ICIC Express Letters
Part B: Applications
Volume **14**, Number **5**, May **2023**

 ISSN 2185-2766
pp. 449–456

LQR COMPENSATED BY FUZZY FOR KICKING BALANCE CONTROL OF A HUMANOID ROBOT

Andi Dharmawan*, Jazi Eko Istiyanto, Agfianto Eko Putra and Muhammad Auzan

Electronics and Instrumentation Research Laboratory
Department of Computer Science and Electronics
Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada
Bulaksumur BLS 21, Jogjakarta 55281, Indonesia
{ jazi; agfi; muhammadauzan }@ugm.ac.id
*Corresponding author: andi_dharmawan@ugm.ac.id

Received November 2022; accepted January 2023

Abstract. *Robots have various shapes optimized to fulfill specific functions and tasks. One type of robot based on its form is a humanoid robot with a human-like figure. With a similar body shape, humanoid robots make it possible to carry out activities with humans. One of them is a humanoid robot playing soccer. For this reason, humanoid robots must be able to perform basic movements in soccer, such as kicking a ball. When kicking, the robot tends to fall during swinging leg movements due to extreme speed changes. The fixed gain value in control given to the robot is not optimal for overcoming this problem because of the setpoint changes that occur. For the needs of motion stability simulation, we use the inverted pendulum model as the basis for the mathematical model of the humanoid. This mathematical model is combined with Linear Quadratic Regulator (LQR), which is compensated by the Fuzzy method, and then simulated. This simulation aims to see the response of the control system on the robot. Providing LQR-based and Fuzzy controls has been shown to play a role in reducing excessive torque when kicking. This condition makes the humanoid robot succeed in kicking stably without falling.*
Keywords: Robot, Inverted pendulum, State space, Fuzzy, Bezier

1. **Introduction.** Humanoid robots are designed to perform activities like humans, one of which is walking on both feet. The ability of the robot to walk in a controlled manner using both legs is essential. Humanoid robots are designed not only to adjust and maintain control on flat and level surfaces, but also on uneven surfaces. One of the walking concepts of a humanoid robot that can be applied to walk on uneven surfaces is to model a humanoid robot like an inverted pendulum [1], with the Center of Mass (CoM) of the robot located above its axis of motion which is on the sole of the pedestal. Based on this model, the exact location of the foot placement can be calculated to compensate for the robot's tilt [2].

Robot balance is affected by the position of the robot's center of mass on the robot's polygon support [3]. The humanoid robot has a center of mass position in the center of the robot body. In a balanced state, the projection position of the robot's center of mass is always within the support polygon [4]. When the robot is standing with two supports, the robot's polygon support area is in the two-leg area, while when it is standing on one foot, the robot's polygon support area is smaller. Changes in the number of pedestals cause the projection position of the center of mass not to be in the polygon support. If it occurs for a long time, the robot will not be able to maintain its balance which causes the

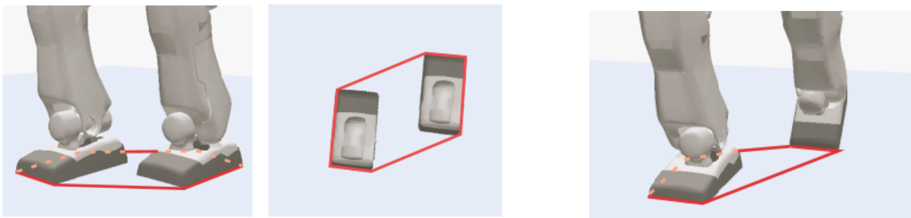
DOI: 10.24507/icicelb.14.05.449

robot to fall. So, it takes a kicking motion pattern with additional balance stabilization on the robot when making a kicking motion.

The addition of stabilization to the kicking motion aims to maintain the robot's balance while performing the movement. One of the concepts used to balance the attitude of a humanoid robot is the concept of a linear inverted pendulum. This concept is like the Center of Mass (CoM) of a robot centered at a point on the robot with a certain height and will shift when the robot's limbs move [5]. The Linear Quadratic Regulator (LQR) method is the optimal solution to minimize the value of the cost function in the quadratic form required by the system [6-8]. LQR can be used to make controls on humanoid robots that have many inputs and many outputs. In the research on the robot's balance, the robot managed to get good stability while walking using the LQR together with Fuzzy. In this research, Fuzzy has a role as compensator to the stabilization control system [9]. By using a fuzzy inference system [10,11], we use additional Fuzzy gain to compensate for the LQR control method so that the kicking stabilizer system on the humanoid robot is more optimal. This condition can increase the optimality of stability control and prevent the robot from falling when kicking.

The rest of the paper is structured as follows: Section 2 addresses the problem statement and preliminaries, Section 3 describes the control design, Section 4 describes results and discussion about the experimental results, and finally, Section 5 concludes the paper.

2. **Problem Statement and Preliminaries.** Humanoid robot is one type of robot that has a shape like a human. Humanoid robot is composed of several links and joints. The combination of links and joints forms parts of the robot such as the body, hands, and feet [12]. When standing upright, the humanoid robot is in a state of equilibrium. The balance limit of the humanoid robot is determined by the support polygon, as shown in Figure 1. Support polygon is an area formed by including all points of contact between the robot's feet and the surface to form an area that determines the balance of the robot. There are two kinds of support polygons on humanoid robots, namely double support polygons and single support polygons [13].



(a) (b)

FIGURE 1. Support polygon robot: (a) Two legs; (b) one leg

Bezier curve is one of the spline methods that can be used to create a movement pattern or trajectory [14]. In the Bezier pattern there are points consisting of 3 parts, namely initial points, control points, and goal points. The initial point is the starting point of the pattern, the goal point is the goal point, and the control point is the auxiliary point that will form the Bezier pattern from the starting point to the destination point [15].

The Bezier curve producing a trajectory is shown by Equation (1).

$$Q_0(t) = (1-t)P_0 + tP_1 \quad t \in [0,1] \tag{1}$$

In a linear Bezier curve, there are only two points, namely P_0 and P_1, so the pattern formed from phases 0-1 is linear. This pattern can be used for linear movements. In the quadratic Bezier curve equation, three points are needed, namely the curve's starting point and the endpoint (P_2). The quadratic Bezier curve equation is shown by Equation (2).

$$B_0(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2 \quad t \in [0,1] \tag{2}$$

The humanoid robot used in this study is a type-C Bioloid robot with a configuration of 14 servos, each consisting of 5 servos on the left and right legs, then two servos in the right hand and two servos in the left hand. Each servo has a limited angle of motion so as not to damage the existing components. Based on the datasheet, the AX-12 Dynamixel servo has a swivel angle of 330°.

When the robot performs a kicking motion, there will be a change in the tilt or orientation of the robot, which causes a difference in the position of the robot's center of mass against the robot's polygon support. If the robot cannot quickly overcome the change in the CoM and the position of the robot's center of mass is outside the polygon support for a long time, it will lose its balance and fall. When the robot performs a kicking motion, the part of the foot that treads on the floor is only one foot. This part is called the support foot. The support foot must be kept in a static state. The support foot is also a reference point for the system coordinates in kinematics [16].

The Fuzzy compensated LQR method is applied to determining the gain value that functions in optimizing system control. The most optimal gain value is used for the robot control's full state feedback gain. The control of the kicking motion is used to calculate the amount of torque needed to move the robot to a predetermined reference angle on the roll and pitch axis of rotation. Inverse kinematics moves the robot's legs towards a certain point. The kick pattern is created using a Bezier curve pattern.

3. **Control Design.** The design of the kick pattern is needed to determine the trajectory of each joint robot and determine the control of the robot [17]. The kicking pattern is made into seven steps. The first step is standing upright as a form of initiation position before kicking. The position of the robot's CoM against the robot's polygon support is at the midpoint of the double support foot. In the second step, the robot will shift the body's tilt by moving the two ankle servos so that the position of the robot's CoM moves to the y-axis of the polygon support. The third step will lift the kicking leg back as a square-off to make a kicking motion so that the robot gains kicking momentum, moving the ball from its starting point to its x-axis. The fourth step swings its legs on its x-axis to the ball's midpoint so that the ball can move faster. In the fifth step of the kicking, its leg will return to standing on one leg pose. In the sixth step of the kicking, the leg lowers until it touches the surface, so it returns to double support. The seventh step of the robot shifts the body's tilt to stand upright so that the position of the CoM is right in the middle of the polygon support.

3.1. **Kicking pattern.** The pattern of kicking in each step is made using the Bezier pattern. The parameters needed to kick the ball in a kicking pattern are the starting point, bending point, ending point, and the time required to move to each phase. In the step of shifting the body tilt, the robot shifts the body's position until the robot's center of mass is within reach of the robot's foot support.

When making a kicking motion, the kicking foot will follow the completed pattern, and the support foot will be controlled by full state feedback control. Full state feedback control results in the torque converted into angular value and angular velocity. Angle and angular velocity are calculated by integrating the angular acceleration obtained by dividing the torque by the moment of inertia. The relationship between torque, angular acceleration, angle, and angular velocity is found using the uniformly changing straight motion formula. The design of the kicking scenario is shown in Figure 2.

3.2. **Humanoid control model.** The humanoid robot can be modeled as an inverted pendulum representing complex joints into simple ones. The humanoid robot used in this study is a type-C Bioloid robot that can be changed with a robot height of 32 cm with a total mass of 1.38 kg. The humanoid robot has an actuator in the form of a servo motor

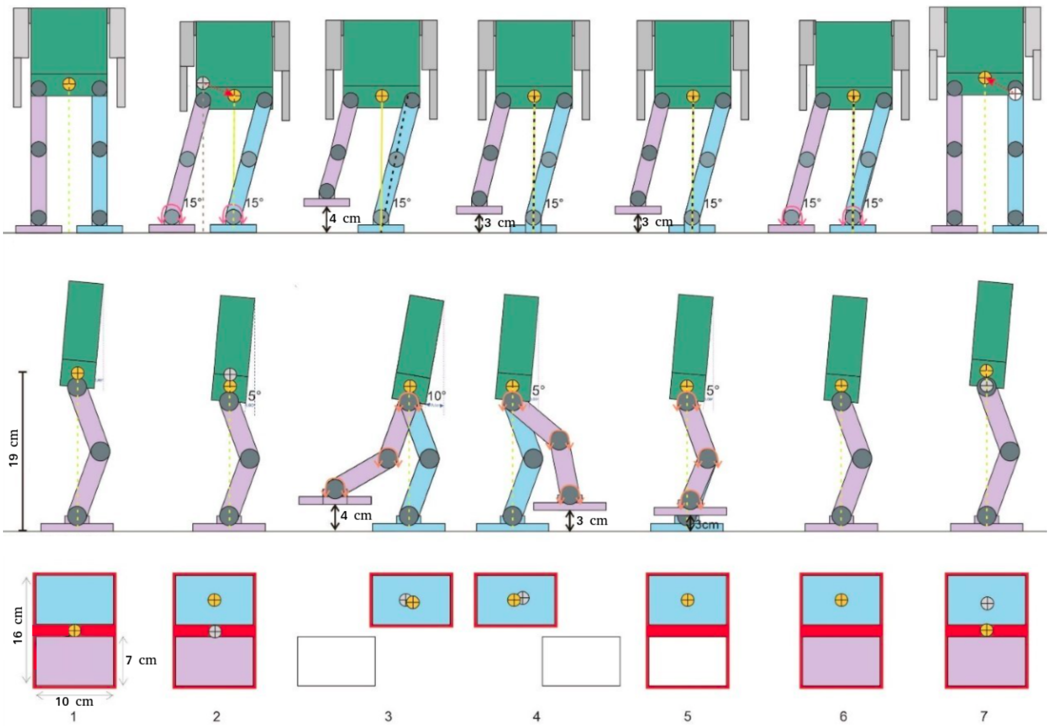


Figure 2. Kicking scenario

installed at each joint. A servo represents one axis of rotation of Degrees of Freedom (DoF). One torque input to the servo represents one controlled DoF.

The humanoid robot has moments of inertia on the x, y, and z axes. The moment of inertia of the z-axis is ignored. The robot is controlled by the x and y axes. The value of the moment of inertia in the inverted pendulum equation can be obtained by finding the value of the total moment of inertia of the robot. The parallel axis theorem calculates the total moment of inertia about a certain point.

The moment of inertia measures the robot's inertia to maintain its position in rotational motion. The moment of inertia depends on the object's mass, shape, and size. The humanoid robot in this study did not change in mass, shape, and size, so the change in the value of the moment of inertia was minimal and almost did not affect the balance of the humanoid robot. Therefore, the value of the moment of inertia of the humanoid robot in this model is constant.

The CoM position of each part of the humanoid robot is obtained through the forward kinematics calculation using the Denavit-Hartenberg (DH) method. The result of the calculation is a transformation matrix containing the value and direction of servo rotation. The position of the humanoid CoM for each x, y, and z axes can be calculated sequentially using Equation (3). m_j is the mass of each component, while x_j, y_j and z_j are the distance from the center of mass of each robot component to the foot fulcrum on the x, y, and z axes.

$$COMx = \frac{\sum m_j x_j}{\sum m_j}, \quad COMy = \frac{\sum m_j y_j}{\sum m_j}, \quad COMz = \frac{\sum m_j z_j}{\sum m_j} \tag{3}$$

The humanoid robot needs a control system to behave according to the desired movement. A humanoid robot with linear inverted pendulum modeling is a system with many inputs and outputs, called Multiple Input Multiple Outputs (MIMO). One method that can be used for this system is the Linear Quadratic Regulator (LQR) [18]. The system is represented in state space to control the pitch and roll angle of the robot's ankle.

The input for the tilt control of the robot body is the angle obtained from the IMU sensor readings on the pitch and roll rotation axes. The Fuzzy compensated LQR method produces the torque needed to drive the robot. The torque is converted into the angle and angular velocity of the ankle that supports the robot. Inverse kinematics will then calculate the angle of the entire robot leg servo according to the control and move.

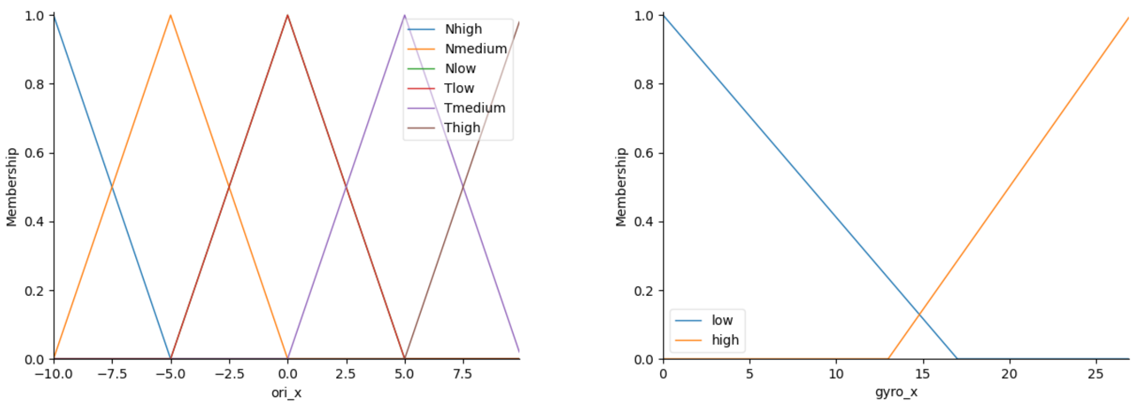
To simplify the model, we use the model in the equation of an inverted pendulum on the roll axis and the pitch axis, which is assumed to be the balance of the humanoid robot concerning these angles. These models are shown in Equation (4). Next, we convert these models into state space equations, as shown by Equation (5).

$$I_{yy}\ddot{\phi} - mgl\phi = \tau_r, \quad I_{xx}\ddot{\theta} - mgl\theta = \tau_p \tag{4}$$

$$\underbrace{\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ \dfrac{mgl}{I_{xx}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \dfrac{mgl}{I_{yy}} & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 & 0 \\ \dfrac{1}{I_{xx}} & 0 \\ 0 & 0 \\ 0 & \dfrac{1}{I_{yy}} \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\mathbf{u}} \tag{5}$$

where θ = pitch angle; $\dot{\theta}$ = pitch angular velocity; $\ddot{\theta}$ = pitch angular acceleration; ϕ = roll angle; $\dot{\phi}$ = roll angular velocity; $\ddot{\phi}$ = roll angular acceleration; I_{xx} = moment of inertia around x-axis; I_{yy} = moment of inertia around y-axis; g = gravity; m = robot total mass; l = distance from the center of mass to the sole of the foot; u_1 = pitch torque; u_2 = roll torque.

3.3. **Fuzzy control design.** To describe the set of error inputs, we used six linguistic variables. They are NK (Small Up), NS (Medium Up), NB (Big Up), TK (Small Down), TS (Medium Down), and TB (Big Down). We used two linguistic variables to describe the input set of delta errors. They are K (Small) and B (Large), as shown in Figure 3. The range of Fuzzy error set values is obtained based on the error angle that can occur in various variations of the tilt of the robot body. The response of the different robots to the variation of the robot's body tilt on the y-axis causes Fuzzy to be needed.



(a) (b)

FIGURE 3. (color online) (a) Fuzzy error set when the robot tilts around the y-axis; (b) Fuzzy delta error set when the robot tilts around the y-axis

Next, using the Mamdani method, we design a Fuzzy Inference System (FIS) that has fuzzy rules (rule-based) [19]. The Mamdani method is a method that has the basic form: If X = A, Then Y = B, so the fuzzy rules are often called the If-Then rules [20]. The rules formed based on the input are shown in Table 1.

After the fuzzy rules are formed, a defuzzification process will be carried out to obtain output in the form of an input value of **Q**, producing the suitable **K** reinforcement so that

TABLE 1. Fuzzy rules when the robot tilts around the y-axis

		Error					
Delta error		**NB**	**NS**	**NK**	**TK**	**TS**	**TB**
	DK	QNS	QNS	QNTK	QNTK	QTS	QTS
	DB	QNB	QNB	QNS	QTS	QTB	QTB

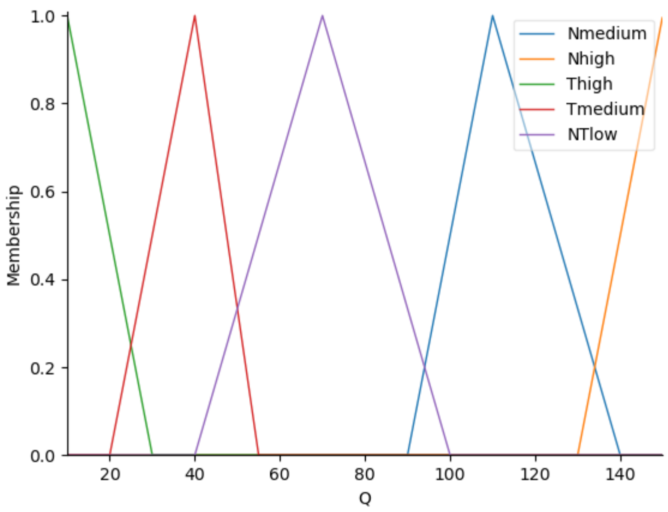


FIGURE 4. (color online) Fuzzy output set when the robot tilts around the y-axis

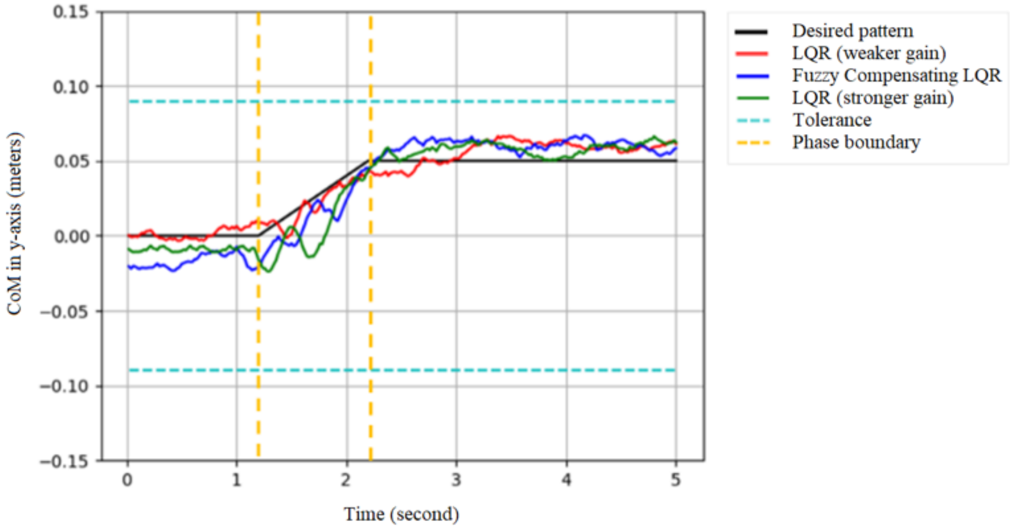
the robot can be stable when its body is tilted. The linguistic variables that describe the **Q** inputs are QNTS (small up and down Q), QNS (medium Q), and QB (large Q). The determination of the range of values for the **Q** variation is obtained by finding the correct value so that the robot remains stable in the tilt variation of the robot's body (10° to −10°). The Fuzzy set design for the **Q** output is shown in Figure 4.

4. **Results and Discussion.** The test results consist of several steps. The first step is testing the shifting phase. The second step focuses on testing the kicking phase. The third step focuses on testing the phase of returning to the initial position. And the last step, kicking the ball, is tested following the sequence of motion in Figure 2.

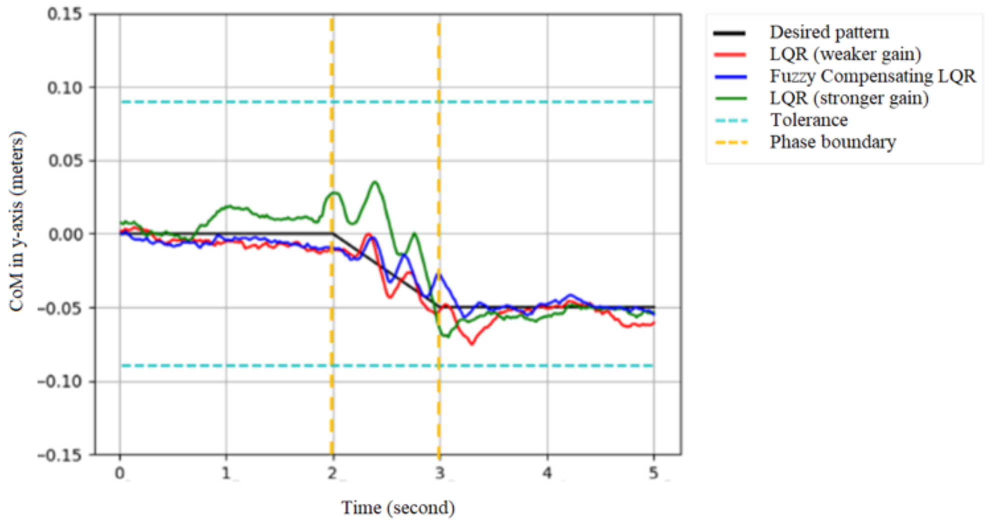
The shifting phase test is carried out to determine the robot's response to changes in the position of the robot body in the shifting phase. The initial position of the robot's center of mass projection to the polygon support is at the midpoint of the polygon support, which is 0 meters to the global x-axis and 0 meters to the global y-axis. The endpoints of the Bezier curve are 0.05 meters for the shifting in the positive y-axis and −0.05 meters for the shifting in the negative y-axis.

The balancing tests were carried out on ankle roll and hip roll robots that used the LQR method. Figure 5 shows the response to several tests. The red line indicates a weak response, while the green line indicates a too-strong response. A lousy reaction can cause the robot to be less responsive when there is a balance disturbance on the robot, while a too-strong response can cause the robot to be too responsive when there is a balance disturbance. Too strong a response also causes excessive oscillations. The blue line shows the ideal response generated by the robot. This blue line uses the Fuzzy compensated LQR method.

In the kicking phase, the robot's kicking motion was tested. The kicker's foot is given the kick parameters in the form of x, y, and z coordinates. The coordinates given are in



(a)



(b)

FIGURE 5. (color online) Movement transition response in the shifting phase: (a) First pattern; (b) second pattern

the form of a starting point, a curve point, and an endpoint of the Bezier curve parameter. In this kicking motion, there are 3 phases of movement: swinging one leg back, kicking forward, and returning to a standing position of 1 foot.

The Bezier curve produces a kicking motion pattern on the kicking leg. The coordinates of the resulting pattern are then used as input for position coordinates on the inverse kinematic of the kicking leg to produce the angle of motion in each servo robot.

5. **Conclusions.** The kicking motion of a humanoid robot using a Bezier curve pattern with the LQR method compensated by Fuzzy can perform a kicking motion without falling. The pattern of kicking motion using a Bezier curve made the robot perform a kicking motion with a right foot kick or a left foot kick. The LQR control system compensated by Fuzzy has successfully solved the balance problem of the humanoid robot when it kicks the ball. With the addition of stabilization, the humanoid robot can maintain balance during the transition stage from standing on two legs to standing on one leg. It can maintain balance when swinging its legs backward and kicking forwards with different kicking times. For further research, the development of walking and running robots while

kicking will be carried out. Besides that, the addition of computer vision will also be added. The computer vision will provide feedback on the ball's direction after kicking it.

Acknowledgment. This work is partially supported by Research Grants from Postgraduate Program of Computer Science, Department of Computer Science and Electronics, Universitas Gadjah Mada. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] A. Dharmawan, C. Habiba and M. Auzan, Walking stability control system on humanoid when turning based on LQR method, *Int. J. Sci. Technol. Res.*, vol.8, no.11, pp.2606-2611, 2019.

[2] N. S. Bhangal, Design and performance of LQR and LQR based fuzzy controller for double inverted pendulum system, *J. Image Graph.*, vol.1, no.3, pp.143-146, 2013.

[3] S. Kajita et al., Biped walking pattern generation by using preview control of zero-moment point, *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, pp.1620-1626, 2003.

[4] S. Behnke, Online trajectory generation for omnidirectional biped walking, *Proc. of 2006 IEEE International Conference on Robotics and Automation (ICRA2006)*, pp.1597-1603, 2006.

[5] S. Mason, N. Rotella, S. Schaal and L. Righetti, Balancing and walking using full dynamics LQR control with contact constraints, *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp.63-68, 2016.

[6] G. Nugroho and A. Dharmawan, Undesirable rolling minimization on the EDF missiles flight based on LQR methods, *2017 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)*, pp.85-90, 2017.

[7] G. Nugroho, A. Dharmawan, D. Lelono and A. M. Handayani, Waypoint tracking of a fixed-wing UAV using the L1 cross track error control, *ICIC Express Letters, Part B: Applications*, vol.13, no.2, pp.115-122, 2022.

[8] A. Ashari and A. Dharmawan, Altitude and flight speed control system on VTOL-plane UAVs using the LQR method, *ICIC Express Letters, Part B: Applications*, vol.13, no.3, pp.225-232, 2022.

[9] T. K. Priyambodo and A. Dharmawan, Auto vertical takeoff and landing on quadrotor using PID-fuzzy, *J. Eng. Appl. Sci.*, vol.12, no.3 SI, pp.6420-6425, 2017.

[10] M. Hentschke and E. P. de Freitas, Design and implementation of a control and navigation system for a small unmanned aerial vehicle, *IFAC-PapersOnLine*, vol.49, no.30, 2016.

[11] H. Suryoatmojo, D. R. Pratomo, Soedibyo, M. Ridwan, D. C. Riawan, E. Setijadi and R. Mardiyanto, Robust speed control of brushless DC motor based on adaptive neuro fuzzy inference system for electric motorcycle application, *International Journal of Innovative Computing, Information and Control*, vol.16, no.2, pp.415-428, 2020.

[12] S. Mason, L. Righetti and S. Schaal, Full dynamics LQR control of a humanoid robot: An experimental study on balancing and squatting, *2014 IEEE-RAS International Conference on Humanoid Robots*, pp.374-379, 2014.

[13] S. Kajita, H. Hirukawa, K. Harada and K. Yokoi, *Introduction to Humanoid Robotics*, Springer Berlin, Heidelberg, 2014.

[14] Y. Zhou and L. Chen, The path trajectory planning of swinging legs for humanoid robot, *IECON 2020 the 46th Annual Conference of the IEEE Industrial Electronics Society*, pp.384-389, 2020.

[15] N. Jouandeau and V. Hugel, Optimization of parametrised kicking motion for humanoid soccer player, *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp.241-246, 2014.

[16] S. Kucuk and Z. Bingul, *Robot Kinematics – Forward and Inverse Kinematics*, Pro Literatur Verlag, Germany, 2007.

[17] R. Cisneros, K. Yokoi and E. Yoshida, Impulsive pedipulation of a spherical object with 3D goal position by a humanoid robot: A 3D targeted kicking motion generator, *Int. J. Humanoid Robot.*, vol.13, no.2, 1650003, 2016.

[18] S. Sherif, T. Slavov and J. Kralev, Hardware-in-the-loop simulation on linear-quadratic controller for stabilization of a humanoid robot during walking, *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, pp.1-4, 2021.

[19] T. Docekal and S. Ozana, Design of fuzzy controller for simple inverted pendulum, *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp.1-6, 2020.

[20] T. K. Priyambodo, A. Dharmawan and A. E. Putra, PID self tuning control based on Mamdani fuzzy logic control for quadrotor stabilization, *AIP Conference Proceedings*, vol.1705, 2016.