy sets labeled as $F_i (i = 1, \dots, 5)$. Fuzzy sets have overlapping parts and were defined on the discrete universe of discourse. By discretizing the universe of discourse and partitioning the fuzzy set on it, a linguistic variable was associated with a term set that represented the levels of the discretized universe of discourse, and also represented fuzzy sets, which were defined on a partitioned universe of discourse. Thus, five corresponding terms of linguistic variables were used and denoted by: Negative Big (NB), Negative Small (NS), Zero (Z), Positive Small (PS), and Positive Big (PB). These terms represented the discretized values of the universe of discourse, at which the membership function of the corresponding fuzzy sets took the value 1. These terms are also often used to name the corresponding fuzzy sets.

Table 2 shows the levels of the discretized universe of discourse, fuzzy sets, linguistic value terms, and their notations, respectively.

Table 2. Description of levels of the discretized universe of discourse, fuzzy sets, linguistic value terms, and notations, of the inputs and outputs.

No.	Levels X_i	Fuzzy Set F_i	Name of Linguistic Value	Notation of Linguistic Value
1	X_1	F_1	Big Negative	NB
2	X_2	F_2	Small Negative	NS
3	X_3	F_3	Zero	Z
4	X_4	F_4	Small Positive	PS
5	X_5	F_5	Big Positive	PB

With the robot's fuzzy controller, the linguistic variables of the input data affected the output data simultaneously. Therefore, for simplicity, while ensuring controller performance, the universe of the discourse of linguistic variables was discretized to the same number of levels, and with a uniform scale mapping. That explains the use of the same notation (X) for the universe of discourse, the same notations (F_i) for fuzzy sets, and the same with the linguistic value terms for linguistic variables. However, the physical value of the linguistic variable and the partitioned universe of discourse was determined by Equation (20), respectively.

5.1.3. Set Up Membership Function of Linguistic Variables

To perform fuzzy operations and fuzzy inference, the input data must be assigned membership values of one or more fuzzy sets into which the universe of discourse has been partitioned. There are different methods for assigning membership values or membership functions to fuzzy variables. Here the membership functions of the right trapezoid, triangle, and left trapezoid were used to represent the membership values of the linguistic variables

$e(t)$, $\dot{e}(t)$, and $U(t)$. Figure 13 shows the membership functions of fuzzy sets of linguistic variables, defined on the partitioned universe of discourse, where the physical values of linguistic variables at the discretized points of the universe of discourse were determined as follows: $NB = x_{i,\min}$; $NS = 0.75x_{i,\min} + 0.25x_{i,\max}$; $Z = 0.5x_{i,\min} + 0.5x_{i,\max}$; $PS = 0.25x_{i,\min} + 0.75x_{i,\max}$; $PB = x_{i,\max}$.

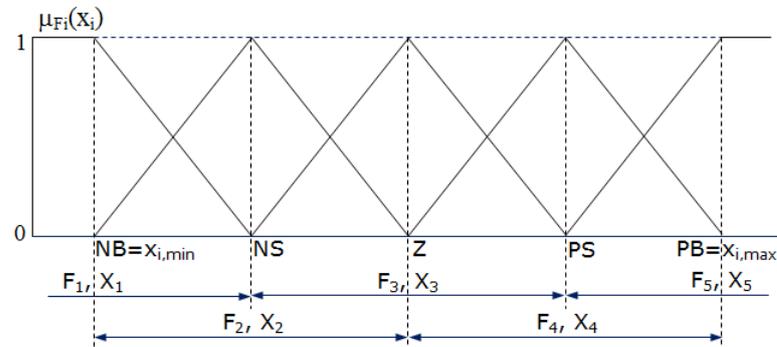


Figure 13. The membership function of position error, velocity error, and driving moment of joint i : $x_i(x_i = e_i, \dot{e}_i, \text{ and } U_i)$.

By applying the membership function types mentioned above and from Figure 13, it was easy to determine the Support (defined domain) of the fuzzy sets of a linguistic variable $\text{Supp}(F_i)(i = 1, \dots, 5)$. In addition, given the physical values of the language variable, it was easy to calculate its membership value as below.

The membership value of the linguistic variable of the fuzzy set F_1 , Support $\text{Supp}(F_1) = [NB, NS]$:

$$\mu_{F_1}(x_i) = \begin{cases} 1 & \text{if } x_i \leq NB \\ \frac{NS-x_i}{NS-NB} & \text{if } NB \leq x_i \leq NS \\ 0 & \text{if } x_i \geq NS \end{cases} \quad (21)$$

The membership value of the linguistic variable of the fuzzy set F_5 , Support $\text{Supp}(F_5) = [PS, PB]$:

$$\mu_{F_5}(x_i) = \begin{cases} 0 & \text{if } x_i \leq PS \\ \frac{x_i-PS}{PB-PS} & \text{if } PS \leq x_i \leq PB \\ 1 & \text{if } x_i \geq PB \end{cases} \quad (22)$$

For the membership value of the linguistic variable of the fuzzy set $F_i(i = 2,3,4)$, let us call its Support as $\text{Supp}(F_i) = [a_1, a_2]$:

$$\mu_{F_i}(x_i) = \begin{cases} \frac{x_i-a_1}{a_0-a_1} & \text{if } a_1 \leq x_i \leq a_0; a_0 = \frac{a_1+a_2}{2} \\ \frac{a_2-x_i}{a_2-a_0} & \text{if } a_0 \leq x_i \leq a_2 \\ 0 & \text{if } x_i \leq a_1 \text{ or } x_i \geq a_2 \end{cases} \quad (23)$$

5.2. Inference Mechanism

5.2.1. Fuzzy Rule Base System

In the inference mechanism block, the output quantity, the driving torque $U(t)$ was calculated based on the input variables, the joint position error $e(t)$, and the velocity error $\dot{e}(t)$. The problem is that a fuzzy rule system needs to be built entirely based on expert knowledge about the controlled object. A fuzzy rule system is characterized by a set of language statements usually represented as “if-then” rules. As is known, there are five fuzzy sets used to represent an input variable and to determine their membership values, as shown by Equations (21)–(23). Thus, with two input language variables, $e(t)$ and $\dot{e}(t)$, and fuzzy rules according to the “if-then” rule, there will be $5^2 = 25$ statements forming

the fuzzy rule base system. Based on expert knowledge on the dynamics of bipedal robots, this system of 25 fuzzy rules is proposed as follows:

$$\left\{ \begin{array}{l} \text{If } e(t) \text{ is NB and } \dot{e}(t) \text{ is NB then } u(t) \text{ is PB or} \\ \text{If } e(t) \text{ is NB and } \dot{e}(t) \text{ is NS then } u(t) \text{ is PB or} \\ \vdots \\ \text{If } e(t) \text{ is PB and } \dot{e}(t) \text{ is PB then } u(t) \text{ is NB} \end{array} \right. \quad (24)$$

The fuzzy rule base system is shown in Table 3. Figure 14 depicts the output control surface describes the relationship between $e(t)$, $\dot{e}(t)$, and $U(t)$ based on the synthesis rule of the fuzzy rule base system for the fuzzy controller.

Table 3. Fuzzy rule base system.

$U(t)$	Position Error $e(t)$					
	Velocity Error $\dot{e}(t)$	NB	NS	Z	PS	PB
NB		PB	PB	PB	PS	Z
NS		PB	PS	PS	Z	NS
Z		PB	PS	Z	NS	NB
PS		PS	Z	NS	NS	NB
PB		Z	NS	NB	NB	NB
NB		PB	PB	PB	PS	Z

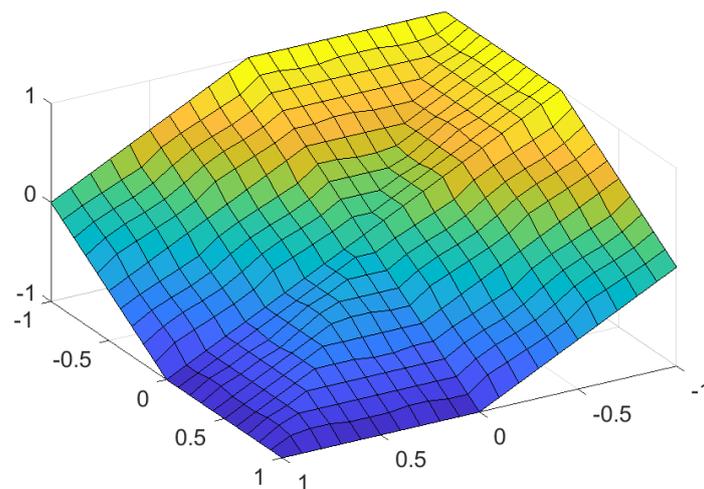


Figure 14. The output control surface describes the relationship between $e(t)$, $\dot{e}(t)$, and $U(t)$ based on the synthesis rule of the fuzzy rule base system for the fuzzy controller.

The equation for the composition (synthesis rule) of the base of fuzzy rules for a fuzzy controller is as follows:

$$R : \bigcup_{i=1}^n R_i \quad (25)$$

$$R_i : [(e(t) \cup \dot{e}(t)) \rightarrow u(t)], i = 1, \dots, 25$$

5.2.2. Compositional Rule of Inference

Based upon the fuzzy rule base system, the inference methods were developed. The inference was determined by the “implication operator” and “composition operator” as follows. The implication operator was implemented based on the application of the Mamdani implication method, the MIN operator. The composition operator was also based on the Mamdani method, which is called the MAX–MIN operator. At each time, the input control signals of position error $e(t)$ and velocity error $\dot{e}(t)$ assumed that: e_i is F_{ei}^l and \dot{e}_i is

$F'_{dei} (de_i = \dot{e}_i)$. Where F'_{ei}, F'_{dei} are the linguistic values of the linguistic variables $e(t), \dot{e}(t)$, respectively. Then the fuzzy set F'_u of the output linguistic variable was determined by the fuzzy implication relation as follows:

$$F'_u = \bigcup_{i=1}^{25} F'_{u_i}; \quad F'_{u_i} = (F'_{ei}, F'_{dei}) \circ R_i; \quad (de_i = \dot{e}_i) \tag{26}$$

The Mamdani’s minimum operation was used for the implication:

$$\mu_{F'_{u_i}}(u) = \alpha_i \wedge \mu_{F_{u_i}}(u); \quad \alpha_i = \mu_{F'_{ei}}(e_i) \wedge \mu_{F'_{dei}}(\dot{e}_i) \tag{27}$$

The aggregate rule of the entire fuzzy rule base system was determined by the membership value based on Mamdani’s MAX–MIN operator:

$$\mu_{F'_u}(u) = \bigvee_{i=1}^{25} [\alpha_i \wedge \mu_{F_{u_i}}(u)] = \bigvee_{i=1}^{25} \mu_{F'_{u_i}}(u) \tag{28}$$

5.3. Defuzzification

Defuzzification is the process by which mapping is performed to convert the membership function value of the output variable into crisp physical values. Here it determined the physical value of the driving moment from its membership function value in the fuzzy set F'_u . In this paper, the well-known “center of area” method was used for defuzzification in the fuzzy controller of a bipedal robot. This method is easy to apply, involving all output fuzzy sets of the compositional rule of inference, and produces high-precision results.

$$u' = \frac{\int_{S'} u \mu_{F'_u}(u) du}{\int_{S'} \mu_{F'_u}(u) du} \tag{29}$$

where, u' is the actual output value, taken according to the coordinates of the central point of the region, formed by the membership function line and the horizontal axis; S' is the defined domain of the fuzzy set F'_u .

6. Simulation Results

This section presents numerical simulations applying IDPD, IDPID, and fuzzy controllers to control the robot to take human-like steps. In Section 2, the design of motion trajectories was presented, and the typical motion trajectories were selected for the three steps of the robot—namely the starting step at the first stage, the steady step, and the slow step gradually to a stationary state. The motion trajectories were saved as files.txt, as mentioned in Section 2. To confirm the validity of the fuzzy controller, it was first assumed that the robot’s dynamical model was fully and accurately determined. Then the IDPD controller was applied. The system of differential Equation (7) representing the motion of the robot is presented in Section 3, with the assumption that it is correct and that it is used for this IDPD controller.

In fact, errors always exist in the dynamic model of a robot. The first cause is deviation due to measurement and calculation, followed by uncertain factors such as unknown external force, change in the interaction between robot and environment, noise, etc.

The IDPID controller, which contains a disturbance-limiting and disturbance-eliminating element, was applied. In the system of differential equations of robot motion (7), it is assumed that there are uncertain factors such as friction force, noise, etc.

The proposed fuzzy controller did not use a dynamic model when determining control variables, which are driving moments in the driven joints.

The results of applying the IDPD and IDPID controllers were used to verify the proposed fuzzy controller.

6.1. Simulation of IDPD Controller

The mathematical model of the IDPD controller is presented in Section 4. Here the matrix of position gain and derivative coefficients were selected as follows:

$$K_P = \text{diag}\{0, 6400, 6400, 6400, 6400, 6400, 6400\} \quad (30)$$

$$K_V = \text{diag}\{0, 160, 160, 160, 160, 160, 160\} \quad (31)$$

The simulation results of the IDPD controller for the starting step, steady step, and final step are shown in Figures 15a, 16a and 17a, respectively, in which the motion of each joint i ($i = 1, \dots, 7$) changes with time and is represented, respectively, in each graph, including the designed desired trajectory mentioned in Section 2 ($qd1, \dots, qd7$), the received trajectory (real trajectory) from the control result ($qr1, \dots, qr7$), and the errors ($dq1, \dots, dq7$) between the desired trajectory and the real trajectory.

6.2. Simulation of IDPID Controller

Assume that there are errors in the robot dynamics model—for example, suppose that the system of motion differential equations has unknown disturbance and frictional forces, as shown in Equations (16) and (17). In this case, the matrix of position gain and integral and derivative coefficients are selected as follows:

$$K_P = \text{diag}\{0; 16,000; 8100; 8100; 8100; 22,000; 22,000\} \quad (32)$$

$$K_V = \text{diag}\{0; 250; 180; 180; 180; 280; 280\} \quad (33)$$

$$K_I = \text{diag}\{0; 1600; 810; 810; 810; 2200; 2200\} \quad (34)$$

The simulation results of the IDPID controller for the starting step, steady step, and final step are shown in Figures 15b, 16b and 17b, respectively, in which the motion of each joint i ($i = 1, \dots, 7$) changes with time and is represented, respectively, in each graph, including the designed desired trajectory mentioned in Section 2 ($qd1, \dots, qd7$), the received trajectory (real trajectory) from the control result ($qr1, \dots, qr7$), and the errors ($dq1, \dots, dq7$) between the desired trajectory and the real trajectory.

6.3. Simulation of Fuzzy Controller

For a fuzzy controller to work efficiently and with high accuracy, a suitable system of fuzzy rules is first built. This was discussed in Section 5. In addition, the domain of physical values of input and output quantities also plays a particularly important role. For the fuzzy controller for the bipedal robot studied here, the physical domain of the input and output variables was checked and selected based on expert knowledge and is presented in Table 4.

Table 4. The physical value domain of input and output quantities of the bipedal robot's fuzzy controller.

Link	e_i	\dot{e}_i	U_i
1	[−0.0, 0.0] (rad)	[−0.0, 0.0] (rad/s)	[−0.0, 0.0] (Nm)
2	[−0.001, 0.001] (rad)	[−0.2, 0.2] (rad/s)	[−180, 180] (Nm)
3	[−0.001, 0.001] (rad)	[−2.4, 2.4] (rad/s)	[−95, 95] (Nm)
4	[−0.001, 0.001] (rad)	[−3.0, 1.0] (rad/s)	[−50, 30] (Nm)
5	[−0.001, 0.001] (rad)	[−0.3, 0.3] (rad/s)	[−38, 38] (Nm)
6	[−0.001, 0.001] (rad)	[−0.4, 0.3] (rad/s)	[−12, 12] (Nm)
7	[−0.001, 0.001] (rad)	[−0.8, 0.8] (rad/s)	[−50, 50] (Nm)

The simulation results of the fuzzy controller for the starting step, steady step, and final step are shown in Figures 15c, 16c and 17c, respectively, in which the motion of each joint i ($i = 1, \dots, 7$) changes with time and is represented, respectively, in each graph,

including the designed desired trajectory mentioned in Section 2 ($qd1, \dots, qd7$), the received trajectory (real trajectory) from the control result ($qr1, \dots, qr7$), and the errors ($dq1, \dots, dq7$) between the desired trajectory and the real trajectory.

Below are the simulation results presented:

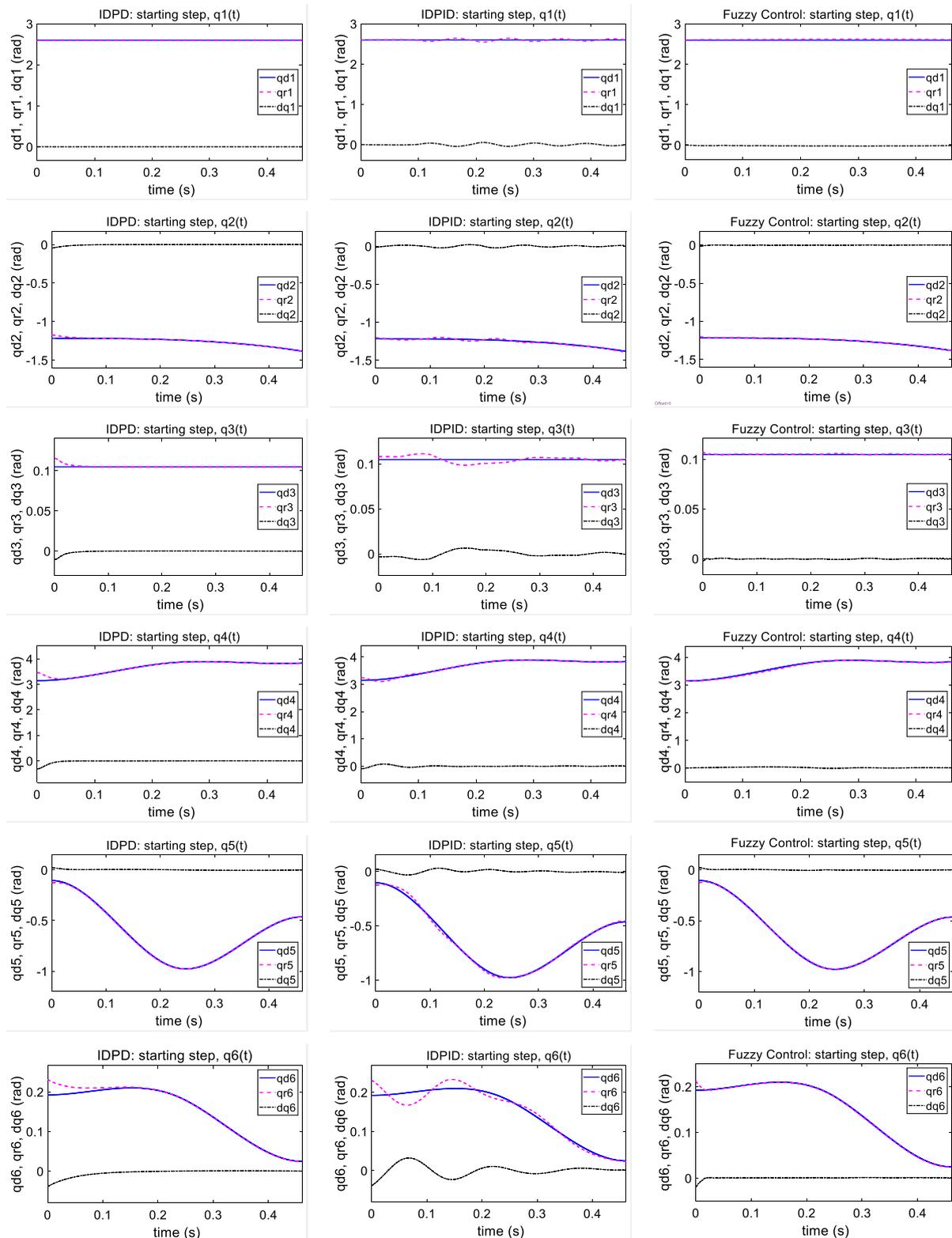


Figure 15. Cont.

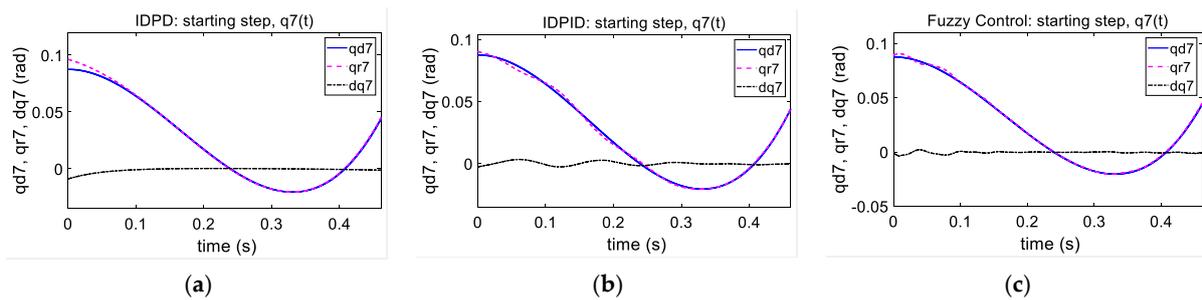


Figure 15. Simulation results of controllers for bipedal robots at the starting step: (a) IDPD controller; (b) IDPID controller; (c) fuzzy controller.

Figure 15 shows the simulation results of the controllers for the bipedal robot at the starting step, during which the speed increased gradually. The figures in the right column, Figure 15a, are the results of the IDPD controller, which assumed that the dynamics model of the robot was correct.

The figures in the middle column, Figure 15b, are the results of the IDPID controller, assuming the dynamics model of the robot was not precisely and completely defined, i.e., there were additional friction and disturbance forces, collectively known as the disturbance force. Then, in the control law of IDPID, there were additional elements that reduced or eliminated the disturbance forces.

The figures in the right column, Figure 15c, are the result of the fuzzy controller. The fuzzy controller determined the control quantities, which were the driving moments at the driven joints based on the fuzzy law system (24), fuzzy composition rules, inference and defuzzification rules (25)–(29), and physical value domain in Table 4.

On each figure, three graphs are drawn: the blue line represents the desired joint coordinates q_{di} ($q_{d[i]}$, $i = 1, \dots, 7$); the red line represents the actual joint coordinates q_{ri} ($q_{r[i]}$, $i = 1, \dots, 7$); The black line represents the deviation of the joint coordinate value between the actual and desired trajectory dq_i ($dq[i]$, $i = 1, \dots, 7$). To further verify the controller's ability to adjust, the initial values of the control simulation are often made different from the correct values, and the controller soon returns the system to the correct position, follows the desired trajectory, and is stable. The average error between the desired joint coordinate value and the actual coordinate in stable motion is in the range of $\sim 10^{-3}$ radians. For mobile robots in general and bipedal robots, such errors are very small because the robot's walking accuracy is obviously not as demanding as the accuracy of robots applied in mechanical processing.

In a similar way, results are shown for the simulation of the controllers for the bipedal robot at a steady walk, Figure 16, and walk to a stop state, Figure 17.

Figure 16 shows the simulation results of controllers for a bipedal robot at a steady walk, during which the horizontal velocity of the robot body was almost constant.

The calculation results obtained the average error between the desired joint coordinate value and the actual coordinates at the step in the steady period, which is also in the range of $\sim 10^{-3}$ radians. The steady step simulations were similar to those of the starting step, including three graphs. The graph can be observed to see the accuracy and reliability of the controllers. The fuzzy controller can be as precise as the controller that is based on the exact dynamic model IDPD. The graph observation shows that the IDPID controller based on the dynamic model including noise was slightly less accurate.

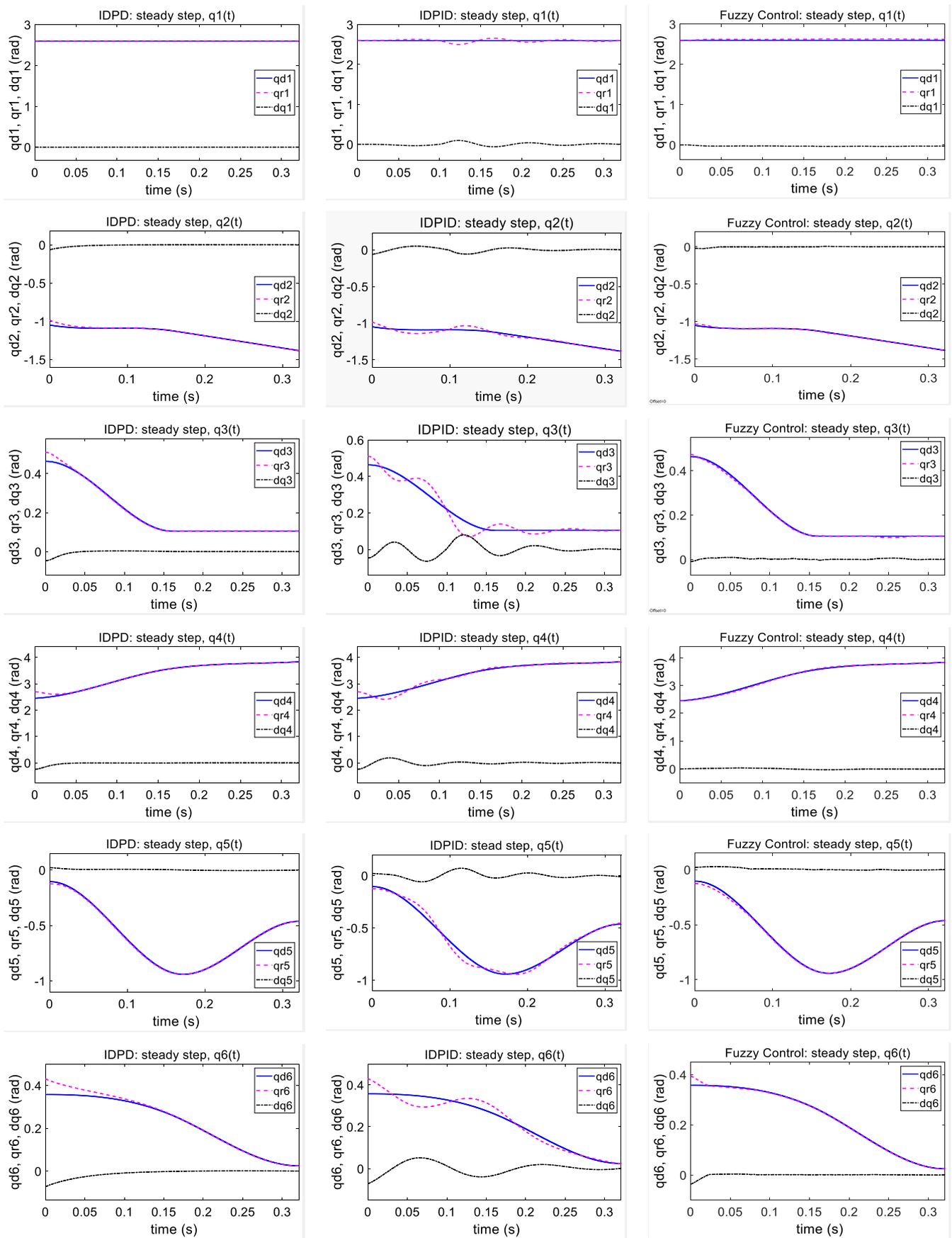


Figure 16. Cont.

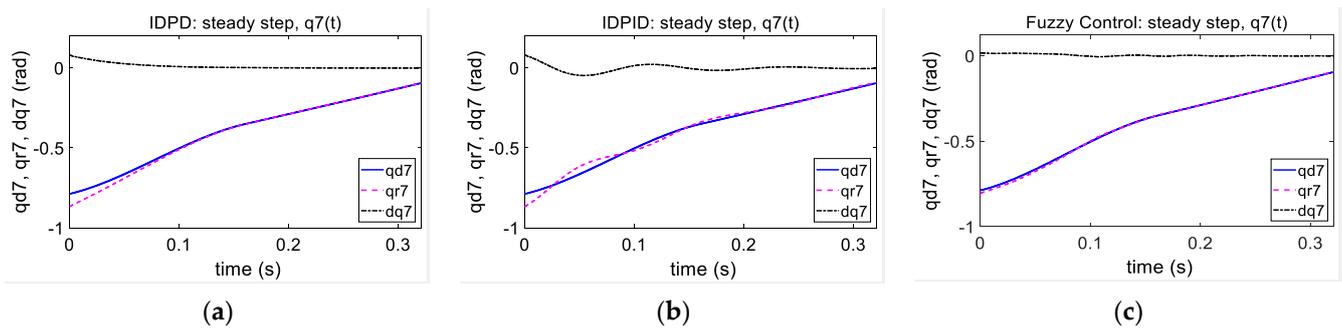


Figure 16. Simulation results of controllers for bipedal robots at the steady step: (a) IDPD controller; (b) IDPID controller; (c) fuzzy controller.

Figure 17 shows the controller simulation results for the bipedal robot at the last step when it comes to a stationary state.

From the simulation results, it can be seen that:

The IDPD controller was applied with the assumption that the robot dynamics model was accurate, so it gave accurate results.

Even though the IDPID controller had an element to limit and eliminate the disturbance force, the control results still have less accuracy.

The fuzzy controller with the proposed fuzzy law allowed accurate results to be obtained.

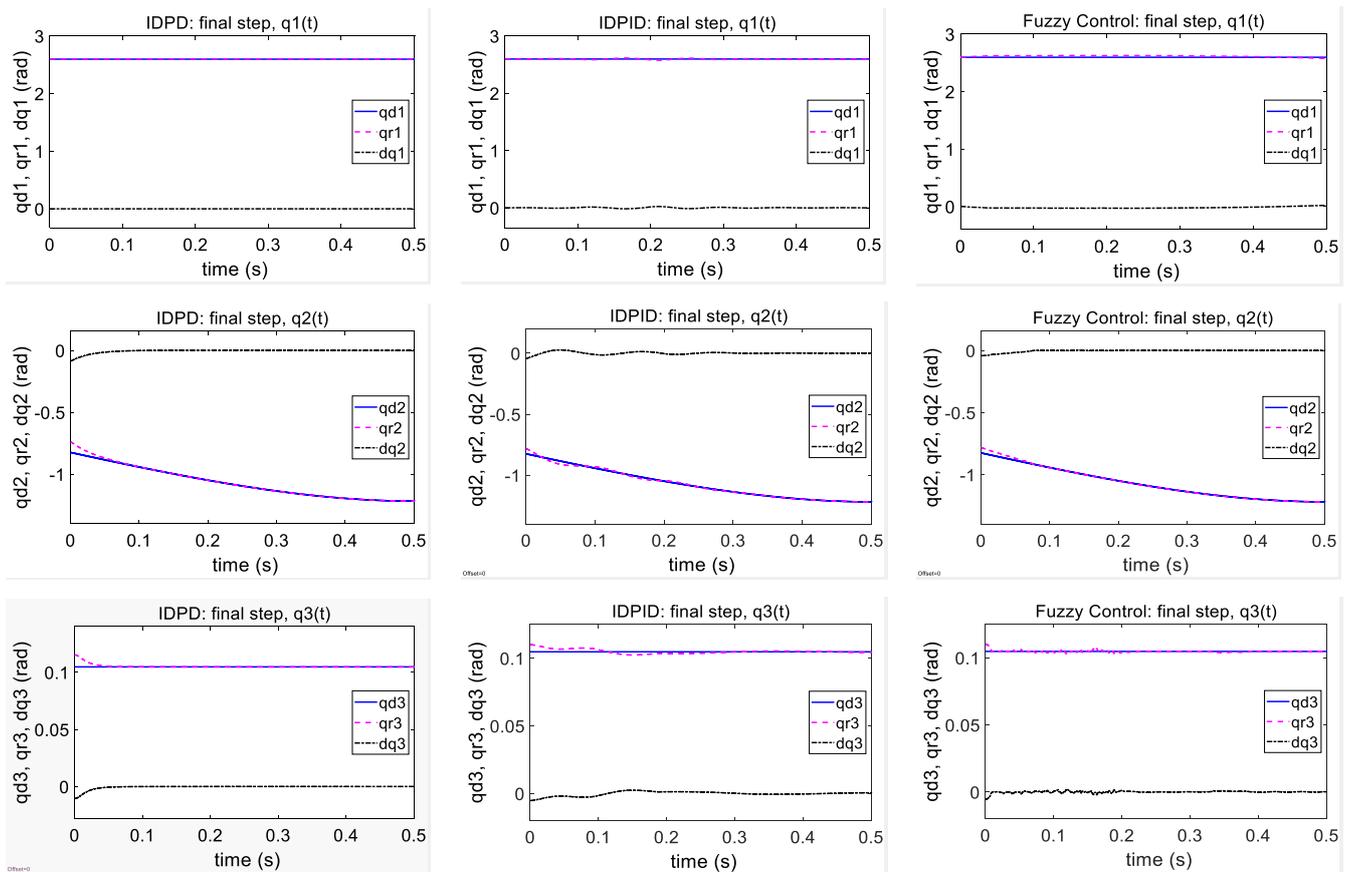


Figure 17. Cont.

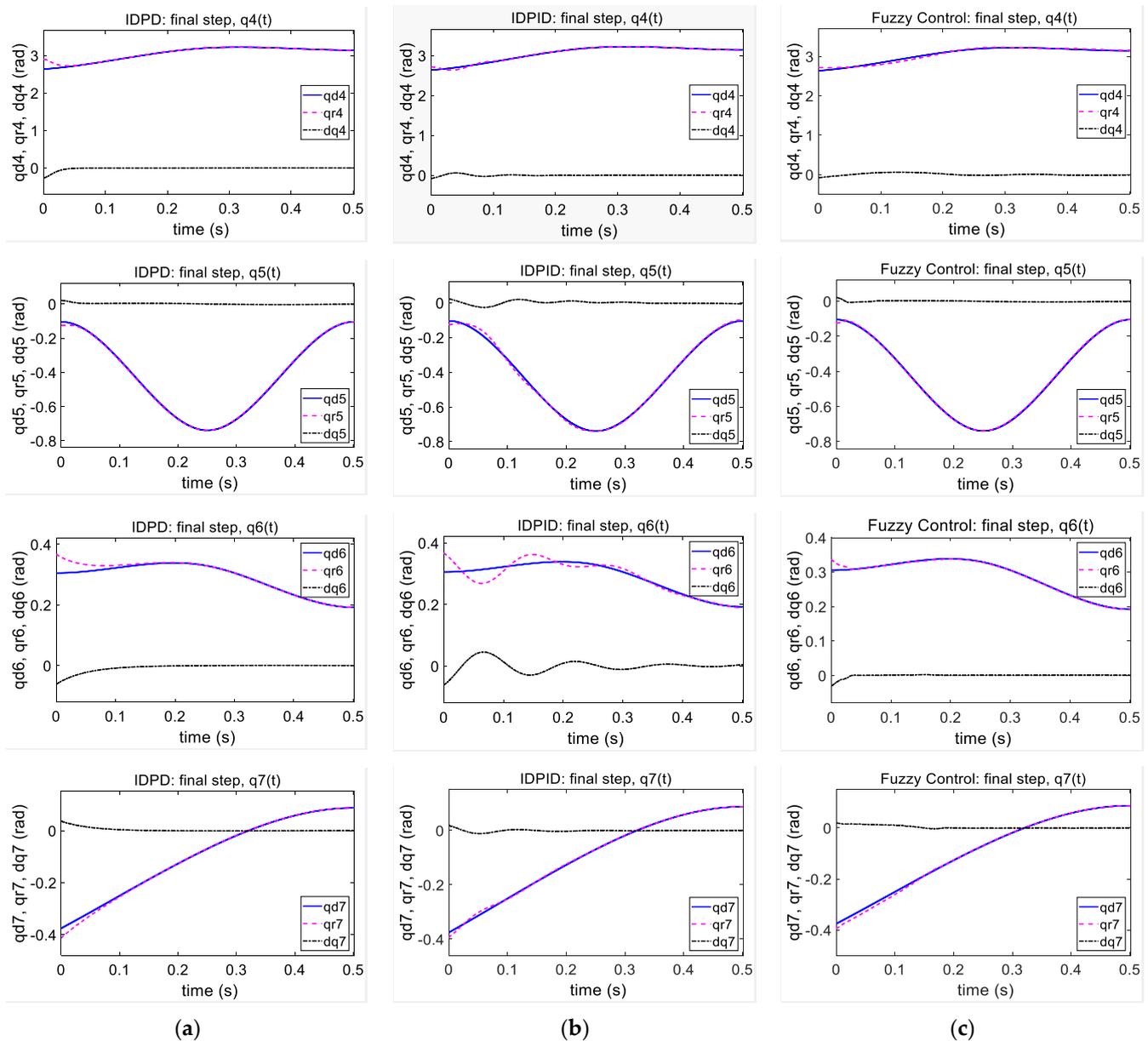


Figure 17. Simulation results of controllers for bipedal robots at the final step: (a) IDPD controller; (b) IDPID controller; (c) fuzzy controller.

The calculation results show that, during stable motion at all three stages of motion, the average error between the desired joint coordinate value and the actual coordinate was $\leq 10^{-3}$ radians. As stated, it is practically impossible to determine the exact dynamics model. On the other hand, IDPD and IDPID controllers had to perform computations of cumbersome and complex mathematical expressions of the robot’s dynamic model. When there are many abnormal and complex disturbances, it is more difficult for an IDPID controller to ensure control accuracy.

The calculation expressions based on the fuzzy rule system and fuzzy inference are very simple and easy.

7. Conclusions

With the proposed fuzzy rule system and the applied fuzzy inference rule, the fuzzy controller applied in the paper provided accurate results. Computational expressions based on fuzzy rules and fuzzy inference rules were very simple and easy to implement.

The robot model used in the simulation had parameters close to those of a real human and kinematic parameters such as velocity, acceleration, etc. that were also close to human activities. Therefore, it is feasible to apply the research results in practice.

The dynamic model of human-like bipedal robots is quite diverse, and the dynamic quantities often change with the operations. The problem of controlling the bipedal robot needs to be such that it can simultaneously move according to the set trajectory and satisfy the balance condition because the robot's two-legged structure is less stable. The crisp controllers based on dynamical models require large computations. On the other hand, the dynamical model of a bipedal robot often changes, so with a closed computational model it will be difficult to adapt and flexibly adjust the control quantity to flexibly respond to multi-tasking in a human-like bipedal robot.

The construction of the fuzzy rule system took into account the influence and relation to the uncontrolled coordinate, which is the angle between the support foot and the ground. Therefore, the fuzzy law system allowed the calculation and adjustment of the driving forces at the controlled joints, so as to both meet the robot's motion trajectory and ensure the relative balance.

A human, based on natural reasoning and through training, can operate and balance easily in many complicated postures and states.

The high cost in time and effort in building a fuzzy controller is building the fuzzy rule system (24) and defining the physical value domain in Table 4. However, once the fuzzy rule system has been created and the physical value domain of language variables has been determined, the implementation of fuzzy operations is simple and easy, allowing real-time control requirements to be met. Equations from (21) to (29) are easy to apply and implement.

During the numerical experiment to find the physical value domain in Table 4, a number of tests based on Modern Evolutionary Synthesis were applied, which provided good results. Applying this algorithm to find the physical value domain and to reduce the calibration time is one of the next studies needed to perfect the fuzzy controller.

Regarding the fuzzy rule system, it is a rule system built based on natural reasoning, much like the reasoning of humans. To be able to control a human-like bipedal robot that operates flexibly and stably like a human, it is necessary to have a controller capable of operating based on human natural reasoning and at the same time capable of being trained in a way similar to a human. This indicates that the development direction of fuzzy logic-based control is integrated with artificial intelligence, combining fuzzy controllers capable of deep learning, reinforcement learning, ant colony algorithms, Modern Evolutionary Synthesis, and other intelligent algorithms.

Author Contributions: Conceptualization, P.B.K.; methodology, software, formal analysis, investigation, writing—original draft preparation, writing—review and editing, P.B.K. and H.N.X.; validation, P.B.K.; layout editing, H.N.X.; supervision, P.B.K. All authors have read and agreed to the published version of the manuscript.

Funding: The authors received no financial support for the research, authorship, and publication of this article.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Siciliano, B.; Khatib, O. *Modeling and Control for Efficient Bipedal Walking Robots*; Springer: Berlin/Heidelberg, Germany, 2009.
2. Vukobratovic, M.; Borovac, B.; Surla, D.; Stokic, D. *Biped Locomotion: Dynamics, Stability, Control and Application*; Springer: Berlin/Heidelberg, Germany, 1990.

3. Chevallereau, C.; Bessonnet, G.; Abba, G.; Aoustin, Y. *Bipedal Robots: Modeling, Design and Walking Synthesis*; John Wiley & Sons: New York, NY, USA, 2009.
4. Buschmann, T. Simulation and Control of Biped Walking Robots. Ph.D. Thesis, Fakultät für Maschinenwesen, Technischen Universität München, Garching, Germany, 2010.
5. Bezerra, C.A.D.; Zampieri, D.E. Biped Robots: The State of Art. In *International Symposium on History of Machines and Mechanisms*; Kluwer Academic Publishers: Cassino, Italy, 2004; pp. 371–389.
6. Gupta, S.; Kumar, A. A brief review of dynamics and control of underactuated biped robots. *Adv. Robot.* **2017**, *31*, 607–623. [[CrossRef](#)]
7. Ficht, G.; Behnke, S. Bipedal Humanoid Hardware Design: A Technology Review. *Curr. Robot. Rep.* **2021**, *2*, 201–210. [[CrossRef](#)]
8. Katja, M.; Karsten, B. *Modeling, Simulation and Optimization of Bipedal Walking*; Springer: Berlin/Heidelberg, Germany, 2013.
9. Bajrami, X.; Dermaku, A.; Shala, A.; Likaj, R. Kinematics and dynamics modelling of the biped robot. *IFAC Proc. Vol.* **2013**, *46*, 69–73. [[CrossRef](#)]
10. Reher, J.; Ames, A.D. Dynamic Walking: Toward Agile and Efficient Bipedal Robots. *Annu. Rev. Control. Robot. Auton. Syst.* **2021**, *4*, 535–572. [[CrossRef](#)]
11. Hernández-Santos, C.; Rodriguez-Leal, E.; Soto, R.; Gordillo, J.L. Kinematics and Dynamics of a New 16 DOF Humanoid Biped Robot with Active Toe Joint. *Int. J. Adv. Robot. Syst.* **2012**, *9*, 190. [[CrossRef](#)]
12. Corral, E.; García, M.J.G.; Castejon, C.; Meneses, J.; Gismeros, R. Dynamic Modeling of the Dissipative Contact and Friction Forces of a Passive Biped-Walking Robot. *Appl. Sci.* **2020**, *10*, 2342. [[CrossRef](#)]
13. Rameez, M.; Khan, L.A. Modeling and Dynamic Analysis of the Biped Robot. In Proceedings of the 5th International Conference on Control, Automation and Systems, Busan, Korea, 13–16 October 2015; pp. 1149–1153.
14. Sugihara, T.; Morisawa, M. A survey: Dynamics of humanoid robots. *Adv. Robot.* **2020**, *34*, 1338–1352. [[CrossRef](#)]
15. Kljuno, E.; Williams, R.L. Humanoid Walking Robot: Modeling, Inverse Dynamics, and Gain Scheduling Control. *J. Robot.* **2010**, *2010*, 278597. [[CrossRef](#)]
16. Buschmann, T.; Lohmeier, S.; Ulbrich, H.; Pfeiffer, F. Dynamics Simulation for a Biped Robot: Modeling and Experimental Verification. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 2673–2678.
17. Tang, Z.; Zhou, C.; Sun, Z. Trajectory planning for smooth transition of a biped robot. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003; pp. 2455–2460.
18. Jo, H.S.; Mir-Nasiri, N.; Jayamani, E. Design and Trajectory Planning of Bipedal Walking Robot with Minimum Sufficient Actuation System. *Int. J. Mech. Mechatron. Eng.* **2009**, *3*, 1399–1405.
19. Zhang, R.; Vadakkepat, P. Motion Planning of Biped Robot Climbing Stairs. Available online: <http://ai.stanford.edu/~jrxzhang/Motion%20Planning%20of%20Biped%20Robot%20Climbing.pdf> (accessed on 21 September 2021).
20. Dau, V.-H.; Chew, C.M.; Poo, A.N. Optimized Joint-Torques Trajectory Planning for Bipedal Walking Robots. In Proceedings of the 2008 IEEE International Conference on Robotics, Automation and Mechatronics, Chengdu, China, 21–24 September 2008.
21. Huang, Q.; Arai, H. Planning Walking Patterns for a Biped Robot. *IEEE Trans. Robot. Autom.* **2021**, *17*, 280–289. [[CrossRef](#)]
22. Hashimoto, K.; Takezaki, Y.; Lim, H.; Takanishi, A. Walking stabilization based on gait analysis for biped humanoid robot. *Adv. Robot.* **2017**, *27*, 541–551. [[CrossRef](#)]
23. Vukobratovic, M.; Juricic, D. Contribution to the synthesis of biped gait. *IFAC Proc. Vol.* **1968**, *2*, 469–478. [[CrossRef](#)]
24. Vukobratovic, M.; Borovac, B.A. Zero-Moment Point—Thirty Five Years of its Life. *J. Vib. Control. Int. J. Hum. Robot.* **2004**, *1*, 157–173. [[CrossRef](#)]
25. Gabriel, T.; Han, M. Control of a humanoid robot based on the ZMP method. *IFAC Proc. Vol.* **2008**, *41*, 3065–3069. [[CrossRef](#)]
26. Erbatur, K.; Kurt, O. Natural ZMP Trajectories for Biped Robot Reference Generation. *IEEE Trans. Ind. Electron.* **2009**, *56*, 835–845. [[CrossRef](#)]
27. Usman, M.; Sajid, M.; Uddin, E.; Ayaz, Y. Investigation of Zero Moment Point in a Partially Filled Liquid Vessel Subjected to Roll Motion. *Appl. Sci.* **2020**, *10*, 3992. [[CrossRef](#)]
28. Kurt, O.; Erbatur, K. Biped Robot Reference Generation with Natural ZMP Trajectories. In Proceedings of the 9th IEEE International Workshop on Advanced Motion Control, Istanbul, Turkey, 27–29 March 2006; pp. 403–410.
29. Jiang, C.; Gao, J.; Shi, X.; Tian, D.; Huang, Q. Foot trajectory planning of bipedal walking robot based on a uniform acceleration method. In Proceedings of the 2019 IEEE International Conference on Advanced Robotics and its Social Impacts, Beijing, China, 31 October–2 November 2019; pp. 251–255.
30. Dekker, M.H.P. Zero-Moment Point method for stable biped walking. *Eindh. Univ. Technol.* **2009**, *2009*, 1–15.
31. Caballero, R.; Armada, M.A.; Alarcón, P. Methodology for Zero-moment Point Experimental Modeling in the Frequency Domain. *J. Vib. Control.* **2006**, *12*, 1385–1406. [[CrossRef](#)]
32. Wu, G.; Xiao, X.; Liu, L.; Luo, W.; Ye, X.; Yang, Y. *Stable Gait Planning for Biped Robot's Lower Limb Based on Natural ZMP Trajectories*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 331–341.
33. Erbatur, K.; Koca, Ö.; Taşkıran, E.; Yılmaz, M.; Seven, U. ZMP Based Reference Generation for Biped Walking Robots. *Int. J. Mech. Mater. Eng.* **2009**, *3*, 853–860.
34. Van Toan, N.; Khoi, P.B. A control solution for closed-form mechanisms of relative manipulation based on fuzzy approach. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1–11. [[CrossRef](#)]

35. Phan, K.B.; Ha, H.T.; Hoang, S.V. Eliminating the Effect of Uncertainties of Cutting Forces by Fuzzy Controller for Robots in Milling Process. *Appl. Sci.* **2020**, *10*, 1685. [[CrossRef](#)]
36. Van Toan, N.; Khoi, P.B. Fuzzy-based-admittance controller for safe natural human–robot interaction. *Adv. Robot.* **2019**, *33*, 815–823. [[CrossRef](#)]
37. Bououden, S.; Abdessemed, F.; Abderraouf, B. Control of a Bipedal Walking Robot Using a Fuzzy Precompensator. In Proceedings of the Agent and Multi-Agent Systems: Technologies and Applications, Uppsala, Sweden, 3–5 June 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 853–862.
38. Aldair, A.A.; Rashid, A.T.; Rashid, M.T.; Alsaedee, E.B. Adaptive Fuzzy Control Applied to Seven-Link Biped Robot Using Ant Colony Optimization Algorithm. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2019**, *43*, 797–811. [[CrossRef](#)]
39. Ankarali, A. Fuzzy Logic Velocity Control of a Biped Robot Locomotion and Simulation. *Int. J. Adv. Robot. Syst.* **2012**, *9*, 124. [[CrossRef](#)]
40. Li, Z.; Cheng, X.; Peng, X.B.; Abbeel, P.; Levine, S.; Berseth, G.; Sreenath, K. Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots. In Proceedings of the 2021 International Conference on Robotics and Automation, Xi'an, China, 30 May–5 June 2021.
41. Myeongseop, K.; Dong-Ki, H.; Jae-Han, P.; Jung-Su, K. Motion Planning of Robot Manipulators for a Smoother Path Using a Twin Delayed Deep Deterministic Policy Gradient with Hindsight Experience Replay. *Appl. Sci.* **2020**, *10*, 1–15.
42. Khoi, P.B.; Giang, N.T.; van Tan, H. Control and Simulation of a 6-DOF Biped Robot based on Twin Delayed Deep Deterministic Policy Gradient Algorithm. *Indian J. Sci. Technol.* **2021**, *14*, 2460–2471. [[CrossRef](#)]
43. Bipedal Robot Simulation. Available online: <https://youtu.be/8A3849PUCyM> (accessed on 21 September 2021).