https://doi.org/10.31436/iiumej.v27i1.3587

Stability and Control of Humanoid Robots during Walking and Sudden Stops on Uneven Terrain using Inverted Pendulum Modeling and Fuzzy-enhanced Linear Quadratic Regulator

ANDI DHARMAWAN, JAZI EKO ISTIYANTO*, BAKHTIAR ALLDINO ARDI SUMBODO, MUHAMMAD AUZAN, OSKAR NATAN

Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, 55128, Indonesia

**Corresponding author: jazi@ugm.ac.id*

(Received: 21 January 2025; Accepted: 24 September 2025; Published online: 12 January 2026)

ABSTRACT: Humanoid robots, designed to resemble human structure with a head, arms, and legs, stand out among various robot types due to their ability to interact with human-designed environments. Their kinematic structure, composed of links connected by joints, enables humanoid robots to perform tasks such as walking on uneven terrain, lifting objects, and opening doors. Walking is a critical function for humanoid robots, but it shifts their center of mass (CoM), which can lead to instability and falls. The inverted pendulum model is a widely used approach to humanoid robot walking, particularly on uneven surfaces, enabling CoM projection and balance adjustment. However, disturbances from floor unevenness, mechanical issues, and other factors can still cause the robot to fall. This underscores the need for effective control systems to stabilize movement and maintain balance. The Linear Quadratic Regulator (LQR) is a suitable control approach for humanoid robots, particularly in multiple-input, multiple-output systems. It stabilizes the robot while managing computational load, generating complex humanoid movements through linear approximations. To address uncertainties and adapt control gains, fuzzy control can be integrated, thereby enabling smoother transitions and improved disturbance handling. When walking on uneven terrain, a humanoid robot's body orientation may shift, causing the CoM to exceed tolerance limits and leading to falls. Thus, a control system that adjusts the walking pattern to maintain CoM stability is essential. In conclusion, humanoid robots, with their human-like structure, can perform complex tasks in human environments. Control systems, such as LQR and fuzzy control, are critical for maintaining balance and stability, even under challenging conditions, thereby enabling humanoid robots to keep their trajectory and prevent falls.

ABSTRAK: Robot seperti manusia (humanoid), direka menyerupai struktur manusia dengan kepala, lengan, dan kaki, menonjol dalam pelbagai jenis robot kerana keupayaannya berinteraksi dengan persekitaran yang direka oleh manusia. Struktur kinematiknya, terdiri daripada pautan yang dihubungkan oleh sendi, membolehkan robot humanoid melakukan tugas seperti berjalan di permukaan tidak rata, mengangkat objek, dan membuka pintu. Berjalan adalah fungsi penting robot humanoid, tetapi pergerakan mengubah pusat jisim (CoM), menyebabkan ketidakstabilan dan jatuh. Model bandul terbalik adalah pendekatan biasa digunakan bagi robot humanoid berjalan, terutama pada permukaan tidak rata, membolehkan unjuran CoM dan pelarasan keseimbangan. Namun, gangguan ketidaksamaan permukaan lantai, masalah mekanikal, dan faktor lain masih boleh menyebabkan robot jatuh. Ini menunjukkan keperluan pada sistem kawalan berkesan yang menstabilkan pergerakan dan mengekalkan keseimbangan. Regulator Kuadratik Linear (LQR) adalah pendekatan kawalan

https://doi.org/10.31436/iiumej.v27i1.3587

sesuai untuk robot humanoid, terutama dalam sistem masukan dan keluaran berganda. Ia menstabilkan robot sambil menguruskan beban pengiraan, menghasilkan pergerakan humanoid yang kompleks melalui pendekatan linear. Bagi menangani ketidakpastian dan menyesuai keuntungan kawalan, kawalan kabur boleh diintegrasikan, menjadikan peralihan lebih lancar dan pengendalian gangguan lebih baik. Apabila berjalan di permukaan tidak rata, orientasi badan robot humanoid mungkin berubah, menyebabkan CoM bergerak melebihi had toleransi, yang mengakibatkan jatuh. Oleh itu, sistem kawalan penyesuaian corak berjalan bagi mengekalkan kestabilan CoM adalah penting. Kesimpulannya, robot humanoid, dengan struktur seperti manusia, dapat melakukan tugas kompleks dalam persekitaran manusia. Sistem kawalan seperti LQR dan kawalan kabur adalah penting bagi memastikan keseimbangan dan kestabilan, walaupun dalam keadaan mencabar, membolehkan robot humanoid mengekalkan trajektori mereka dan mencegah jatuh.

KEYWORDS: *humanoid, robot, CoM, LQR, fuzzy logic, control*

1. INTRODUCTION

Robots come in various forms, optimized to perform specific functions and tasks [1]. Among the different types of robots, humanoid robots stand out due to their human-like structure [2][3]. Structurally, humanoid robots typically have a head, two arms, and two legs. Kinematically, these robots consist of links connected at joints that function like human joints, forming kinematic chains [4]. The resemblance to the human form enables humanoid robots to interact with equipment and environments designed for humans, such as walking on uneven terrain, lifting objects, opening doors, and running [5].

Walking is a fundamental capability for humanoid robots to interact with and adapt to human-designed environments effectively [6]. While a humanoid robot is stable when both feet are on the ground, movement causes its center of mass (CoM) to shift, which can lead to a loss of balance and potentially a fall [7]. One of the most common walking models for humanoid robots, particularly on uneven surfaces, is the inverted pendulum model. This model places the robot's CoM above its foot, which acts as the pivot, resulting in a projected motion of the CoM during walking [8]. Based on this inverted pendulum model, the foot placement can be calculated to balance the robot's body movement during walking [9].

However, despite accounting for balance during walking, humanoid robots can still fall due to various disturbances, such as uneven floors, mechanical misalignments, or other factors [10]. A sloped walking surface can cause the robot to fall as the CoM shifts towards the gravitational force acting on the inclined plane. This highlights the need for a control system to stabilize the humanoid robot's movement and maintain balance in line with the intended walking pattern.

Control systems play a crucial role in regulating a robot's motion, ensuring it follows the planned walking pattern. One modern control system suitable for humanoid robot walking, particularly in systems with multiple inputs and outputs, is the Linear Quadratic Regulator (LQR) [11][12]. LQR can address disturbances within the system and stabilize it while maintaining lightweight control with manageable computational requirements [13]. It uses linear approximations to generate complex humanoid movements.

To handle uncertainties and adapt control gains in response to disturbances, a fuzzy control system can be implemented. Fuzzy control addresses the limitations of rigid gain scheduling, providing smoother transitions and better adaptation to changing conditions. When walking on uneven terrain, the robot's body orientation may shift with each step, causing the CoM to move outside the tolerance limits, which can lead to a loss of balance and a fall. Therefore, a control

system that adjusts the walking pattern to maintain the CoM within these tolerance limits is essential for ensuring the robot remains upright on uneven surfaces.

In summary, humanoid robots, with their human-like structure, can perform activities like those of humans, such as climbing stairs, lifting objects, and opening doors. As these robots are further developed, they will be able to better adapt to human-designed environments [14]. The most common concept in humanoid robot modeling is the Linear Inverted Pendulum Model (LIPM), known for its simplicity and effectiveness in generating stable movement [15]. However, sudden stops or uneven surfaces can cause robots to lose balance, underscoring the need for control systems that can maintain stability during unexpected conditions. Using LQR and fuzzy control, humanoid robots can maintain balance, walk, and make sudden stops even in challenging environments, ensuring that their CoM remains stable and within the desired trajectory. This paper is organized as follows: Section 2 outlines the problem statement; Section 3 describes the system design; Section 4 presents experimental results and performance analysis; and Section 5 concludes.

2. PROBLEM STATEMENT

Humanoid robots face significant challenges in maintaining balance, especially during sudden stops or when walking on inclined surfaces. When a robot is abruptly stopped, it tends to lose balance due to unpredictable foot placement and reactive forces generated by the sudden deceleration. If the robot cannot quickly counteract these forces, it will fall. This issue is further complicated when the robot walks on inclined surfaces, where the projection of its center of mass (CoM) shifts with the incline, increasing the risk that the CoM moves outside the robot's support polygon and leads to a fall.

To address these challenges, the humanoid robot’s balance is controlled by calculating the required joint torques, specifically along the roll and pitch axes of the servos. A full state feedback control using the Linear Quadratic Regulator (LQR) method is applied to determine the necessary torque [16]. The Inertial Measurement Unit (IMU), which includes accelerometers and gyroscopes, is used to measure the robot's orientation and inclination angles (pitch and roll) during both walking and sudden stops.

The walking pattern of the humanoid robot is generated using forward kinematics, based on the Denavit-Hartenberg (DH) method, to compute the CoM trajectory. Inverse kinematics is then employed to determine the joint angles required to achieve the desired foot positions, thereby ensuring that the robot follows the predefined walking pattern.

To enhance stability, the system is designed to maintain the robot's CoM projection within the support polygon by employing a heel strategy that adjusts the ankle angles to preserve balance. The LQR controller optimizes the system by minimizing the error between the reference and actual states, producing torque commands that adjust the robot's motion to align with the reference walking pattern. This approach ensures that the humanoid robot can maintain balance during sudden stops and when navigating inclined surfaces, thereby preventing falls by continuously adjusting the CoM to keep it within the robot's support polygon.

3. SYSTEM DESIGN

3.1. System Model

A humanoid robot is a type of robot that resembles the human form. Structurally, a humanoid robot consists of one head, one body, two arms, and two legs. Due to their human-

https://doi.org/10.31436/iiumej.v27i1.3587

like structure, humanoid robots are continually being developed to perform and assist with human tasks and to operate human tools in human environments.

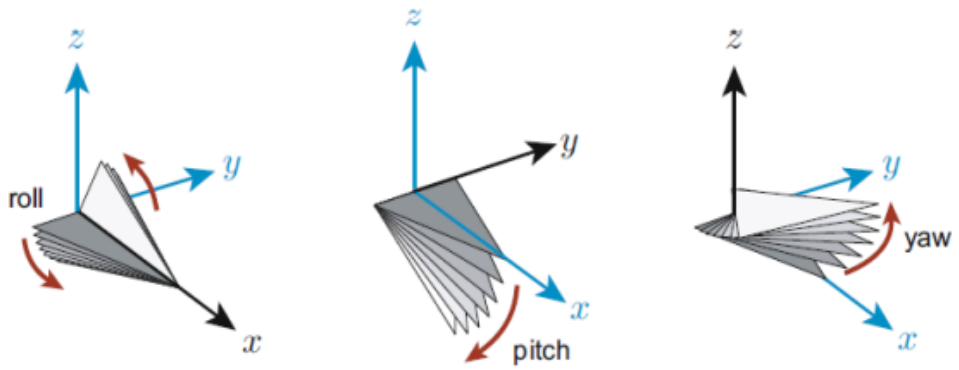


Figure 1. Roll, pitch, and yaw rotations [17]

Humanoid robots are powered by actuators, typically motors, with the number of motors corresponding to the desired degrees of freedom. The motors drive the robot's joints, which rotate about the yaw, pitch, or roll axes. Yaw rotation occurs along the z-axis, pitch rotation along the y-axis, and roll rotation along the x-axis. Figure 1 illustrates roll, pitch, and yaw rotations. Each humanoid robot has actuators that function like human joints. Every joint in the robot is referred to as a "joint," and each joint is connected to a rigid mechanical component called a "link." The combination of joints and links forms various parts of the humanoid robot's body, such as the arms, thighs, and knees.

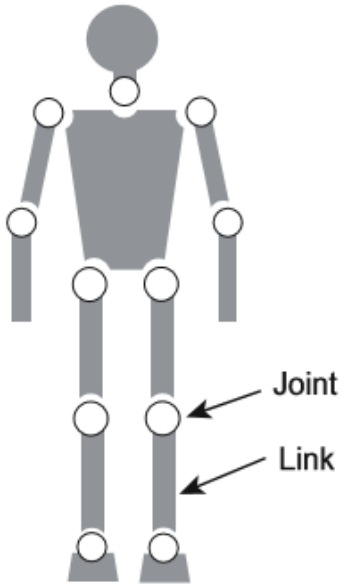


Figure 2. Link and joint of a humanoid robot

The humanoid robot used in this research is equipped with 12 servos, arranged with 10 in the legs and 2 in the arms. Figure 3 shows the robot's global coordinates from a rear view. The robot's x-axis points forward, the global y-axis points to the robot's left, and the z-axis points upwards.

The humanoid robot used in this research is a modified Bioloid Type-C robot, with a height of 32.3 cm and a total mass of 1.38 kg. Each servo on the robot represents a single rotational degree of freedom (DoF). The humanoid robot can be modeled as an inverted pendulum,

https://doi.org/10.31436/iiumej.v27i1.3587

simplifying the complex joints into a more straightforward representation. A single torque input to a servo represents one controlled degree of freedom. The configuration of the servo identification numbers for the humanoid robot in this research is shown in Figure 4.

A humanoid robot maintains its balance by counteracting the forces generated by changes in posture during leg swinging. The torque generated by the kicking leg alters the angles about the pitch and roll axes of the inverted pendulum model. The error caused by the change in body tilt must be zero for the robot to remain balanced; therefore, a torque opposite to the kicking torque is required on the robot's supporting leg.

The general equation for an inverted pendulum is shown in Eq. **Error! Reference source not found.**. An inverted pendulum is a pendulum where the mass is positioned above its pivot point [17]. The position of a human when standing is modeled as an inverted pendulum. The same concept is applied to humanoid robots, as shown in Figure 5. There are two assumptions underlying the modeling of a humanoid robot as an inverted pendulum. First, the mass of the entire body is concentrated at a single point called the Center of Mass (CoM). Second, this point mass is supported by a leg that is assumed to be massless, with the axis of rotation located at the ankle joint [18]. Figure 5 represents the model of an inverted pendulum where τ represents the torque of the pendulum. θ is the angle of pendulum deflection, r is the length of the pendulum, and M is the mass of the pendulum.

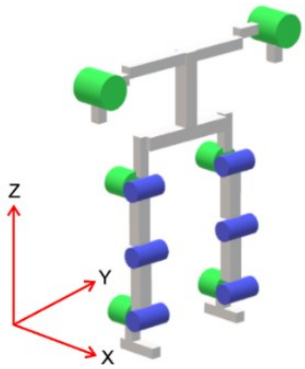


Figure 3. Global Coordinate Axes (x, y, z) of the Robot

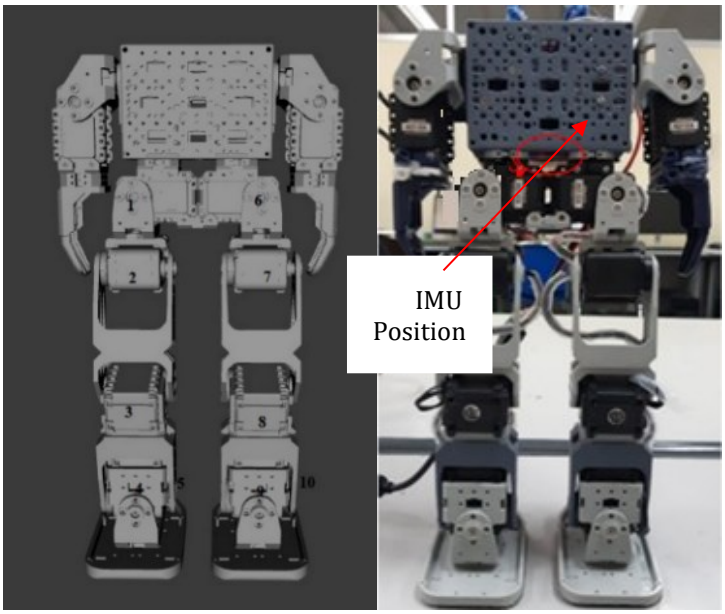


Figure 4. Humanoid Robot Configuration

https://doi.org/10.31436/iiumej.v27i1.3587

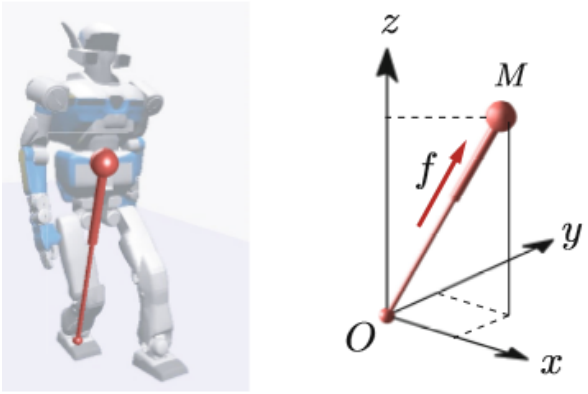


Figure 5. Inverted Pendulum Model

$$Mr\ddot{\theta} - Mgr \sin\theta = \tau \quad (1)$$

The deflection angle used in this study is 0.5°. $\sin\theta$ can be approximated as θ because the deflection angle affecting the humanoid robot's balance is relatively small. Thus, equation **Error! Reference source not found.** can be rewritten as Eq. **Error! Reference source not found.**.

$$I\ddot{\theta} - mgl\theta = \tau_{kick} \quad (2)$$

The inverted pendulum equation involves I, which is determined through mathematical calculations and defined as I_{xx} and I_{yy}. I_{xx} is the moment of inertia on the x-axis, while I_{yy} is the moment of inertia on the y-axis. Equations **Error! Reference source not found.** and **Error! Reference source not found.** show the equations for determining the moments of inertia I_{xx} and I_{yy}.

$$I_{xx} = \sum_{j=1}^{n}\left(I_{G_{xx_j}} + m_j\left(y_j^2 + z_j^2\right)\right) \quad (3)$$

$$I_{yy} = \sum_{j=1}^{n}\left(I_{G_{yy_j}} + m_j\left(x_j^2 + z_j^2\right)\right) \quad (4)$$

Eq. **Error! Reference source not found.** shows that I_{xx} is the moment of inertia on the x-axis. $I_{G_{xx_j}}$ is the moment of inertia of each robot component, represented as a solid block on the x-axis, m_j is the mass of each component, and y_j and z_j are the coordinates of the center of mass of each component relative to the robot's rotational axis. Meanwhile, in Eq. **Error! Reference source not found.**, I_{yy} is the moment of inertia on the y-axis. $I_{G_{yy_j}}$ represents the moment of inertia of each robot component, represented as a solid block on the x-axis, and m_j is the mass of each component. x_j and z_j are the coordinates of the center of mass of each component relative to the robot's rotational axis.

Eq. **Error! Reference source not found.** shows the moment of inertia of a solid block.

$$I_{G_{xx}} = \sum_{j=1}^{n}\left(I_{G_{xx_j}} + m_j\left(y_j^2 + z_j^2\right)\right) \quad (5)$$

$$I_{G_{yy}} = \frac{1}{12}m(b^2 + c^2) \quad (6)$$

https://doi.org/10.31436/iiumej.v27i1.3587

Based on Eqs. **Error! Reference source not found.** and **Error! Reference source not found.**, $I_{G_{XX_j}}$. and $I_{G_{yy_j}}$ are the inertias on the x and y axes of the body, respectively, for each component j; x_j and y_j are the distances of the center of mass of each component j from the axis of rotation on the x and y axes; and m_j is the mass of each component j [19].

The moment of inertia is a measure of a robot's resistance to changes in its rotational motion. The moment of inertia depends on the object's mass, shape, and size. The humanoid robot does not change mass, shape, or size; therefore, the change in its moment of inertia is very small and has almost no effect on its balance. Therefore, the moment of inertia value of the humanoid robot in this model is assumed to be constant.

The moment of inertia measures a robot's resistance to changes in its rotational motion and depends on the object's mass, shape, and size. Since the humanoid robot maintains constant mass, shape, and size, variations in its moment of inertia are minimal and have negligible effects on its balance. Therefore, in this model, the humanoid robot's moment of inertia is assumed to be constant.

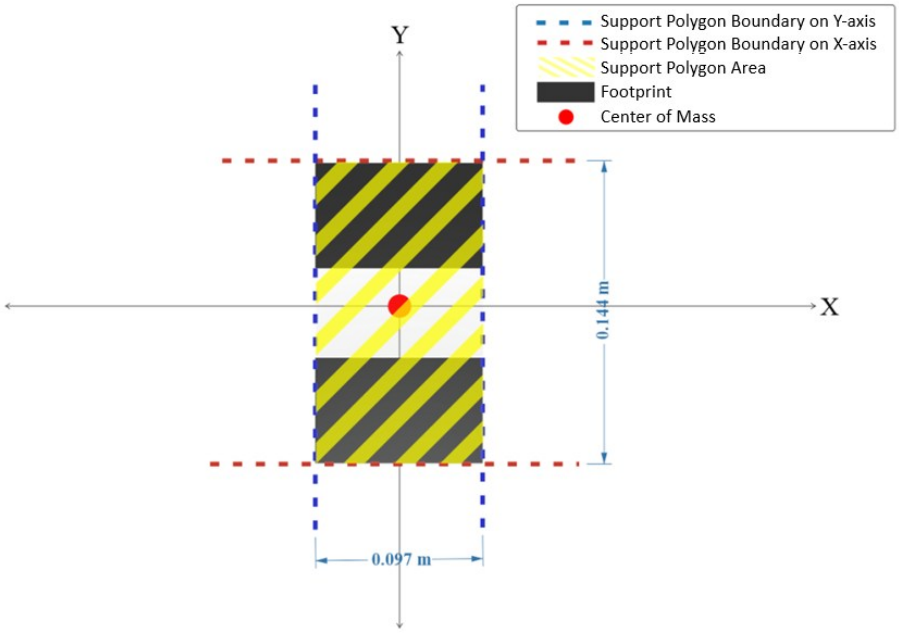
The center of mass (CoM) of each part of the humanoid robot is determined through forward kinematics calculations using the Denavit-Hartenberg (DH) method [20]. These calculations yield transformation matrices that encode the servo positions and rotation directions. The positions of the humanoid's CoM along the x, y, and z axes are calculated sequentially using the equations **Error! Reference source not found.**, **Error! Reference source not found.**, and **Error! Reference source not found.**.

$$x_{CoM} = \frac{\sum m_j x_j}{\sum m_j} \tag{7}$$

$$y_{CoM} = \frac{\sum m_j y_j}{\sum m_j} \tag{8}$$

$$z_{CoM} = \frac{\sum m_j z_j}{\sum m_j} \tag{9}$$

where m_j is the mass of each component, and x_j, y_j, and z_j are the distances of the center of mass of each robot component from the leg's supporting point along the respective axes [21].



https://doi.org/10.31436/iiumej.v27i1.3587

Figure 6. Calculation of the Robot's Support Polygon Area

The balance controller in this study employs a stabilizer control system to manage the CoM, ensuring it aligns with the walking pattern. The robot follows a static walking model, in which the center of mass never extends beyond the support polygon. For the robot used, tolerance limits or the maximum reach of the foot-support area (support polygon) are determined by calculating the support-polygon area for each foot, as illustrated in Figure 6.

Based on these measurements, the upper and lower tolerance limits for CoM movement along the robot's x and y axes are established. Specifically, the CoM projection relative to the support polygon has tolerance values of ±0.0485 meters along the global x-axis and ±0.072 meters along the global y-axis when the robot stands on two feet.

The humanoid robot's walking pattern is derived from a three-dimensional linear inverted pendulum equation, generating reference coordinates for the CoM movement along the x, y, and z axes. The CoM position is obtained from forward kinematics calculations using the DH convention. The walking pattern is divided into four phases:

- Shifting CoM: Preparing to lift a foot by shifting the CoM.
- Foot Lift: Lifting the foot off the ground.
- Stepping and Placement: Moving the foot forward and placing it down.
- CoM Adjustment: Adjusting the CoM to the final position.

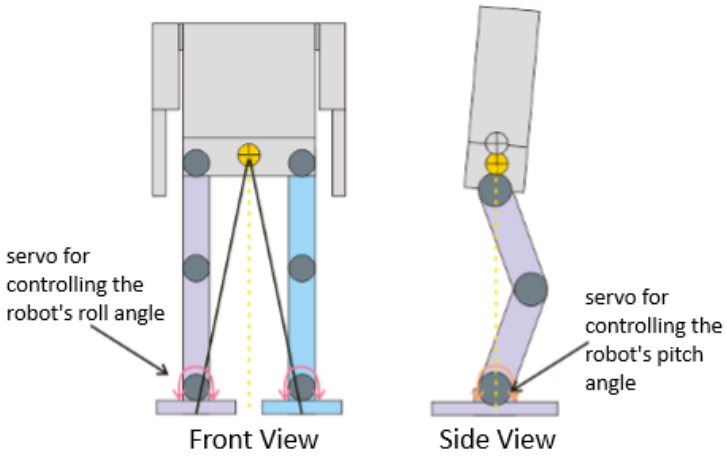


Figure 7. The location of the servos to be controlled

To ensure the humanoid robot's movements align with the reference walking pattern, a combination of an IMU and forward kinematics is used to determine deviations between the robot's current position and the reference pattern. LQR control is employed to determine the optimal gain values for system optimization.

The system uses the BNO055 IMU, which integrates an accelerometer, gyroscope, and magnetometer with an internal sensor fusion engine (Bosch BSX3.0) to provide absolute orientation in Euler angles. This module ensures high stability for short-term control applications, with heading errors typically below ±3° and pitch/roll below ±1° after calibration. Since the scope of this study focuses on short-duration stabilization, specifically the transition from perturbed to steady-state posture, the impact of long-term IMU drift is negligible. Therefore, additional sensor fusion, such as vision-based tracking, is unnecessary for the intended control objective.

https://doi.org/10.31436/iiumej.v27i1.3587

As depicted in Figure 7, the servos controlled are located at the robot's heels and facilitate movement in roll and pitch. The control system inputs are the deviations in roll and pitch angles, as well as their angular velocities at the ankles, compared to the reference values. The system outputs torque values that represent the energy required to adjust the robot toward the reference walking pattern. This is achieved by comparing the measured robot position with the reference pattern and calculating the necessary response speed to correct the position. The roll and pitch servos at the ankles generate the appropriate torque to move the robot and shift its CoM projection, ensuring alignment with the reference walking pattern.

3.2. Walking Pattern Design

The walking pattern design is required to determine the trajectory of the projected center of mass (CoM) during walking. Adjusting this point will affect the robot's joint motion and control. The Zero Moment Point (ZMP) principle is the most widely used method for determining the CoM projection trajectory, and, based on this principle, the Preview Control walking pattern was developed. This concept ensures that the CoM always follows the ZMP pattern, thereby simplifying the concept of walking balance.

It is important to note that the preview control in this study is computed offline during the walking pattern generation stage. The resulting ZMP trajectory serves as a reference for the LQR–fuzzy controller. Because the computation is not performed online at each control iteration, it imposes no real-time computational delay, thereby allowing the control loop to remain responsive and efficient.

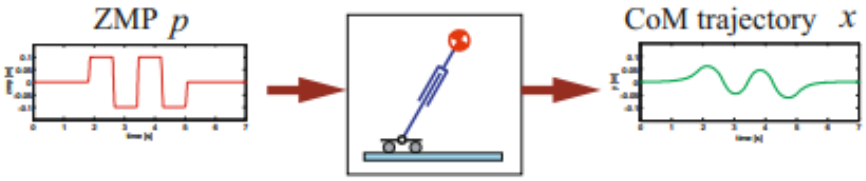


Figure 8. Cart-table model inputs ZMP and outputs CoM motion [22]

Figure 8 illustrates the Preview Control walking pattern, where the ZMP equation can be formulated in state-space form, as shown in Eq. **Error! Reference source not found.**.

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$
$$p = \begin{bmatrix} 1 & 0 & -\left(\frac{z_c}{g}\right) \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \tag{10}$$

Next, the state space Eq. **Error! Reference source not found.** is converted into a discrete-time system equation using a sample time Δt, resulting in Eq. **Error! Reference source not found.**.

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{b}u_k \\ p_k &= \mathbf{c}\mathbf{x}_k \end{aligned} \tag{11}$$

where

$$\begin{aligned} \mathbf{x}_k &\equiv [x(k\Delta t) \quad \dot{x}(k\Delta t) \quad \ddot{x}(k\Delta t)]^T, \\ u_k &\equiv u(k\Delta t), \\ p_k &\equiv p(k\Delta t), \end{aligned}$$

https://doi.org/10.31436/iiumej.v27i1.3587

$$\mathbf{A} \equiv \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{b} \equiv \begin{bmatrix} \frac{\Delta t^3}{6} \\ \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}, \boldsymbol{c} \equiv \begin{bmatrix} 1 & 0 & -\left(\frac{z_c}{g}\right) \end{bmatrix}.$$

The Preview Controller calculation produces a CoM pattern representing the target ZMP. The graph showing the CoM pattern representing the ZMP can be seen in Figure 9. Once the CoM pattern is obtained, the footstep pattern will use a quadratic Bezier curve to generate the stepping pattern for the left and right feet during movement. The footstep pattern will be combined in accordance with the CoM pattern. This pattern is then combined to form a walking pattern, as shown in Figure 10.

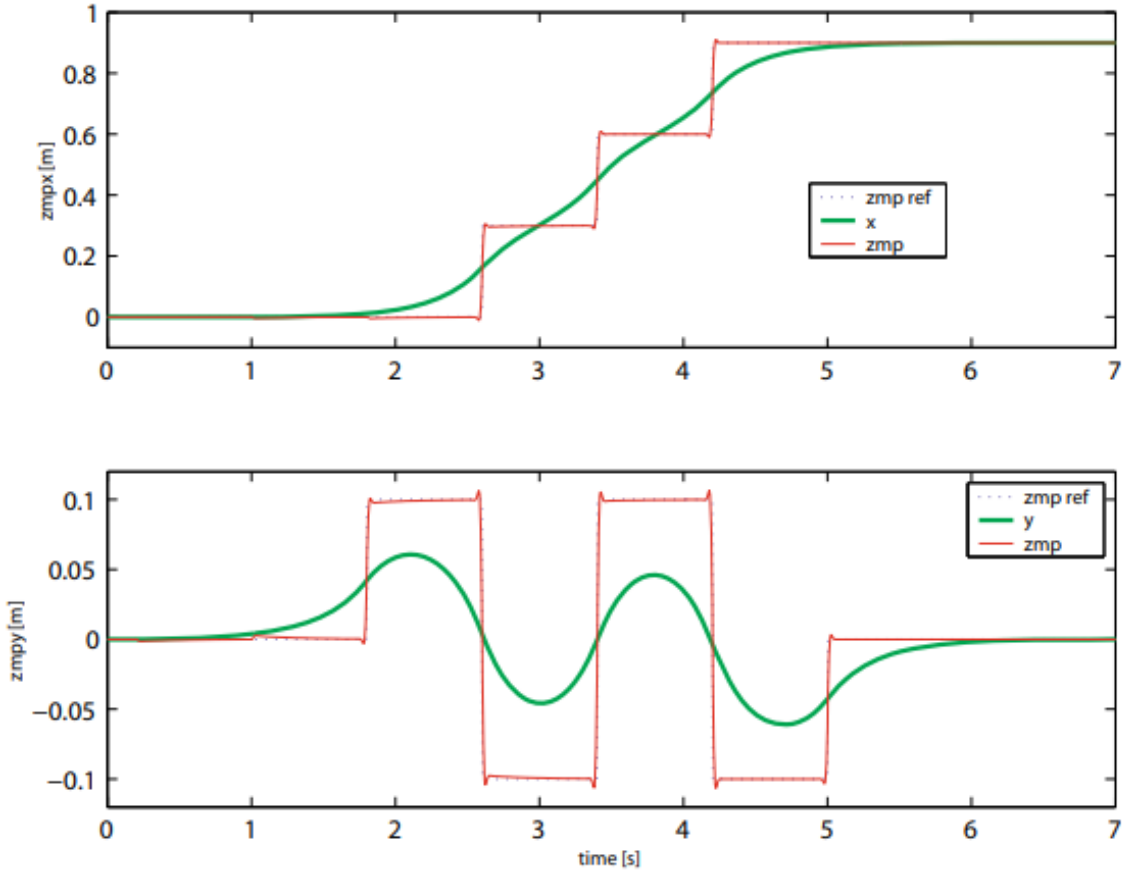


Figure 9. CoM Pattern Generated by the Preview Controller

https://doi.org/10.31436/iiumej.v27i1.3587

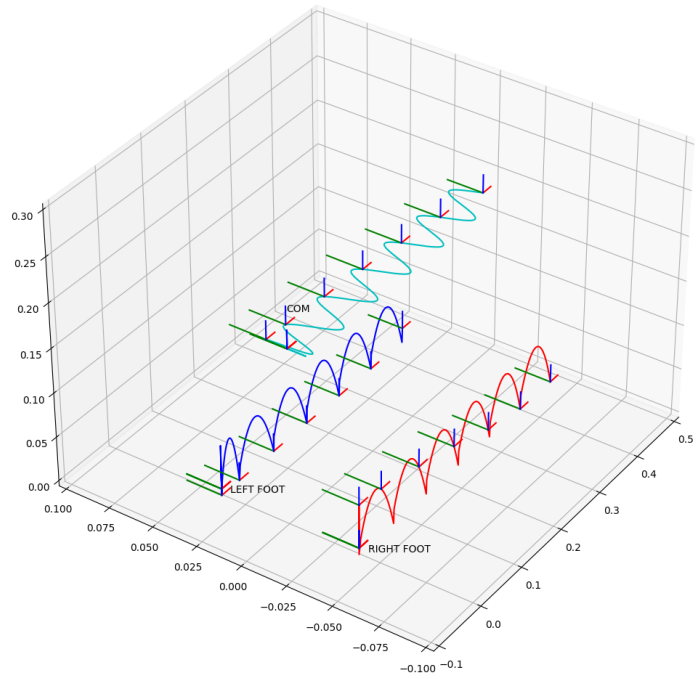


Figure 10. Visualization of the robot's walking pattern using a preview controller

3.3. Control Design

A control system is required for the humanoid robot to walk according to the desired pattern. The control system for the humanoid robot can be modeled as a two-dimensional inverted pendulum to control the ankle's roll and pitch. This inverted pendulum model represents the dynamics of the humanoid robot.

In this study, the control system will use four states as the basic reference for the control parameters. These four states consist of:

- Roll angle (ϕ)
- Pitch angle (θ)
- Roll angular velocity ($\dot{\phi}$)
- Pitch angular velocity ($\dot{\theta}$)

The general dynamics of the inverted pendulum are reduced to a simplified state-space model, as presented in Eq. **Error! Reference source not found.**, which serves as the foundation for the control design.

$$I_{xx}\ddot{\phi} - mgl \sin \phi = \tau_{\phi} \tag{12}$$

Before converting to the state space equation, the ϕ angle and its derivatives are symbolized by y_1, which represents the system outputs, and τ_{ϕ} is symbolized by u_1, representing an action in the form of torque along the roll axis. Additionally, the variable I_{xx} is rearranged to obtain a general model equation, shown by Eq. **Error! Reference source not found.**.

$$\ddot{y}_1 - \frac{mgl}{I_{xx}} y_1 = \frac{1}{I_{xx}} u_1 \tag{13}$$

Eq. **Error! Reference source not found.** is then classified based on the variables y and u along with their derivatives as follows

https://doi.org/10.31436/iiumej.v27i1.3587

$$a_{11} = 0; a_{21} = -\frac{mgl}{I_{xx}}$$
$$b_{01} = 0; b_{11} = 0; b_{21} = \frac{1}{I_{xx}}$$

Auxiliary variables β, x, and $\dot{x}$ are needed to derive the state space equation for the inverted pendulum along the pitch axis, as shown by Equation **Error! Reference source not found.**:

$$\begin{gathered}
\beta_{01} = b_{01} = 0 \\
\beta_{11} = b_{11} - a_{11}\beta_{01} = 0 \\
\beta_{21} = b_{21} - a_{11}\beta_{11} - a_{21}\beta_{01} = \frac{1}{I_{xx}} \\
x_{11} = y_1 - \beta_{01}u_1 = y_1 \\
x_{21} = \dot{y_1} - \beta_{01}\dot{u_1} - \beta_{11}u_1 = \dot{y_1} \\
\dot{x_{11}} = x_{21} + \beta_{11}u_1 = x_{21} \\
\dot{x_{21}} = -a_{21}x_1 - a_{11}x_{21} + \beta_{21}u_1 = \frac{mgl}{I_{xx}}x_1 + \frac{1}{I_{xx}}u_1 \\
\begin{bmatrix} \dot{x_{11}} \\ \dot{x_{21}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{mgl}{I_{xx}} & 0 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I_{xx}} \end{bmatrix} u_1 \\
y_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} + 0\, u_1
\end{gathered} \tag{14}$$

The same process is applied to derive the state space equation for the pitch axis:

$$I_{yy}\ddot{\phi} - mgl\sin\theta = \tau$$
$$I_{yy}\ddot{\phi} - mgl\theta = \tau$$
$$\ddot{y_2} - \frac{mgl}{I_{yy}}y_2 = \frac{1}{I_{yy}}u_2$$

Thus:

$$a_{12} = 0\ ; a_{22} = -\frac{mgl}{I_{yy}}$$
$$b_{02} = 0\ ; b_{12} = 0\ ; b_{22} = \frac{1}{I_{yy}}$$
$$a_{12} = 0\ ; a_{22} = -\frac{mgl}{I_{yy}}$$

With auxiliary variables β, x, and $\dot{x}$:

$$\begin{gathered}
\beta_{02} = b_{02} = 0 \\
\beta_{12} = b_{12} - a_{12}\beta_{02} = 0 \\
\beta_{22} = b_{22} - a_{12}\beta_{12} - a_{22}\beta_{02} = \frac{1}{I_{yy}} \\
x_{12} = y_2 - \beta_{02}u_2 = y_2 \\
x_{22} = \dot{y_2} - \beta_{02}\dot{u_2} - \beta_{12}u_2 = \dot{y_2} \\
\dot{x_{12}} = x_{22} + \beta_{12}u_2 = x_{22} \\
\dot{x_{22}} = -a_{22}x_{12} - a_{12}x_{22} + \beta_{22}u_2 = \frac{mgl}{I_{yy}}x_1 + \frac{1}{I_{yy}}u_2
\end{gathered}$$

https://doi.org/10.31436/iiumej.v27i1.3587

$$\begin{bmatrix} \dot{x}_{12} \\ \dot{x}_{22} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{mgl}{I_{yy}} & 0 \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I_{yy}} \end{bmatrix} u_2$$
$$y_2 = [1 \quad 0] \begin{bmatrix} x_{12} \\ x_{22} \end{bmatrix} + 0\, u_2 \tag{15}$$

From the inverted pendulum modeling and the four states mentioned above, the state space equations **Error! Reference source not found.** and **Error! Reference source not found.** are then combined to obtain the overall state space equations, shown in equations **Error! Reference source not found.** and (17).

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mgl}{I_{xx}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{mgl}{I_{yy}} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{I_{xx}} & 0 \\ 0 & 0 \\ 0 & \frac{1}{I_{yy}} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{16}$$
$$\dot{\mathbf{x}} \qquad \mathbf{A} \qquad \mathbf{x} \qquad \mathbf{B} \quad \mathbf{u}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{17}$$
$$\mathbf{y} \qquad \mathbf{C} \qquad \mathbf{x} \qquad \mathbf{D} \quad \mathbf{u}$$

Explanation:

- 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
- u_1 = Pitch action
- u_2 = Roll action
- I_{xx} = Moment of inertia about the x-axis
- I_{yy} = Moment of inertia about the y-axis
- g = Gravitational force of the Earth
- y_1 = Pitch angle output
- y_2 = Roll angle output
- m = Total mass of the robot
- l = Distance from the center of mass to the sole of the foot

https://doi.org/10.31436/iiumej.v27i1.3587

The control system employed is an LQR controller in a closed-loop configuration. The primary goal of LQR control is to find the gain **K** matrix for full-state feedback. LQR is a control method that can be applied to a robot model represented in state-space form. The LQR control in the humanoid robot system with state space equations requires the K gain to stabilize the system. The block diagram of the control system is shown in Figure 11. This gain **K** is used to determine the system action value, **u**. The LQR method is a regulator control system, designed to drive the system towards a state with a reference value of zero. Adjusting the reference to a specific physical state can be achieved using a reference state ($\mathbf{x}_{ref}$). $\mathbf{x}_{ref}$ consists of θ_{ref}, $\dot{\theta}_{ref}$, ϕ_{ref}, and $\dot{\phi}_{ref}$. This allows the system to either maintain a zero condition or sustain a particular state based on the new reference ($\mathbf{x}_{ref}$). For a humanoid robot, control is necessary to ensure that it can walk according to a specified reference trajectory. If the system output does not match the reference state, the system will repeat the process until the desired state is achieved.

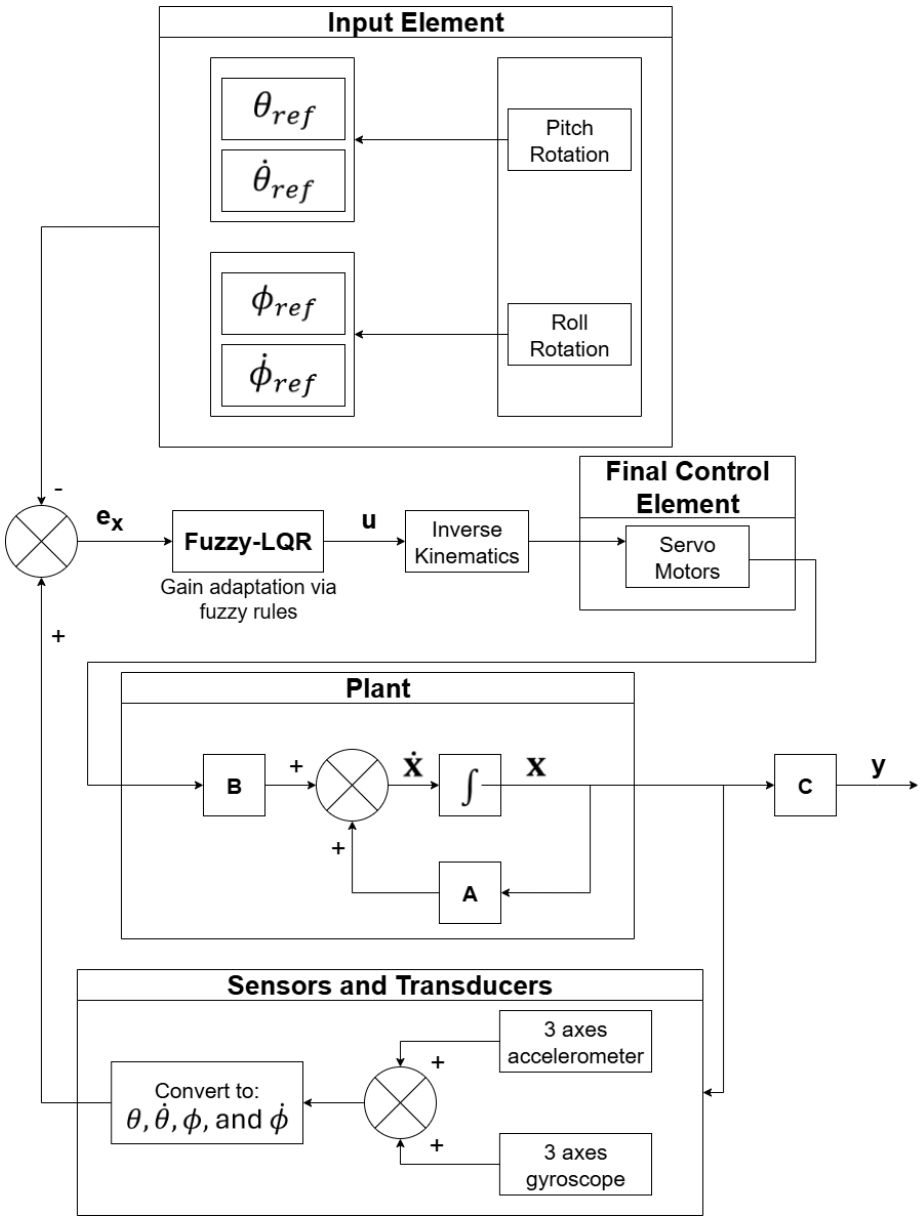


Figure 11. Block diagram of the proposed Fuzzy-LQR control system with gain adaptation based on fuzzy rules for humanoid stability on uneven terrain.

https://doi.org/10.31436/iiumej.v27i1.3587

Figure 11 shows the block diagram of control using gain **K**. The angle sensor within the smart servo measures the angular motion of each servo. These angles are used in forward kinematics to determine the position of the robot's center of mass along the robot's x, y, and z axes. The use of sensor fusion (accelerometer and gyroscope) can provide additional reference points for determining the robot's center-of-mass motion. The predefined reference state will serve as a basis for comparison with the robot's new state. The difference between the reference state and the new robot state is multiplied by the gain **K** value, which is then used as a new action in the system. The system's output is the position and velocity of the center of mass, which are converted into servo angles via inverse kinematics, thereby driving the robot's motion.

Forward kinematics helps recognize terrain differences during walking by mapping the waist position relative to the foot surface and comparing it to x and y axis errors detected by the IMU sensors. When the waist position matches the walking pattern, but x and y errors exceed the reference movement pattern, the gain **K** will be adjusted to account for terrain differences.

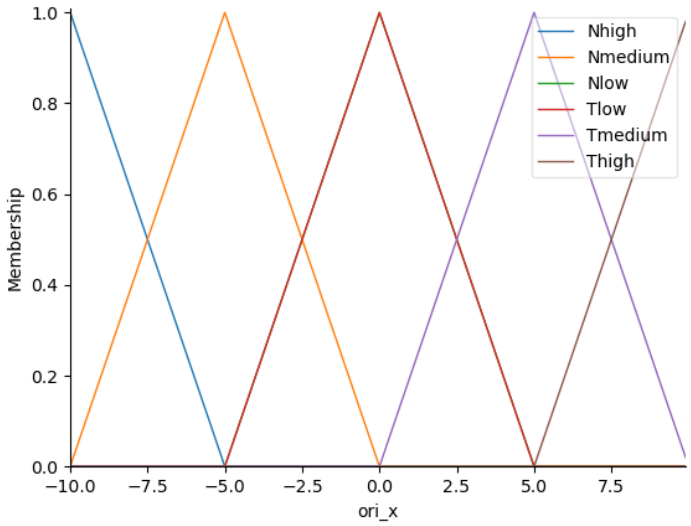


Figure 12. Fuzzy error set on an uneven and sloped surface (relative to the front body of the robot).

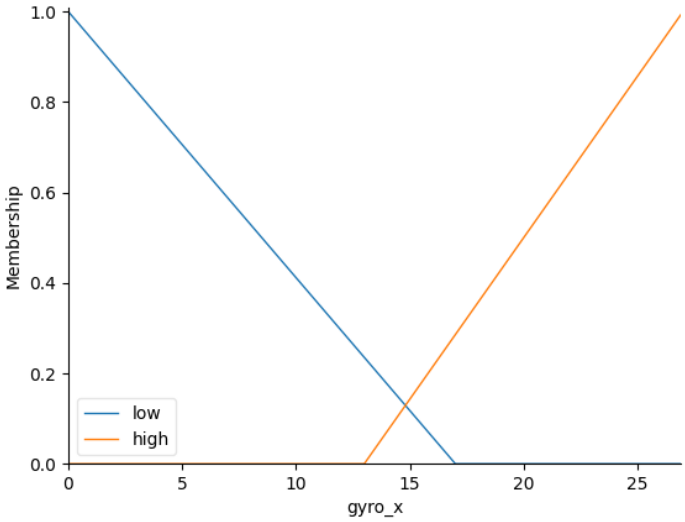


Figure 13. Fuzzy delta error set on an uneven and sloped surface (relative to the front body of the robot).

https://doi.org/10.31436/iiumej.v27i1.3587

The gain **K** varies along the y-axis for sloped surfaces, and the fuzzy Mamdani method is used to determine the appropriate gain **K** for stable robot walking. The method uses error and delta error as inputs, where error is the difference between the angle setpoint and the executed angle, and delta error is the rate of change of the error. To describe the error set on a sloped x-axis, six linguistic variables are used: NK (Small Increase), NS (Medium Increase), NB (Large Increase), TK (Small Decrease), TS (Medium Decrease), and TB (Large Decrease), as shown in Figure 12. Two linguistic variables describe the delta error: “low” (indicating a small change in error) and “high” (indicating a large change in error), as shown in Figure 13. These correspond to the symbolic fuzzy labels K (small) and B (big) used in the fuzzy rule base, even though they are not explicitly labeled in the figure.

The fuzzy error range is based on possible angle variations caused by different terrain slopes, which necessitates the fuzzy control approach for robot stabilization on the y-axis slopes. A fuzzy inference system (FIS) is then designed using the Mamdani method, which employs fuzzy rules in the form of "If X = A, Then Y = B". The fuzzy rules based on the input are shown in Table 1.

Table 1. Fuzzy Rules for the y-axis slope

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

After forming the fuzzy rules, a defuzzification process is carried out to obtain an output **Q**, which is used to calculate the appropriate gain **K** for maintaining balance on sloped surfaces. The linguistic variables representing the output **Q** are QNTS (Small Increase-Decrease), QNS (Medium), and QB (Large). The range of **Q** values are determined by finding the correct input **Q** to stabilize the robot on various sloped terrains, both uphill (0° to 10°) and downhill (0° to -10°). The fuzzy set design for the output **Q** is shown in Figure 14.

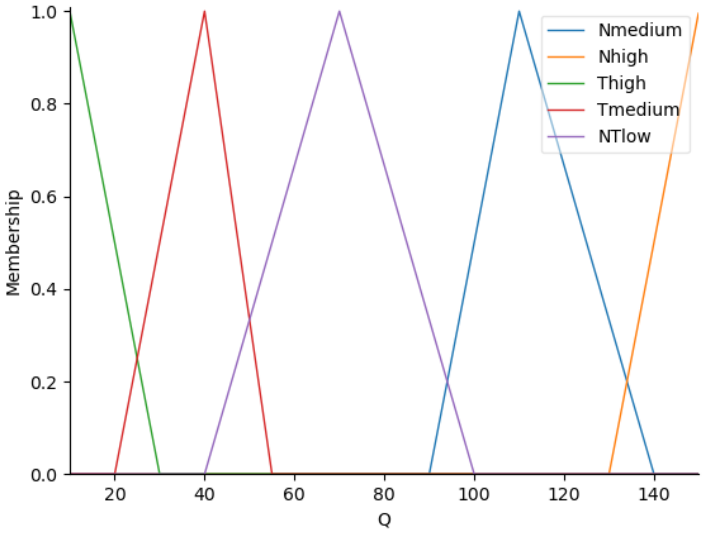


Figure 14. Fuzzy Set for **Q** on the y-axis slope

Therefore, Figure 12. Fuzzy error set on an uneven and sloped surface (relative to the front body of the robot).depicts the fuzzy membership for the input error, while Figure 14 illustrates

the output membership function for the **Q** gain adjustment, ensuring they represent different stages of the fuzzy control process.

Although advanced control techniques such as Model Predictive Control (MPC) or adaptive control may offer benefits in handling nonlinear dynamics, we consider them unnecessarily complex for the scope of this study. The proposed stability control problem is constrained to walking and sudden stopping on moderately uneven terrain, with small pitch and roll deviations (within ±0.5°). In such cases, the inverted pendulum model provides a sufficiently accurate linear approximation, making LQR a practical and computationally efficient solution.

Additionally, because the embedded system on the humanoid robot operates under limited computational resources, implementing MPC, which requires solving an optimization problem at each time step, would be unnecessary and impractical. Instead, integrating fuzzy logic into the LQR framework enables adaptive gain tuning to handle terrain variation effectively, without the complexity of full nonlinear or predictive control. This hybrid LQR–fuzzy approach thus balances controller performance, robustness, and real-time feasibility for the intended application.

Furthermore, the construction of the fuzzy rule base was performed using a hybrid approach that combined experimental observation with automatic gain computation. For each combination of body inclination error and delta error, the required LQR gain was systematically calculated to keep the COM within the predefined tolerance limits of the support polygon. These gain values were then grouped into fuzzy output sets and mapped to fuzzy rules using the Mamdani inference method. This ensures that the fuzzy-enhanced controller can dynamically tune the LQR gain in real time in response to terrain-induced disturbances, while maintaining stability within the safe operating region.

The system's controllability was verified during controller design and confirmed through simulation, ensuring that the state-space formulation supports LQR-based control. However, formal derivation is omitted here to maintain focus and limit page length. Observability analysis was not conducted because the control system employs direct full-state feedback without state estimation. For the current application, which only requires a subset of measurable states, the implemented LQR–fuzzy approach has proven sufficient. Nevertheless, in future scenarios with more complex dynamics or incomplete state measurements, the controller can be extended to a Linear Quadratic Gaussian (LQG) framework by incorporating observer-based estimation, such as the Linear Quadratic Estimator (LQE).

4. RESULTS AND DISCUSSION

4.1. Testing of Walking Robot Stabilization

All results presented in this section are obtained from real-time experiments conducted on a physical humanoid robot. The robot is a modified Bioloid Type-C equipped with 12 servo actuators and an onboard IMU. No simulation data were included in this study to maintain conciseness and practical relevance. The experimental results shown in Figures 15–19 reflect the robot's behavior under real-world disturbances, including sensor noise, actuator delays, and mechanical limitations.

Walking Stability testing aims to determine the robot's response when walking. The test was conducted with the robot's legs initially straight, parallel to the position in which both knees were bent. We then set the walking order to a stride length of 0.025 meters and a frequency of 30 Hz, using a preview-control walking pattern. This test uses the heel strategy

https://doi.org/10.31436/iiumej.v27i1.3587

as a control system. The heel strategy uses full-state feedback control that uses a gain **K** from the LQR method to control the robot's heel servo so that the robot's movement can match the walking pattern. The robot's center-of-mass response to motion was then observed and compared with the reference center of mass derived from the walking pattern.

The balance control response graph of each gain when the robot is walking is shown in Figures 15 and 16. The strengthening of **K** affects the system's feedback value, expressed as the robot's heel angle, thereby shifting the center of mass. The value of **K** is obtained by tuning the matrix **Q** in the control simulation. The most optimal **K** value is determined based on system observations by comparing the responses of each **Q** matrix. The variation of the **Q** value is based on the form of the response. This response is generated by the system and tested on the robot. The greater the **Q** value, the greater the value of **K** gain. The resulting **K** value is entered into the full-state feedback equation and will affect the torque produced. The smaller the difference between the robot's center of mass and the reference center of mass, the more optimal the system. Tuning the **Q** component in the simulation using a Python-based application produces a **K** gain, as shown in Table 2.

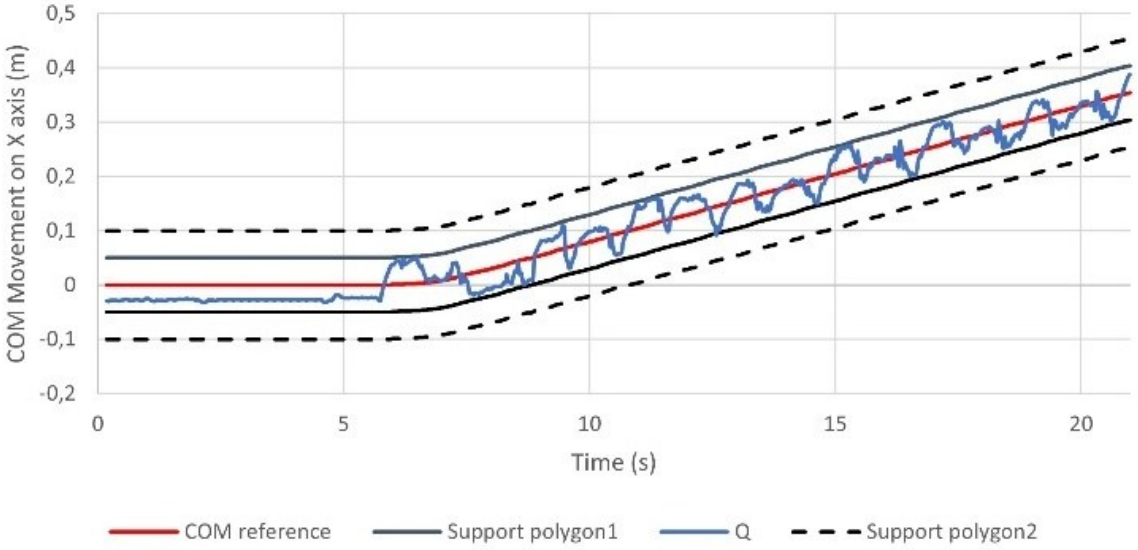


Figure 15. Balance control response on the **x** axis when the robot is walking

Table 2. The **Q** value of the LQR method to the full-state feedback **K** value.

Q	K
$\mathbf{Q}_1 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 750 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 6.06 & 3.24 & 0 & 0 \\ 0 & 0 & 29.68 & 3.51 \end{bmatrix}$
$\mathbf{Q}_2 = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 750 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 9{,}61 & 1{,}34 & 0 & 0 \\ 0 & 0 & 29{,}68 & 1{,}83 \end{bmatrix}$
$\mathbf{Q}_3 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 750 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 6{,}06 & 7{,}11 & 0 & 0 \\ 0 & 0 & 29{,}49 & 3{,}51 \end{bmatrix}$

https://doi.org/10.31436/iiumej.v27i1.3587

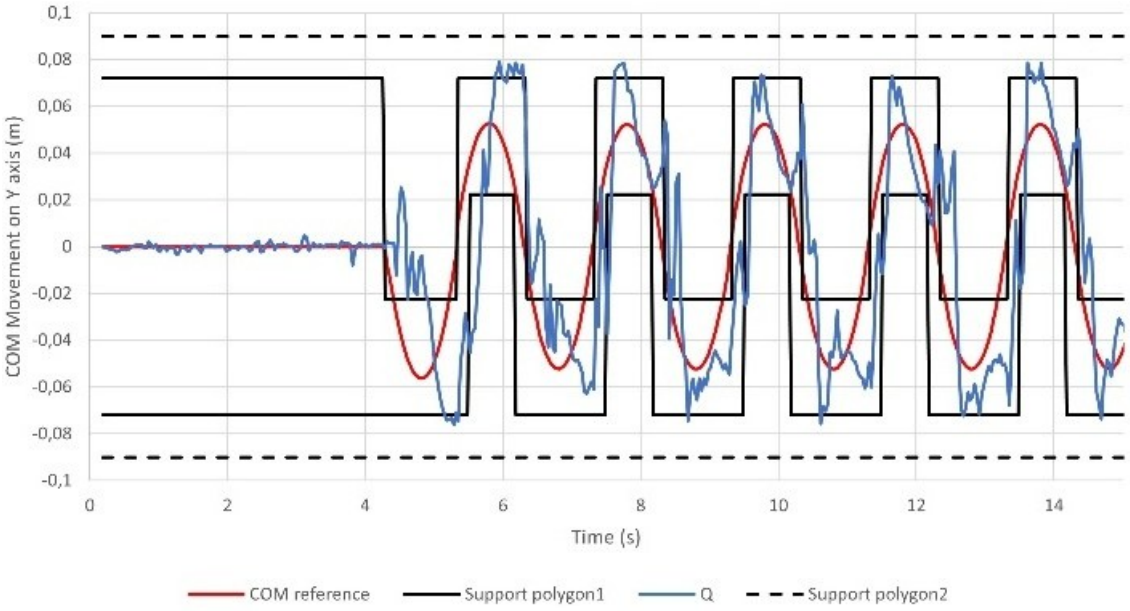


Figure 16. Balance control response on the **y** axis when the robot is walking

Based on Table 3, it is shown that the optimal value of **Q** lies in the value of Q_2 because it has the smallest maximum error among other values. This value is also considered the most optimal because the robot's center of mass is always in the support polygon on the x-axis. There is no error, and although there is an error in the y-axis, the robot can immediately overcome it, so it doesn't fall and returns to the support polygon area.

In the x and y axes control response graphs, we can see that the value of the matrix **Q** affects the robot's balance while walking. The Control system response to the x and y axes when the robot is walking are shown in Table 3.

Table 3. Control response on the x and y axes when the robot is walking.

Q	x**-axis**	y**-axis**
	Error Maximum	**Error Maximum**
$\boldsymbol{Q_1}$	1.23×10^{-2}	9.43×10^{-2}
$\boldsymbol{Q_2}$	≈ 0	8.48×10^{-2}
$\boldsymbol{Q_3}$	1.99×10^{-2}	7.89×10^{-2}

4.2. Testing of Sudden Stop Stabilization

Stability testing during a sudden stop aims to observe the robot's response when it receives an order to stop abruptly, thereby inducing a thrust from within. In testing, the robot stop indicator is implemented by adding an ultrasonic sensor to detect an obstacle in front of the robot. When the ultrasonic sensor detects an object in front of the robot at a distance of up to 10 cm, the robot stops with unspecified leg condition.

https://doi.org/10.31436/iiumej.v27i1.3587

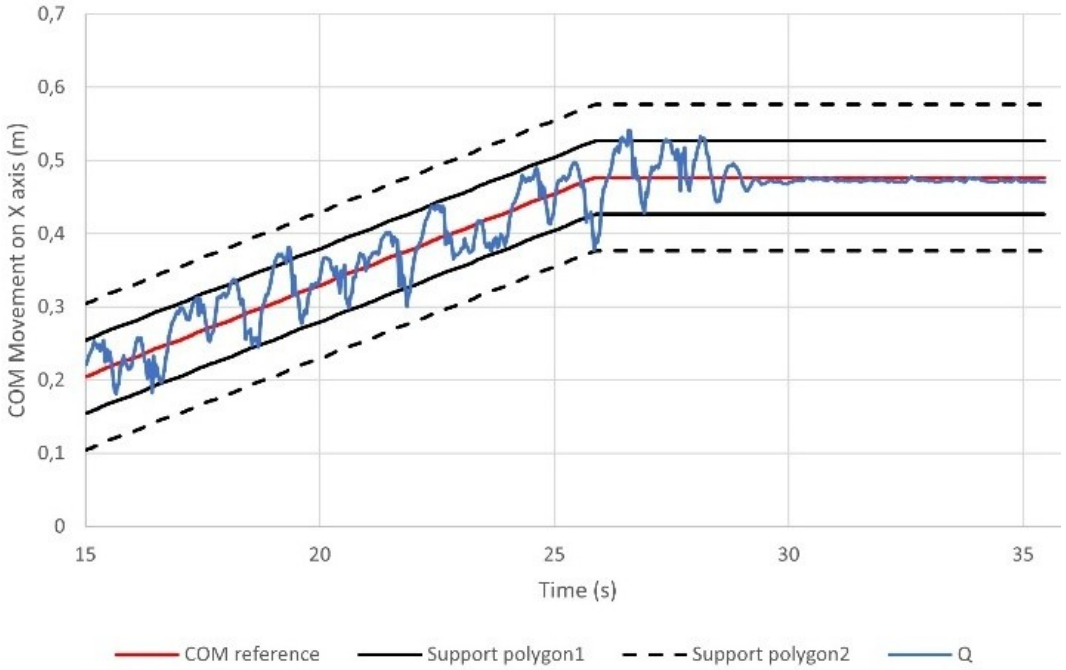


Figure 17. Balance control response on the x-axis when a sudden stop occurs.

This test uses the heel strategy as a control system. The heel strategy uses full-state feedback control that uses **K** gain from the LQR method to control the robot's heel servo so that the robot can still maintain its balance when it stops suddenly. From the previous test, the most optimal **Q** results were also used for this test. In testing, the robot overcame its instability when it was suddenly stopped at any leg state. Figure 17 and Figure 18 show a graph of the balance control response when a sudden stop occurs.

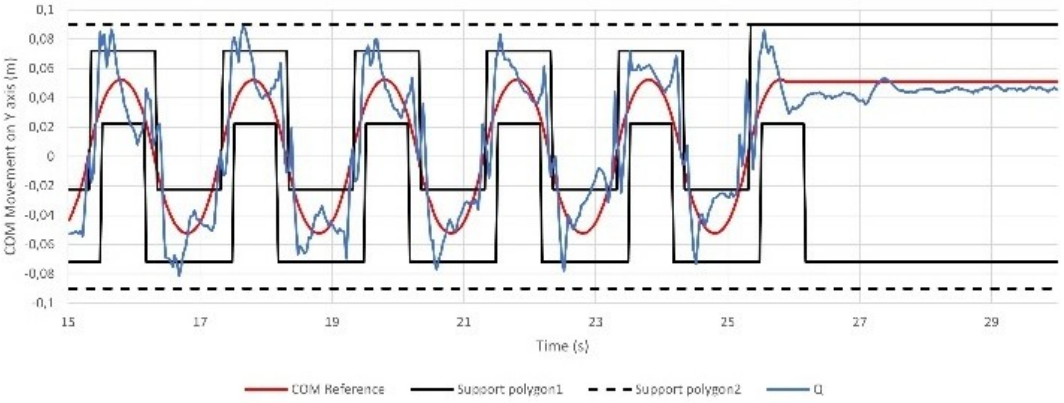


Figure 18. Balance control response on the y-axis when a sudden stop occurs.

Figure 17 and Figure 18 are graphs of the robot's balance control response resulting from the same **Q** tuning as in the previous test. Figure 17 and Figure 18 show that when the robot is suddenly stopped, the projection of its center of mass will pass the support limit of polygon 1 (SP1), thereby making the robot unstable. However, given the **K** gain from the heel strategy control system, the pedestal can directly control the robot's balance so that the projection of the center of mass can return to the polygon support, and the robot does not fall.

The heel strategy was selected due to its compatibility with the robot's limited number of actuators and the requirement for real-time computation. While more advanced balance recovery strategies, such as whole-body motion adaptation or reflex-based responses, may offer

https://doi.org/10.31436/iiumej.v27i1.3587

improved stabilization under extreme inertial forces, they demand additional degrees of freedom and higher computational capacity, which exceed the capabilities of the current platform. Future work will further explore these advanced strategies to improve robustness in high-disturbance scenarios.

4.3. Testing on Uneven Surfaces

The robot's walking performance on different terrains is illustrated in Figures 15 and 16, where the robot navigates flat terrain while maintaining its Center of Mass (CoM) within the boundaries of the walking pattern defined by the preview control system. This indicates that robots achieve balanced movement without falling. In contrast, when walking on inclined surfaces, the CoM's behavior changes significantly, as shown in Figure 19. On inclined surfaces, the CoM shifts toward the robot's support polygon, indicating increased demands on balance control. Specifically, when walking uphill, the CoM moves ahead of the predetermined walking pattern, approaching the front boundary of the support polygon as the slope angle increases. Conversely, as the slope steepens, the CoM shifts backward relative to the walking pattern, progressively approaching the rear support polygon.

Figure 19 demonstrates the relationship between the slope angle and the integral absolute error of the balance control system. The graph shows that as the slope angle increases, the integral absolute error also increases in both uphill and downhill scenarios. This indicates that maintaining balance becomes increasingly challenging as slopes become steeper. Importantly, experimental results show that when the robot walks on sloped surfaces along the y-axis, it experiences tilting along the x-axis. This tilting causes the CoM to shift outside the boundaries of the support polygon on the y-axis, leading to a loss of balance and eventual falling. This phenomenon is exacerbated as the slope angle increases, underscoring the limitations of the standard control system in maintaining stability on uneven terrain.

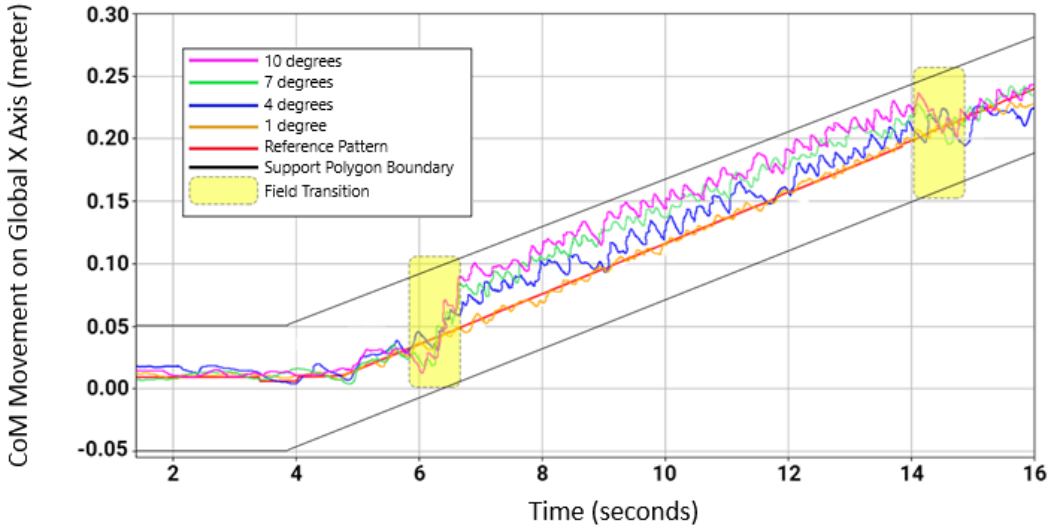


Figure 19. Balance control response when walking from a flat surface to an incline with slope variations of 1, 4, 7, and 10 degrees.

To address these challenges, fuzzy logic control is introduced, which dynamically adjusts the **K**-gain in response to variations in the slope along the y-axis. The integration of fuzzy logic enables the robot to determine optimal control parameters in real time, thereby allowing it to adapt to changing terrain conditions. The effectiveness of this system is evident when the robot transitions between flat and sloped surfaces, where the fuzzy controller selects the appropriate

K-gain to modulate the CoM's movement. This adjustment ensures that the CoM remains within the support polygon, effectively preventing the robot from falling.

Moreover, the adaptability of the fuzzy control system is crucial for enhancing the robot's overall stability on both flat and sloped surfaces. The system's ability to autonomously modify the **K**-gain based on terrain variations allows for smoother and more reliable transitions between different surface inclinations. This demonstrates the superiority of the fuzzy logic control approach over traditional methods, especially in maintaining balance on complex terrains, such as uneven or sloped surfaces.

The combination of preview control and fuzzy logic control provides a robust solution for handling varying terrain inclinations. While preview control ensures balance on flat surfaces by keeping the CoM within the walking pattern, fuzzy logic control enhances stability on sloped surfaces by adjusting the **K**-gain to maintain the CoM within the support polygon. This approach enables the robot to maintain balance effectively across a wide range of slope angles, reducing the risk of falling and improving overall walking performance on both flat and inclined terrain.

While the current experiments focused on walking over planar slopes ranging from 1° to 10°, the proposed controller structure is designed to be modular and adaptable. Terrain inclinations exceeding 10° were excluded from the current study, as the system was not designed or tuned for such extreme conditions, where stability may no longer be guaranteed under the current fuzzy-LQR configuration. The fuzzy-LQR approach was chosen for its simplicity, low computational demand, and effective performance in short-duration stabilization tasks. Although direct benchmark comparisons with other advanced balancing strategies, such as deep reinforcement learning (DRL) or adaptive impedance control (AIC), were not included in this study, such methods are typically more complex and computationally intensive. Given the application scope of this work, rapid recovery and transition to steady-state, the fuzzy-LQR method offers a practical balance between robustness, efficiency, and real-time feasibility. Future research will explore more diverse and unpredictable terrain conditions, including uneven surfaces, ground compliance variability, and dynamic obstacles. Additional sensor inputs, such as force sensors, IMUs, or vision-based terrain estimation, along with comparative evaluations against more adaptive controllers, will be considered to validate further and generalize the proposed approach.

In terms of generalizability, although the proposed control system was developed for a rigid-body humanoid robot with a specific kinematic structure, its reliance on body inclination and rate-of-change feedback enables adaptation to other bipedal robots with different degrees of freedom. Only minor tuning of the fuzzy rule base and gain scheduling would be required, without retraining or redesign of the core architecture. However, for configurations involving soft robotics or highly compliant actuators, structural modifications to the control framework may be necessary. This presents a relevant direction for future research on controller portability across robotic platforms.

This study focuses on pre-fall stability and disturbance rejection, ensuring that the robot can recover from moderate perturbations without losing balance. Fall detection and post-fall recovery strategies, such as those based on whole-body reflex motion or impact mitigation, are not included in the current framework. However, integrating such reactive mechanisms would be a valuable future enhancement, particularly for improving resilience to high-magnitude external disturbances that exceed the controller's stabilizing range

5. CONCLUSION

This study demonstrated the effectiveness of using a Linear Quadratic Regulator (LQR) and fuzzy logic control to enhance the stability of humanoid robots, particularly during walking on uneven surfaces and during sudden stops. By employing an inverted pendulum model, the robot's center of mass (CoM) was controlled to maintain balance. The LQR controller optimized the torque needed to adjust the robot's posture by minimizing the error between the reference and actual states. Additionally, the fuzzy logic control helped adapt to varying slope conditions, dynamically adjusting the gain to ensure stability. The results showed that, while the CoM closely followed the reference trajectory on flat surfaces, walking on inclined surfaces resulted in significant deviations. The use of fuzzy control allowed the robot to navigate these slopes effectively, maintaining balance by continuously adjusting the CoM's position to stay within the support polygon. The combination of LQR and fuzzy logic control is a robust approach to improving the balance and stability of humanoid robots on uneven terrain. These control methods enable the robot to maintain a stable walking pattern, even in challenging environments, making them suitable for applications in human-designed environments.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to the Department of Computer Science and Electronics at Universitas Gadjah Mada for providing the necessary resources and support to conduct this research. We are also grateful to our colleagues and reviewers for their valuable feedback and insights that significantly improved the quality of this work. Additionally, the authors acknowledge the financial support from the 2024 Academic Excellence grant awarded by the Directorate of Research, Universitas Gadjah Mada, which made this study possible.

REFERENCES

[1] S. N. Izzah, A. Dharmawan, M. Auzan, B. A. A. Sumbodo, and J. E. Istiyanto, "Stabilizing Quadruped Robot Movement Using Fuzzy Logic Control for Yaw Angle Adjustment in Walking and Troting Gait," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 12, no. 4, pp. 975–990, Dec. 2024, doi: 10.52549/ijeei.v12i4.5837.

[2] A. Dharmawan, C. Habiba, and M. Auzan, "Walking stability control system on humanoid when turning based on LQR method," *International Journal of Scientific and Technology Research*, vol. 8, no. 11, pp. 2606–2611, 2019.

[3] C. R. de Lima, S. G. Khan, M. Tufail, S. H. Shah, and M. R. O. A. Maximo, "Humanoid Robot Motion Planning Approaches: a Survey," *J Intell Robot Syst*, vol. 110, no. 2, p. 86, Jun. 2024, doi: 10.1007/s10846-024-02117-z.

[4] M. Murooka *et al.*, "Humanoid Loco-Manipulations Pattern Generation and Stabilization Control," *IEEE Robot Autom Lett*, vol. 6, no. 3, pp. 5597–5604, Jul. 2021, doi: 10.1109/LRA.2021.3077858.

[5] P. M. Viceconte *et al.*, "ADHERENT: Learning Human-like Trajectory Generators for Whole-body Control of Humanoid Robots," *IEEE Robot Autom Lett*, vol. 7, no. 2, pp. 2779–2786, Apr. 2022, doi: 10.1109/LRA.2022.3141658.

[6] L. Rossini, E. M. Hoffman, S. H. Bang, L. Sentis, and N. G. Tsagarakis, "A Real-Time Approach for Humanoid Robot Walking Including Dynamic Obstacles Avoidance," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, IEEE, Dec. 2023, pp. 1–8. doi: 10.1109/Humanoids57100.2023.10375191.

[7] S. Said, K. Youssef, B. Prasad, G. Alasfour, S. Alkork, and T. Beyrouthy, "Design and Implementation of Adam: A Humanoid Robotic Head with Social Interaction Capabilities," *Applied System Innovation*, vol. 7, no. 3, p. 42, May 2024, doi: 10.3390/asi7030042.

[8] İ. H. Şanlıtürk and H. Kocabaş, "Precise Calculation of Inverse Kinematics of the Center of Gravity for Bipedal Walking Robots," *Applied Sciences*, vol. 14, no. 9, p. 3706, Apr. 2024, doi: 10.3390/app14093706.

[9] W. Zuo, J. Gao, J. Liu, T. Wu, and X. Xin, "Whole-Body Dynamics for Humanoid Robot Fall Protection Trajectory Generation with Wall Support," *Biomimetics*, vol. 9, no. 4, p. 245, Apr. 2024, doi: 10.3390/biomimetics9040245.

[10] Vikas, D. R. Parhi, and A. K. Kashyap, "Humanoid robot path planning using memory-based gravity search algorithm and enhanced differential evolution approach in a complex environment," *Expert Syst Appl*, vol. 215, no. December 2022, p. 119423, Apr. 2023, doi: 10.1016/j.eswa.2022.119423.

[11] A. Dharmawan, A. Ashari, and A. E. Putra, "Translation Movement Stability Control of Quad Tiltrotor Using LQR and LQG," *International Journal of Intelligent Systems and Applications*, vol. 10, no. 3, pp. 10–21, Mar. 2018, doi: 10.5815/ijisa.2018.03.02.

[12] S. Singh, R. P. Russell, and P. M. Wensing, "On Second-Order Derivatives of Rigid-Body Dynamics: Theory and Implementation," *IEEE Transactions on Robotics*, vol. 40, pp. 2233–2253, 2024, doi: 10.1109/TRO.2024.3370002.

[13] A. Dharmawan, A. Ashari, A. G. Aprilia, and A. M. Handayani, "Auto VTOL System on Quadrotor Using Madgwick Quaternion Kalman Filter and LQR," in *2018 4th International Conference on Science and Technology (ICST)*, IEEE, Aug. 2018, pp. 1–6. doi: 10.1109/ICSTC.2018.8528613.

[14] G. Menga, "A Novel Paradigm for Controlling Navigation and Walking in Biped Robotics," *Electronics (Basel)*, vol. 13, no. 11, p. 2224, Jun. 2024, doi: 10.3390/electronics13112224.

[15] J. Kim, B. Park, H. Lee, and J. Park, "Hybrid Position/Torque Ankle Controller for Minimizing ZMP error of Humanoid Robot," in *2021 18th International Conference on Ubiquitous Robots (UR)*, IEEE, Jul. 2021, pp. 211–216. doi: 10.1109/UR52253.2021.9494640.

[16] R. A. Kadhim, M. Q. Kadhim, H. Al-Khazraji, and A. J. Humaidi, "Bee Algorithm Based Control Design for Two-links Robot Arm Systems," *IIUM Engineering Journal*, vol. 25, no. 2, pp. 367–380, Jul. 2024, doi: 10.31436/iiumej.v25i2.3188.

[17] R. Bimarta, A. E. Putra, and A. Dharmawan, "Balancing Robot using Proportional Integral Derivative Control Method," *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 5, no. 1, pp. 89–96, May 2015, doi: 10.22146/ijeis.7157. [in Indonesian]

[18] A. Dharmawan, J. E. Istiyanto, A. E. Putra, and M. Auzan, "LQR Compensated by Fuzzy for Kicking Balance Control of a Humanoid Robot," *ICIC Express Letters, Part B: Applications*, vol. 14, no. 5, pp. 449–456, May 2023, doi: 10.24507/icicelb.14.05.449.

[19] M. M. Kawakibi, A. Dharmawan, M. A. Jazi Eko Istiyanto, and D. Lelono, "Stability Control of Humanoid Robot Walking Speed Using Linear Quadratic Regulator Method," *ICIC Express Letters, Part B: Applications*, 2024.

[20] H. Iman and Md. R. Khan, "Repurposing A Sampling-Based Planner for A Six-Degree-Of-Freedom Manipulator to Avoid Unpredictable Obstacles," *IIUM Engineering Journal*, vol. 24, no. 1, pp. 319–332, Jan. 2023, doi: 10.31436/iiumej.v24i1.2642.

[21] M. Auzan, D. Lelono, and A. Dharmawan, "Humanoid Walking Control Using LQR and ANFIS," *Journal of Robotics and Control (JRC)*, vol. 4, no. 4, pp. 548–556, Aug. 2023, doi: 10.18196/jrc.v4i4.16444.

[22] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*, vol. 101. in Springer Tracts in Advanced Robotics, vol. 101. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. doi: 10.1007/978-3-642-54536-8.