Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 3 (2024)

Design, Simulation and Optimal Control of Two-Wheel Self-balancing Robot Using LQR and PID

Bakiya lakshmi R[1*], Bashir S. M.[2], Daniel G. S. J.[3]

[1, 2] Department of Electronics and Instrumentation Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, India

[3] Department of Mechanical Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, India

Abstract: Two-Wheel Self-Balancing Robot (TWSBR), based on the principle of Inverted Pendulum on a cart, is designed to balance vertically. Without control, it has unstable, non-linear dynamics, falling forward or backward when disturbed. This type of robot, with its flexibility to turn in narrow paths, has applications in hospitals, catering, warehouse, surveillance and many more. The linearized TWSBR model's state-space representation was analyzed at equilibrium and a full-state feedback controller was designed and simulated. This was done using Linear Quadratic Regulator (LQR) and a hybrid controller (LQR+PID), focusing on the system states namely: cart position/velocity, pendulum tilt angle/angular velocity. An STM32 microcontroller was used for digital control, with an MPU6050 sensor for angle measurement. Comparing LQR and LQR+PID, the latter showed better tilt angle response, with faster response time, lower per-centage overshoot, and faster settling time while maintaining stability. Findings of this research highlight the potential of hybrid controllers in achieving better performance and robustness in controlling two-wheel self-balancing robots.

Keywords: LQR, Self-balancing robot, Inertial Measurement Unit (MPU6050), PID Inverted pendulum, optimal stability

1. Introduction

Technological advancements in the 21st century have seen so many developments in the field of automation with the mass production of both industrial and domestic devices and machines that are faster, efficient and more convenient when used. This has significantly contributed to improving human life and creating a safer environment. One device that has garnered considerable attention in the research field is the Two-Wheel Self-Balancing Robot (TWSBR) [1]. Despite its inherent instability, the two-wheeled robots, in various applications, present a compact and user-friendly mobile platform. This necessitates an optimized system capable of delivering enhanced performance, user satisfaction, and ultimately and value for the money invested. The mobile robot finds applications in local transportation, hotels and catering, surveillance, warehouses, patient wheelchairs, and more [2]. Wide range of applications and the challenges related to stability control of mobile robot were among others the motivating factors for undertaking this project.

In this paper, design, simulation and optimal control of TWSBR are presented. The control objective is basically aimed at balancing the robot whenever an external force act on it causing a change in the vertical equilibrium state thereby prevent the TWSBR from falling. The sequence of physical actions which characterized the control action of the two-wheel self-balancing robot is a similitude of balancing a vertical bar place or fixed on a horizontal surface, for instance a stick on a flat surface of human finger. Similar concept can be applied in the Self-balancing robot system by signalling the actuator (the DC motor) to move the robot in the direction where the main body inclinedto. This forces it to maintain its equilibrium position [2-3].

1.1 Structure of the Self-Balancing Robot.

The two-wheeled balancing robot is associated with the concept of a free-fall system, emphasizing its inherent instability due to the utilization of only two wheels. In the absence of internal or external forces, this robot isprone to tipping either forward or backward, resembling an inverted pendulum that demands dynamic adjustments for maintaining equilibrium. In a scenario where the robot faces a loss of balance, the need for actuator, whether a DC motor or stepper motor becomes crucial. Its rapid response and adjustments in the direction of the impending fall are vital for counteracting destabilizing forces and ensuring that the chassis remains in an upright position.

Unlike a stable system, the two-wheeled robot constantly contends with delicate equilibrium, similar to the intricacy of balancing an inverted pendulum. This continuous balancing act necessitates ongoing adjustments by the actuator, presenting a captivating and complex engineering challenge [4]. Figure 1 below shows the free body diagram of the TWSBR. Fundamentally, the free-fall characteristic of the two-wheeled balancing robot underscores its perpetual interaction with gravitational forces and thus requires real-time control action to preserve stability. This concept not only highlights the engineering complexities involved but also forms the basis for the development of robust control strategies, ensuring the robot's effective self-balancing across diverse conditions and scenarios.

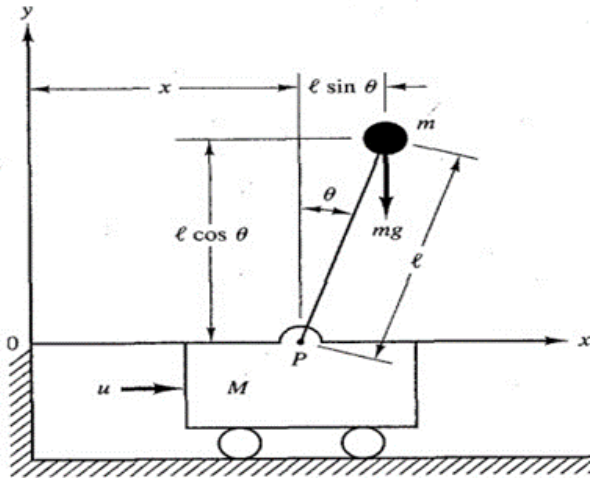

Fig. 1: TWSBR (IP) Free body diagram [4]

1.2 Outline of the paper

Section 1 of this paper introduces the project and its significance in the field of robotics and control systems. Section 2 delves into the review of existing literature of two-wheel self-balancing robots and their control algorithms. Section 3 Describes the derivation of the mathematical model of the proposed system, description of the proposed control algorithms as well as hardware and software components. Results obtained are shown in section 4 with Discussions of the results in section 5. Conclusion and future scope were presented in Chapter 6.

2. Literature Review

This section of the report provides an in-depth review on the relevant literatures.

2.1 Review of related work

The review of related works presented herein reflects on a brief timeline during which this type of robot evolves as a dynamic and multidisciplinary research field. Researchers continuously contribute to the development of more robust, adaptive, and efficient control strategies [5]. As the field progresses, the focus on achieving optimal performance, user-friendly interfaces, and robustness in real-world scenarios remains the absolute goal. Several literary works were published on this interesting topic and in one way or the other authors proposed working algorithms using the available

set of resources and technologies to achieve proper balancing of the robot. Sensors such as Infrared sensors and gyroscope were employed in balancing the robot while the likes of ultrasonic sensor, "LiDAR"sensor and others are used for obstacle avoidance and Navigation [3-5]. Back in time when curiosity and industrial demands rose significantly about new emerging technologies in automation and artificial intelligence for navigation applications, the first autonomous electronic robot known as Elsie and Elmer was introduced by William Grey Walter in the year 1948[6]. Moving fast forward, initial attempts were made to produce the inverse pendulum type robot in the year 1991-1992. In the year 1996 a 2-dimensional mobile inverse pendulum robot was designed capable of moving with constant velocity using control algorithm [7]. The most popular application of mobile robot was the Segway PT invented in 2001 by D. Kamen was the first commercialized in the year 2021 [8]. The vehicle shown in Figure 2 is powered by a battery and has the capability of balancing itself with the help of a complex control system and uses five gyroscope sensors to measure tilt angle [9]. It was the first successful commercial robot vehicle which brought Segway Inc. great fortunes and recognition. A novel approach to the design of mobile robot using linearization technique was adopted to formulate its state space equation. Researchers reported that several controllers were designed to maintain the desired angular orientation while separate other controllers were used to maintain the horizontal position of the mobile robot. A comparative was conducted to investigate the performance of Proportional–Integral–Derivative (PID), Linear–Quadratic Regulator (LQR), and Artificial Neural Network (ANN) controllers. The performance of each control algorithm was assessed after subjecting the system to external disturbance. Results of their experiments showed that Artificial Neural Network(ANN) controllers have better performance in terms of disturbance rejection [10].

Fig. 2: Segway PT [8]

The nonlinear differential equation representation of the two-wheel robot model and the linearized form of the model were analyzed in the state-space. Fixing the linearized model around the equilibrium point PID controller and an optimal linear quadratic regulator were designed and simulated with the aim of balancing the mobile robot in the vertical equilibrium position amidst external disturbances using MATLAB and the Simulink software platform. Comparing the simulation results obtained between PID and LQR, the transient response of the system with linear quadratic regulator produces faster response and lower Overshoot [11]. In another study, researchers connected an L3G4200D gyroscope to the NI motor control board, which was interfaced with the NI myRIO 1900 via the I2C protocol. The gyroscope acted as the I2C slave, while the myRIO module served as the I2C master. The angular velocity in degrees per second was computed by multiplying the raw output of the gyroscope by 0.00875 degrees/seconds/LSB. This angular velocity was then fed through an integrator block to obtain the angular displacement. The researchers also implemented a path planning algorithm in LabVIEW software, a graphical-based programming platform, to find the shortest path for the robot to navigate from the start to the end position. An obstacle detection system was incorporated into the model prototype using an ultrasonic sensor for collision avoidance. The upright balancing on two wheels was achieved with 99% accuracy [12]. The Linear Quadratic Regulator (LQR) algorithm was applied to control the state space model of the Two- Wheel mobile robot, ensuring its equilibrium in the upright position. Programming and analysis were conducted using the Rensselaer Arduino Support Package and Simulink support package for Arduino hardware. Acknowledging the sensitivity and critical nature of the Q and R

values in the LQR regulator, which significantly influence the accuracy of the desired control signals. The study recommends the incorporation of a Kalman filter to mitigate noise from the gyro data. [13]. Recent studies have focused on improving the stability, control, and performance of two-wheel self-balancing robots. The integration of advanced sensors, such as gyroscopes and accelerometers, has enabled precise measurement of orientation and motion, thereby enhancing the balancing capabilities of these robots. Researchers have also utilizedPID control techniques, including the Ziegler-Nichols and Cohen-Coon tuning methods, to achieve efficient control and stability [14]. The design and control of two-wheel self-balancing robots pose several challenges, including maintaining dynamic balance, responding to external disturbances, and achieving accurate positioning. Researchers have addressed these challenges through innovative control algorithms, sensor fusion techniques, and robust hardware implementations. The use of feedback systems, motor control modules, and microcontrollers, such as the Arduino Mega 2560, has facilitated precise control and real-time adjustments in these robots [15-16]. Two-wheel self-balancing robots have found diverse applications across various industries, including surveillance, logistics, entertainment, and education. These robots offer mobility, agility, and versatility, making them suitable for tasks that require dynamic movement and precise control. The integration of Bluetooth modules, such as the HC-05, has enabled wireless communication and remote-control options, enhancing the functionality and usability of these robots [16-18]. Ryuichi T et a proposed a self-balancing obstacle avoidance robot using a combination of PID and Fuzzy Logic Controller (FLC) algorithms. The FLC takes input distance data between the robot and obstacles, and outputs control commands that steer the robot away from obstacles. The proposed method utilizes a cost-effective, fixed-point digital signal processor (DSP) control system that reduces computational complexity and costs compared to floating-point arithmetic. The system's calculations were tested and compared against those of a conventional software. implementation, showing that the custom instructions were faster than libfixmath. The authors' simulation results revealed that the proposed method has lower computational complexity and cost compared to other approaches such as dynamic window algorithm and potential field method [19]. Aldhalemi et al. presented the design and implementation of a remotely controlled two-wheel self-balancingrobot. The control system's hardware development consists of two Arduino boards, a Bluetooth HC05, a gyroscope sensor, a stepper motor, and a motor driver. The robot is controlled through an Android GUI application on a smartphone. The authors analyzed the dynamic model and control algorithms for the robot's responses in moving mode. They also discussed potential applications of the proposed robot in surveillance, exploration, and entertainment, and suggested further research for system optimization. However, the use of a stepper motor may limit the long-term usage of the robot due to its higher battery drain capability compared to other types of DC motors [20]. Nguyen H.V. et al. proposed a balancing method for a unicycle robot using a PD controller. The authors developed a PD controller for a two-wheeled self-balancing robot and a reaction wheeled inverted pendulum. They performed simulations and experiments to find a suitable set of PD controllers and calibrate the control parameters, with a focus on the proportional gain Kp. The authors used genetic algorithms and trial-and-error tests to find a set of PD controllers that balance the unicycle robot well. They noted that increasing Kp causes the system to vibrate more, while decreasing it results in insufficient control signals. The paper includes comparison graphs of system response for different cases, demonstrating the success of the proposed PD control method in balancing the unicycle robot [21].

2.2 Review analysis

This review highlights the evolution and development of two-wheel self-balancing robots as a dynamic and multidisciplinary research field, with significant milestones from the first autonomous electronic robot, Elsie and Elmer, in 1948 to the commercialization of the Segway PT in 2021. Researchers have utilized various sensors, including Infrared sensors, gyroscopes, ultrasonic sensors, and LiDAR sensors, for balancing and navigation. Recent research has focused on enhancing stability, control, and performance, with researchers integrating advanced sensors and PID control techniques for efficient control and stability. The literature survey also highlights the diverse applications of two-wheel self-balancing robots across various industries, including surveillance, logistics, entertainment, and education. The integration of Bluetooth modules has enabled wireless communication and remote-control capabilities, enhancing functionality and usability. Recent studies on novel control algorithms provide insights into the future directions of research in this field. The study demonstrates the potential of hybrid controllers in

achieving better performance and robustness in controlling two- wheel self-balancing robots [22-24]

3. Methodology

3.1 Model Preparation

This section of the paper presents a detailed description of the proposed electromechanical system. It is an unstable system with a basic framework built on the principle of Inverted pendulum. A Set of differential equations describing the laws governing the motion of the robot were developed and expressed in state-space form. The proposed robot is designed in such a way that the chassis (pendulum arm) which stands on a horizontal platform about a movable pivot must balance itself, otherwise it will fall in either forward or backward direction. A key component in this project work is the Inertial measurement unit (MPU6050) carrying in it a gyroscope and accelerometer measures angular position of the chassis and provide such signal in closed loop feedback to the PID controller for generating the required control command.

3.2 Mathematical modelling

Three sub-sections of the physical model namely: the Chassis, DC motor and the wheels were presented here. analyzing the forces acting at a given reference point in each of the subunits, differential equations were developed using Newton's second law of motion. The major parts of the inverted pendulum as shown in Figure 3 include a movable cart of mass, M rolling sideways on two identical wheels, pendulum arm of mass m hinged on a movable cart. The center of gravity is located at a distance *l* from one end which is half the length of the pendulum arm for a uniformly distributed arm, the tilt angle θ.

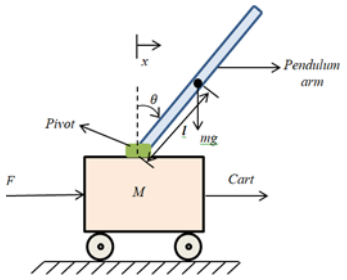

Fig. 3: Sketch Diagram of Inverted pendulum

Key model parameters of the self-balancing robot used in this paper are:

Scalars:

- r = Wheel radius
- Mw = Wheel mass
- Iw = Wheel moment of inertia,
- Mp = Mass of the Chassis,
- Ip = Chassis moment of Inertia,
- R = terminal resistance of the DC motor

Vectors:

- $\boldsymbol{F}$ = force applied on the cart
- $\boldsymbol{x}$ = displacement of the cart
- $\boldsymbol{Km}$ = Motor torque,
- $\boldsymbol{Ke}$ = Back EMF constant
- $\boldsymbol{g}$ = Gravitational acceleration,

3.3 The DC motor model

Figure 4 shown below describes the circuit diagram of the DC motor model.

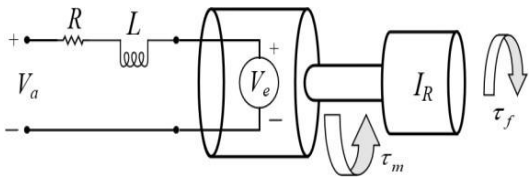

Fig. 4: DC motor Circuit Diagram [24]

Applying the Kirchhoff's voltage law on the electric circuit shown in Figure 4 gives:

$$V_a = + L \frac{di}{dt} + V_e \quad (1)$$

Where V_a and V_e are the Terminal voltages and the back Electromotive force (EMF) respectively. 'L' is the inductance and 'R' is the terminal resistance and *'I' is the* electrical current through the motor stator windings.

The torque due to friction was neglected to simplify the model. Therefore, based on this assumption, the total torque developed is given by:

$$\tau_m = \frac{V_a - V_e}{K_m R} \quad (2)$$

The back EMF is a linear function of the angular velocity (ω) as shown in equation below.

$$V_a = K_e \omega \quad (3)$$

Substituting equation (3) into equation (2) gives:

$$I_W \dot{\omega} = \frac{k_m}{R} V_a - \frac{k_m k_e \omega}{R} \quad (4)$$

Since the rotor speed of the DC motor is given by $\omega = \dot{\theta}$. Therefore, the linear model of the DC motor can be writtenas:

$$\dot{\omega} = \frac{k_m}{I_W}\left(\frac{V_a}{R} - \frac{k_e \omega}{R}\right) \quad (5)$$

- **The wheel model.**

Figure 5 shown below illustrates the free body diagram of the wheels.

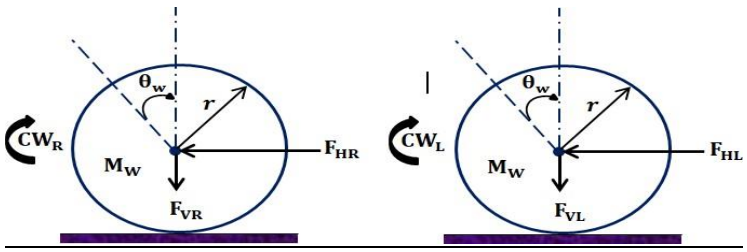

Fig. 5: Free Body Diagram of the Wheels

The Newton's second law of motion is applied on equation 5 to obtain the equations governing the movement of boththe right and left wheels as follows: -

The forces *HR* and *HL* are the frictional forces on the right and left wheels respectively. Combining equation 5 and 6 gives:s

$$M_W\ddot{x} = \frac{k_m}{Rr}V_a - \frac{k_m k_e}{Rr^2}\dot{x} - \frac{I_W}{r^2}\ddot{x} - F_{HR} \qquad (6)$$

$$M_W\ddot{x} = \frac{k_m}{Rr}V_a - \frac{k_m k_e}{Rr^2}\dot{x} - \frac{I_W}{r^2}\ddot{x} - F_{HL} \qquad (7)$$

$$2\left(M_W + \frac{I_W}{r^2}\right)\ddot{x} = -2\frac{k_m k_e}{Rr^2}\dot{x} + 2\frac{k_m}{Rr}V_a - (F_{HR} + F_{HL}) \qquad (8)$$

- **The Chassis model.**

Figure 6 below shows the free body diagram of the Chassis (pendulum arm).

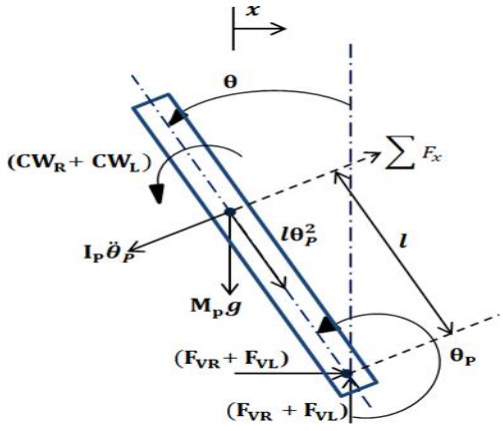

Fig. 6: Chassis free body diagram

Applying second Newton's law of motion in terms of moment of Inertia, ***IP*** of the pendulum and the distance, ***l*** from the pivot point to the pendulum's center of mass and integrating with the DC model we have the following equations [24]: -

$$(I_p + l^2 M_p)\ddot{\theta}_p - \frac{2k_m k_e}{Rr}\dot{x}_a + \frac{2k_m}{Rr}v_a + M_p g l \sin\theta_p = -M_p l \ddot{x}\cos\theta_p \quad (9)$$

$$\frac{2k_m}{Rr}v_a = 2\left(M_W + \frac{2I_W}{r^2} + M_p\right)\ddot{x} + \frac{2k_m k_e}{Rr^2}\dot{x} + M_p l\ddot{\theta}\cos\theta_p - M_p l\dot{\theta}^2\sin\theta_p \quad (10)$$

It is obvious that equations (9) and (10) above are non-linear in nature. To linearize the equations, we set $\theta_p = \pi + \theta$ obtaining the following equations:

$$\ddot{\theta} = -\frac{M_p l}{(I_p + l^2 M_p)}\ddot{x} + \frac{2k_m k_e}{Rr(I_p + l^2 M_p)}\dot{x} - \frac{2k_m}{R(I_p + l^2 M_p)}v_a + \frac{M_p g l}{(I_p + l^2 M_p)}\theta \quad (11)$$

$$\ddot{x} = -\frac{M_p l}{\left(2M_W + \frac{2I_W}{r^2} + M_p\right)}\ddot{\theta} - \frac{2k_m k_e}{Rr\left(2M_W + \frac{2I_W}{r^2} + M_p\right)}\dot{x} + \frac{2k_m}{R\left(M_W + \frac{2I_W}{r^2} + M_p\right)}v_a \quad (12)$$

Let us consider the state-space representation of the multivariable Linear Time Invariant system of the form:

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \quad (13)$$

$$y = \mathbf{C}x + \mathbf{D}u \quad (14)$$

$\mathbf{A} \in \mathbb{R}^{n\times n}$ is the system matrix.
$\mathbf{B} \in \mathbb{R}^{n\times m}$ is the input matrix.
$\mathbf{C} \in \mathbb{R}^{k\times n}$ is the output matrix.
$\mathbf{D} \in \mathbb{R}^{k\times m}$ is the feedforward matrix.

Combining the equations (11) and (12) gives the state-space model of the two-wheel self-balancing robot as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m k_e (M_n lr - I_n - M_n l^2)}{Rr^2 a} & \frac{M_n^2 l^2 g}{a} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_m k_e (r\beta - M_p l)}{Rr^2 a} & \frac{M_p g l \beta}{a} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_m(-M_p lr + I_n + M_p l^2)}{Rr^2 a} \\ 0 \\ \frac{2k_m(-r\beta + M_p l)}{Rr^2 a} \end{bmatrix} \cdot [u] \quad (15)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (16)$$

Here:

$$\beta = 2M_W + \frac{2I_W}{r^2} + M_p \quad \text{and} \quad \alpha = I_p\beta + M_p l^2\left(M_W + \frac{I_W}{r^2}\right)$$

Where the chosen state variables are defined as follows:

x_1 = displacement of the cart(x) x_2 = linear velocity of the cart($\dot{x}$)

x_3 = tilt angle of the pendulum arm(θ)

x_4 = angular speed of the pendulum arm ($\dot{\theta}$)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m k_e (M_p l r - I_p - M_p l^2)}{Rr^2 a} & \frac{M_p^2 l^2 g}{a} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_m k_e (r\beta - M_p l)}{Rr^2 a} & \frac{M_p g l \beta}{a} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_m (-M_p l r + I_p + M_p l^2)}{Rr^2 a} \\ 0 \\ \frac{2k_m (-r\beta + M_p l)}{Rr^2 a} \end{bmatrix} \cdot [u] \quad (15)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (16)$$

Here:

$$\beta = M_w + \frac{2I_w}{r^2} + M_p \quad \text{and} \quad a = I_p \beta + M_p l^2 \left(M_w + \frac{I_w}{r^2}\right)$$

Where the chosen state variables are defined as follows:

x_1 = displacement of the cart(x) x_2 = linear velocity of the cart($\dot{x}$)

x_3 = tilt angle of the pendulum arm(θ)

x_4 = angular speed of the pendulum arm ($\dot{\theta}$)

3.4 Control Algorithm

3.4.1 The PID

The PID controller, a versatile control scheme, is employed to regulate various process parameters such as angular speed, temperature, pressure, and speed etc. Operating on closed loop feedback, it calculates the disparity between the actual and desired values, generating a control signal for the actuator. With tenable parameters including the Proportional, Integral, and Derivative, the PID control minimizes rise time, eliminates steady-state error, and suppresses overshoot. This helps in achieving a precise match between the reference set-point and the output signal.

Proportional Gain (P or K_P): The P-action is directly proportional to the error or process variable (PV). Multiplying the error (or PV) by the proportional gain and adding it to the controller output imparts a corrective 'kick'in the appropriate direction. A controller relying solely on P action requires a non-zero error for a non-zero output, making precise tracking unattainable.

Integral Gain (I or K_I): The I-action operates at a slower pace compared to the proportional action but gradually reduces the error to zero, a task the proportional action alone cannot accomplish. The integral action considers past performance, checking if the error converges to the set-point; if not, it adjusts the output, guiding the system toward the intended direction.

Derivative Gain (D or K_d): Unlike the integral action, which lacks the ability to predict error behaviour, the derivative action anticipates future error changes. It contributes to the output based on how the error is changing, acting as a brake to prevent overshooting when the error is positive but declining. The D action reduces oscillations induced by other actions, potentially, speeding up the controller's approach to the desired reference signal.

The PID output equation is given by:

$$u(t) = K_p e(t) + K_i \int e(t)\,dt + K_d \frac{d}{dt} e(t) \quad \text{— (17)}$$

3.4.2

Linear Quadratic Regulator (LQR)

The control objective is to determine the matrix gain K such that for any initial state x(0), the control law

$u(t) = -Kx(t)$(18)

Minimizes the performance index:

$$J = \int_0^\infty (x^T Q x + u^T R u)\,dt \quad (19)$$

$$where\ Q \geq 0\ and\ R > 0$$

Subject to the dynamics of the systems represented in state space model (equation 13 and 14) **A, B, C,** and **D** represent the dynamics of the system and the optimal feedback gain is given by:

$$K = R^{-1}B^T P \quad (20)$$

P is obtained from the State Dependent Riccati Equation given by:

$$\mathbf{A}^T\mathbf{K} + \mathbf{PA} - \mathbf{PBR}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0 \quad (21)$$

In this case **K** = $[K_1\ K_2\ K_3\ K_4]$ which are all computed in the MATLAB environment.

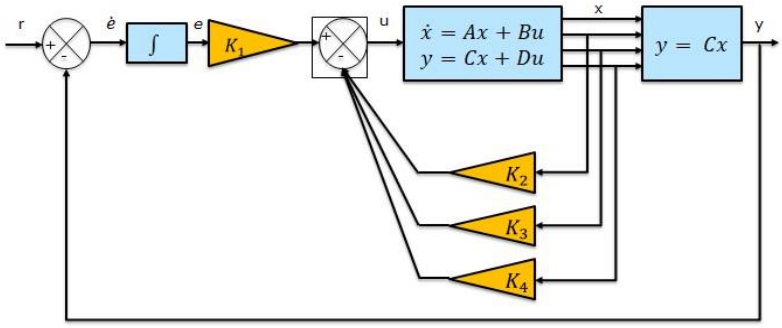

Fig. 7: Block Diagram of the LQR control scheme

3.5 Hardware.

The MPU6050 sensor is a Micro-Electro-Mechanical Systems (MEMS) module made up of a 3-axis Accelerometer and 3-axis Gyroscope as shown in Figure 8. It is used primarily for measuring acceleration, orientationand other motion-related parameters of a system depending on the specific application. The module is also equipped with a Digital Motion Processor (DMP) which is a powerful tool for performing complex operations. The module also comes with

two auxiliary pins which can be used to interface with Inter-Integrated Circuit (I2C) modules. Since the I2C address of the module is configurable, it therefore means that more than one MPU6050 sensor can serially interface to communicate with a single Microcontroller. This module is open-sourced and has fully documented libraries. These features simplify its usage across different development platforms.

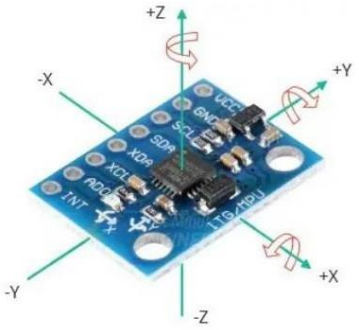

Fig. 8: Inertial Measurement Unit (MPU06050)

The NUCLEO-F401RE development board built with STM32F401REE microcontroller based on Cortex- M4 architecture from STMicroelectronics was selected for this project. The development board can be powered with a DC voltage range of 1.7V-3.6V in addition to a wide range of 7V–12V and has a 32-bit RISC core operating at maximum frequency of 4MHz with a floating–point unit (FPU). It is equipped with 512Kilobytes flash memory for program storage offering 96 Kilobytes of RAM for data storage. It also has 50 GPIO pins with external interrupt capability as well as 12-bit ADC pins on board. Communication interfaces available on the board include USART, SPI, I2C and USB 2.0 OTG FS. STM32CubeIDE is used for programming the board which also has Arduino UnoV3 connectivity and ST morpho headers [25]. The development board support for advanced motor control makes it suitable for precise motor control required in this project. Figure 9 shows the front image of the development board. Direct current (DC) geared motor is chosen to drive the wheels whenever the tilt angle deviates from thereference set-point.

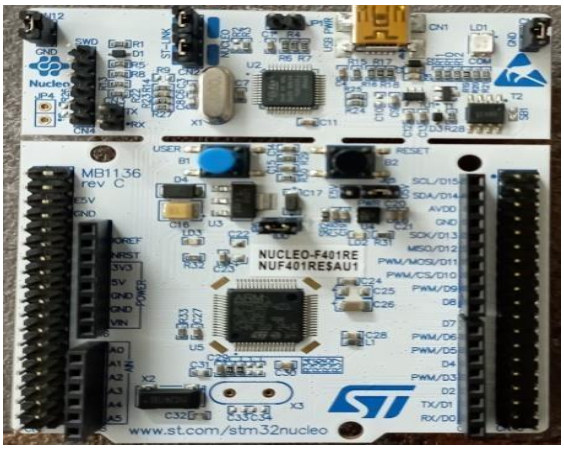

Fig. 9: NUCLEO-F401RE development board [25]

4. Results

This section of the paper presents the simulation results obtained during the project implementation. Using the state space model presented in equation (15) and (16), a MATLAB program for the parameter setting and generation of the model matrices was developed as shown in Figure 9. Table 1 shows the parameters of the developedtwo-wheel robot.

The moment of inertia for the chassis and the wheels were calculated using equation (22) and (23) respectively:

$$I = \frac{1}{12} M (h^2 + w^2) \quad (22)$$

$$I_W = \frac{1}{2} m_W r^2 \quad (23)$$

After arranging the simulation setup, the first test conducted was that of controllability test to determine whetherthe state variables of the system could be steered from any initial state to another state by applying suitable control action. Equation 24 gives the controllability matrix Q_C as follows:

$$Q_c = [B \;\; AB \;\; A^2B \;\; A^3B \ldots \;\; A^{N-1}B] \quad (24)$$

Where: N is the order of the system. Here, the determinant of is found to be:

$$|Qc| \neq 7.795 \times 10^{-19} \quad (25)$$

Since the determinant of the controllability matrix $|Qc| \neq 0$, the rank of the matrix Qc is N, thus the systemis completely state controllable.

```
%PARAMETERS OF THE SELF-BALANCING ROBOT
r=0.035;       %Wheel radius (m)
Mw = 0.3;        %wheel mass (kg)
Iw = 0.18375;     %wheel inertia (kgm^2)
l = 0.010;       %distance from body's centre
                % of mass (m)
Mp = 0.404;       %Mass of the body (kg)
Ip = 0.001988;     %body inertia (kgm^2)
Km = 0.0343;    %motor torque (Nm/Amp)
Ke = 0.0458;    % Back EMF constant (Vs/radians)
g = 9.81;       %gravitational acceleration (m/s^2)
R = 33.3;       % Terminal resistance (Ohms)

%Intermediate Calculation
q = 2*Mw + 2*Iw/r^2 + Mp;             %beta
p = Ip*q + Mp*l^2*(Mw + Iw/r^2);     %alpha

%State_Space matrices
A=[0, 1, 0, 0;
    0, 2*Ke*Km*(Mp*l*r-Ip-Mp*l^2)/R*r^2*p,
Mp*g*l^2/p, 0;
    0, 0, 0, 1;
    0, 2*Ke*Km*(r*q - Mp*l)/R*r^2*p, Mp*g*l*q/p, 0];

B = [0; 2*Km*(-Mp*l*r + Ip + Mp*l^2)/R*r*p; 0;
    2*Km*(-r*q + Mp*l)/R*r*p];
C = [1 0 0 0; 0 0 1 0];
D = [0;0];
```

Fig. 10: Model parameter settings

The poles of the open-loop system are found to be:

$$\lambda = \begin{bmatrix} 0 \\ -1.5628e-10 \\ -4.4425 \\ 4.4425 \end{bmatrix} \quad (26)$$

It could be seen from the equation (26) that only two of the open-loop poles are located on tt half of thecomplex s-plane, one is located at the origin and the other at the right half of the complex s-plane. Figure 11 shows unbounded output (Tilt angle) response to a bounded step input. This is due to the existence of the open-lop pole on the right half of the complex s-plane. This made it clear that the open-loop system of the robot is unstable.

Table 1: Parameters of the developed robot

Parameter	**Value**	**Unit**
Wheel radius, *r*	0.035	m
Wheel mass, *Mw*	0.300	Kg
Wheel inertia, I_w	0.18375	kgm^2
distance from body's centre, *l*	0.005	m
Mass of the Chassis, *Mp*	0.404	Kg
Chassis Inertia, *Ip*	0.001988	kgm^2
Motor torque, *Km*	0.0458	Nm/Amp
Back EMF constant *Ke*	0.0458	Vs/radians
Gravitational acceleration, *g*	9.81	m/s^2
Terminal resistance, *R*	33.3	Ohms

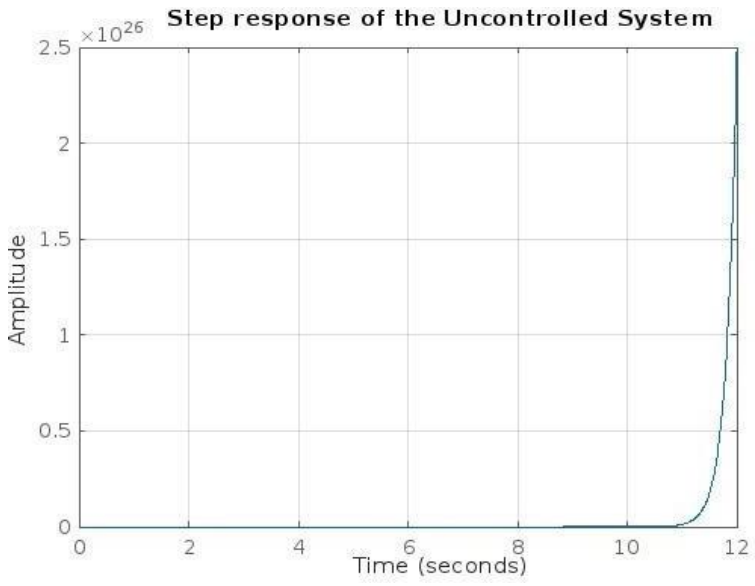

Fig. 11: Step response of the uncontrolled robot

```
%Parameters of the LQR
Q=C'*C;
R=1;
[K, P, E]=lqr(system1,Q,R,0)

K =
   1.0e+04 *

   -0.0001   -0.4530   -8.6006   -1.9360

P =
   1.0e+10 *

    0.0000    0.0010    0.0000    0.0000
    0.0010    4.6469    0.0088    0.0018
    0.0000    0.0088    0.0833    0.0187
    0.0000    0.0018    0.0187    0.0042
E =

  -0.0002 + 0.0002i
  -0.0002 - 0.0002i
  -4.4425 + 0.0001i
  -4.4425 - 0.0001i
```

Fig. 12: Parameter settings of the LQR

Figure 12 shows the parameter settings of the Linear Quadratic Regulator as implemented in the MATLAB environment. The result shows critical value of LQR gain which is used for regulating the desired state variables to zero. Having obtained the gain constant K of the LQR, its value was used to implement LQR whose block diagram shown in Figure 7. The result is shown in Figure 13 in which the robot's displacement is made to track a reference signal while the remaining three state variables namely linear velocity, tilt angle, and the angular velocity are regulatedto zero. Figure 14 shows the block diagram of the hybrid model.

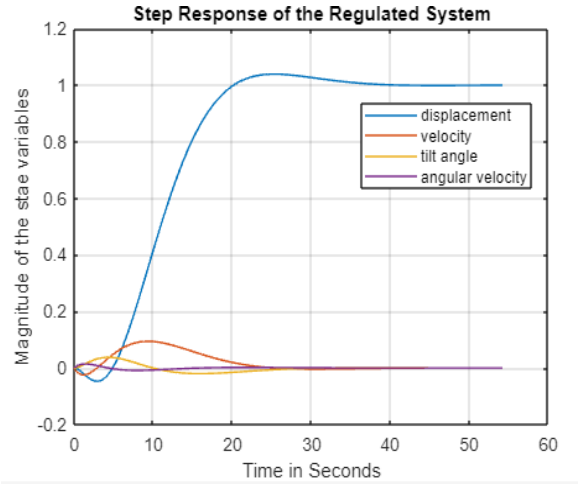

Fig. 13: LQR Step response with zero initial condition

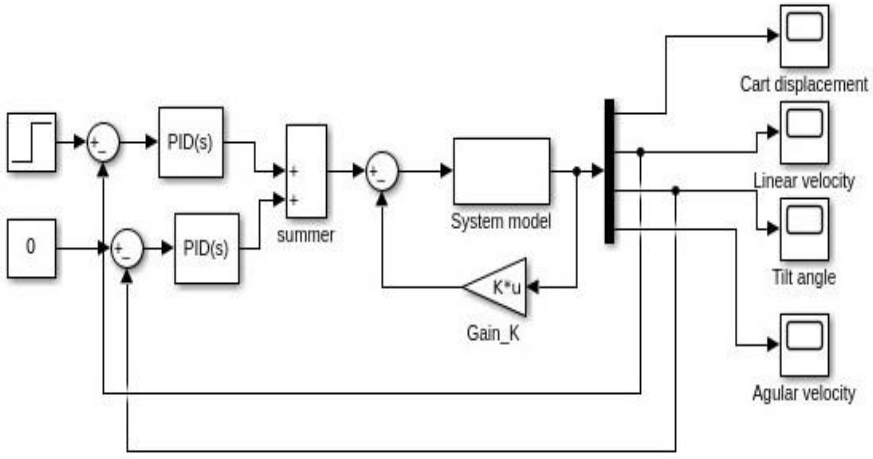

Fig. 14: Block diagram of the hybrid model

4.1 Tilt Angle Balance Testing:

The ability of the model to balance itself was tested by setting the initial tilt angle to 0.15rad (27^0 deviation)while the reference angle was kept at 0 rad i.e. at the vertical equilibrium position. The simulation results obtained from LQR, and hybrid control controller (LQR+PID) are shown in Figure 15 and Figure 16 respectively. The tuned PID parameters for the balance testing are (K_P = 105, K_I = 15.8 and K_D = 180) and for the LQR, the performance matrices are Q = C'*C and R=1. It could be seen from the respective plots that the hybrid control action of combinedLQR and PID produced a tilt angle response with faster response time, lower % overshoot and faster settling time.

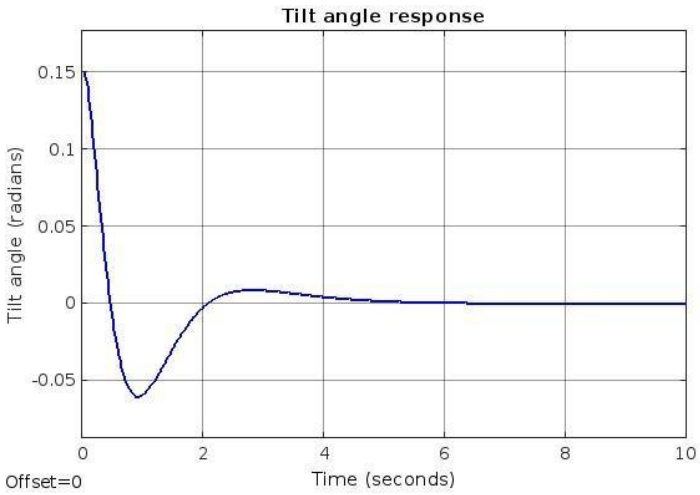

Fig. 15: Tilt angle response for LQR

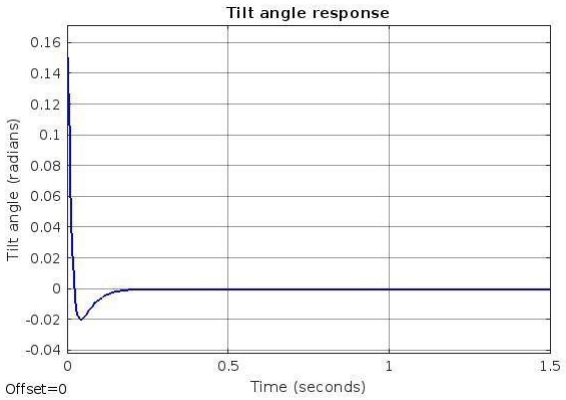

Fig. 16: Tilt angle response for LQR+PID

Table 2: Performance metrics for Tilt angle control

Controller Type	Performance Metric			
	% Overshoot -	Rise time t_r(s)	Settling time t_s(s)	Steady-state error e_{ss}
LQR	40.0	1.000	5.00	0
PID+LQR	13.3	0.002	0.19	0

4.2 Velocity control testing

The goal of the velocity control is to test the ability of the mobile robot to move in the direction of the tilt angle while trying to maintain the vertical equilibrium position. In this case, the reference tilt angle is kept at 0 rad and the reference velocity is set to 0.15ms^{-1} (or 15 cm per second). The tuned PID parameters for the balance testing are (K_P= 109, K_I = 8.7 and K_D= 22) and for the LQR the performance matrices are Q = C'*C and R=1. Figures 17 and Figure 18 show the linear velocity response of the LQR and the hybrid controller respectively.

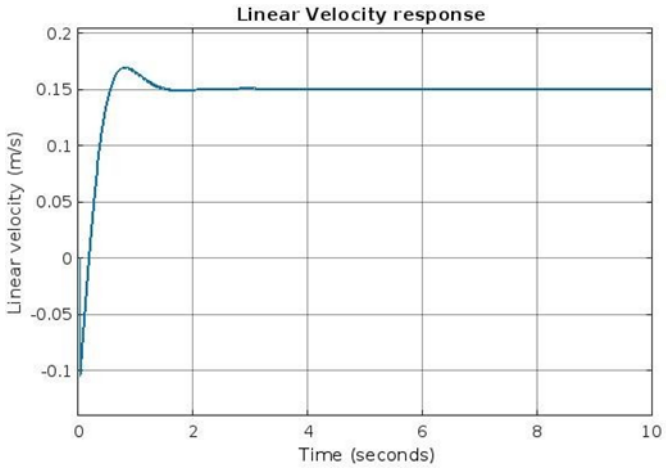

Fig. 17: Linear velocity response for LQR

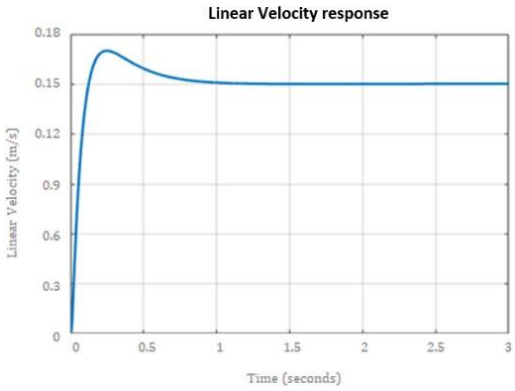

Fig. 18: Linear velocity response for LQR+PID

Table 3: Performance metrics for linear velocity control

Controller Type	Performance Metric			
	% Overshoot -	Rise time tr(s)	Settling time ts(s)	Steady-state error ess
LQR	16.67	0.50	1.50	0
PID+LQR	10.00	0.13	1.00	0

5. Discussion

This section discusses the results obtained, which aimed to compare the performance of two control strategies for two-wheel self-balancing robots. The findings demonstrate the potential of hybrid controllers in achieving better performance and robustness in controlling these robots, highlighting the significance of integrating advanced sensors and control techniques for optimal control.

5.1 Tilt angle response

Considering the parameters obtained from the tilt angle response, the hybrid controller algorithm outperforms the LQR controller in several aspects when controlling the tilt angle response of model mobile robot.

i. Percentage overshoot: The hybrid controller has a significantly lower percentage overshoot (13.3%) compared to the LQR controller (40%). This indicates that the hybrid controller has better stability and reduces the risk of oscillations or instability, which is crucial for maintaining balance in a two-wheel self- balancing robot.

ii. Rise time: The rise time of the hybrid controller (0.002 seconds) is considerably faster than that of the LQR controller (1 second). A faster rise time implies that the hybrid controller can quickly respond to changes in the tilt angle, allowing the robot to maintain balance more efficiently.

iii. Settling time: The settling time of the hybrid controller (0.19 seconds) is also much shorter than that of the LQR controller (5 seconds). A shorter settling time indicates that the hybrid controller can reach the desired tilt angle more quickly, which is essential for the robot's stability and manoeuvrability.

iv. Steady-state error: Both the LQR and hybrid controller have zero steady-state error, which means that the tilt

angle converges to the desired value without any steady-state error.

The hybrid controller algorithm demonstrates superior performance compared to the LQR controller in terms of percentage overshoot, rise time, and settling time. These improvements are crucial for maintaining balance and ensuring the stability and manoeuvrability of a two-wheel self-balancing robot. The hybrid controller's ability to quickly respond to changes in tilt angle and reach the desired angle more efficiently makes it a more suitable choice for this application.

5.2 Velocity Response

Considering the parameters obtained from the velocity response, the hybrid controller algorithm here also outperforms the LQR controller in several aspects when controlling the velocity response of a mobile robot as discussed below: -

i. Percentage overshoot: The hybrid controller has a lower percentage overshoot (10.0%) compared to the LQR controller (16.67%). This indicates that the hybrid controller has better stability and reduces the risk of oscillations or instability, which is crucial for maintaining the desired velocity in a mobile robot.

ii. Rise time: The rise time of the hybrid controller (0.13 seconds) is significantly faster than that of the LQR controller (0.50 seconds). A faster rise time implies that the hybrid controller can quickly respond to changesin velocity, allowing the robot to reach the desired speed more efficiently.

iii. Settling time: The settling time of the hybrid controller (1 second) is slightly shorter than that of the LQR controller (1.5 seconds). A shorter settling time indicates that the hybrid controller can reach the desired velocity more quickly, which is essential for the robot's performance and flexibility.

iv. Steady-state error: Both the LQR and hybrid controller have zero steady-state error, which means that the velocity converges to the desired value without any steady-state error.

It could be said that the hybrid controller algorithm demonstrates superior performance compared to the LQRcontroller in terms of percentage overshoot, rise time, and settling time. These improvements are crucial for maintaining the desired velocity and ensuring the stability and flexibility of model mobile robot. The hybrid controller's ability to quickly respond to changes in velocity and reach the desired speed more efficiently give it an edge over the LQR and as such make it more suitable choice for this application.

6. Conclusion and Future Work

The two-wheel self-balancing is an interesting classical control problem whose development is rooted in theprinciple of inverted pendulum. It is a complex nonlinear system used as a test platform for testing the performance of control algorithms subject to disturbances. In this paper, a comparison was made between performances of LQR and hybrid model consisting of (LQR+PID) where the latter produced by far a better performance in both linear velocity and tilt angle control. Stable Tilt angle response for hybrid controller (LQR+PID) was achieved with a rise time of 2ms, settling time of 190ms and zero steady state error while an overshoot 13.was recorded in the hybrid response ove 40% in the the LQR response. In the same manner, stable control of linear velocity was achieved with t_r = 130ms, t_S = 1 s and e_{SS} = 0. However, a slightly lower percentage overshoot was recorded in hybrid mode (10.00%) as against (16.67%) in the LQR controller configuration. These performance metrics have fully satisfied the set specification (%O.S $\leq$ 20%, t_S = 2s and minimal steady-state error). The results of this study suggest that the hybrid controller (LQR+PID) is more effective in controlling the tilt angle and linear velocity of the two-wheel self-balancing robot compared to the LQR controller. The hybrid controller's superior performance can be attributed to the integrationof PID control, which provides better robustness and disturbance rejection capabilities. The study's findings have

significant implications for the development of control algorithms for two-wheel self-balancing robots, as they demonstrate the potential of hybrid controllers in achieving better performance and robustness.

In summary, this paper presents a comparison between the performance of LQR and hybrid controllers in controlling the tilt angle and linear velocity of a two-wheel self-balancing robot. The results show that the hybrid controller produced a better performance in both control variables, with stable responses and minimal steady-state error. The study's findings highlight the potential of hybrid controllers in achieving better performance and robustnessin controlling two-wheel self-balancing robots.

References

[1] Rubio F, Valero F, Llopis-Albert C. “A review of mobile robots: Concepts, methods, theoretical framework, and applications”. International Journal of Advanced Robotic Systems.16(2). 2019. doi:10.1177/1729881419839596

[2] Rabani M. R. M., Ibrahim A. M., Toha S. F. & Rashid M. “Two-wheel Balancing Robot; Review on Control Methods and Experiments”. International Journal of Recent Technology and Engineering. 7, 2019.

[3] Mohamed Gad O. M., Saleh S. Z. M., Bulbul M. A. and Khadraoui S., "Design and Control of Two Wheeled Self Balancing Robot (TWSBR)," Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 2022, pp.1-6, 2022. doi: 10.1109/ASET53988.2022.9735004.

[4] Katsuhiko Ogata. “Modern Control Engineering,” Pearson Education, Inc., publishing as Prentice Hall, One Lake Street, Upper Saddle River, New Jersey 07458. 2010.

[5] Zimit A. Y., Hwa. J. Y., Mukhtar F. H. and Indrazno S. “Modelling and Experimental Analysis Two-Wheel Self Balancing Robot Using PID Controller” Springer International Publishing AG, part of Springer Nature. 2018. Doi: https://doi.org/10.1007/978-3-319-95165- 2_48

[6] Iwendi C., Alqarni M. A., Anajemba J. H., Alfakeeh A. S., Z. Zhang, and A. K. Bashir, “Robust Navigational Control of a Two-Wheeled Self-balancing Robot in a Sensed Environment,” IEEE Access, vol. 7, 2019.

[7] Koyanagi E., Iida S., K. Kimoto and Yuta S. “A wheeled inverse pendulum type self-contained mobile robot and its two-dimensional trajectory control”, Proceedings of ISMCR’92, pp.891-898, 1992.

[8] Kamen D., “Segway PT”, 2001. Available online on: https://www.segway.com/ Accessed 26 March 2024.

[9] Matsumoto O., Kajita S. and Tani K. “Attitude estimation of the wheeled inverted pendulum using adaptive observer”, Proceedings of 9th Academic Conference of the Robotics Society of Japan, pp.909-910, 1991.

[10] Vinod K. P. and Kamala N. “Design, Dynamic Modelling and Control of Two-wheeled Self-Balancing Robot (TWSBR)” International Journal of Mechanical Engineering Vol. 7 (Special Issue 5, April-May 2022)

[11] Fabian R. J. L., Ilber A. R. R. and Andrés F. J. L. “Modelling and Control of a Two Wheeled Auto-Balancing Robot: A Didactic Platform for Control Engineering Education” 18th LACCEI International Multi-Conference for Engineering, Education, and Technology: July 27-31, 2020. DOI http://dx.doi.org/10.18687/LACCEI2020.1.1.556

[12] K Akhilesh R., Makani R., Jency M., Shriya V. S.1 and Harsha H. IoT Enable Self- Balancing Robot using LabView RV Journal of Science Technology Engineering Arts and Management vol 2 (1) (2021)

[13] Lwin M. M. T. and Ye C. Application of LQR Control for Two-Wheel Self-Balancing Robot J. Myanmar Acad. Arts Sci. 2020 Vol. XVIII.No.2C.

[14] Vicky M. & Jayant D. Self-Balancing Robot With Bluetooth Control *International Research Journal of Modernization in Engineering Technology and Science* Volume (04) Issue:06 June-2022

[15] Mathew A., Ananthu R., Binsy P., Vahid A. Thomas C. & Sidharthan, S. Design and control of a two-wheel self-balancing robot. *IOP Conference Series: Materials Science and Engineering*. 2021 DOI: https://dx.doi.org/10.1088/1757-899X/1114/1/012058

[16] Basant K., Riya C. & Indu B. Two-Wheeled Self-Balancing Robot *International Journal of Research in Engineering and Science (IJRES)*

2021.Volume 9, Issue 6, PP. 47-51

[17] Meheswara N. Venkata S. S. Greeshma S. G. & Ajith K.R. Design and Development of Self Balancing Robot", *International Journal of Research in Engineering, IT and Social Science*, ISSN 2250-0588, Impact Factor: 6.565, Volume 09, Special Issue 2, May 2019, Page 20- 32.

[18] Wang, C., Jianliang, X., & Zhang, C. Fuzzy Neural Network Active Disturbance Rejection Control for Two-Wheeled Self-Balanced Robot.*Journal of Information Processing Systems*, 18(4), 510-523. (2022). DOI: 10.3745/JIPS.01.0089

[19] Ryuichi T., Trong-Thuc H. & Cong-Kha P. "An Obstacle Avoidance Two-Wheeled Self-Balancing Robot" *International Journal of Mechanical Engineering and Robotics Research* Vol. 11, No. 1, January 2022

[20] Aldhalemi1 A. A., Chlaihawi1 A. A. and Al-Ghanimi1 A. Design and Implementation of a Remotely Controlled Two-Wheel Self-Balancing Robot *4th International Conference on Engineering Sciences (ICES 2020) 5th-6th December 2020, Kerbala, Iraq.* https://doi.org/10.1088/1757-899X/1067/1/012132

[21] Nguyen H. V. et al. A method of PD control for Balancing a Unicycle Robot. *Indonesian Journal of Engineering and Science*, Vol. 4, No. 1, 2023 Nguyen et al. https://doi.org/10.51630/ijes.v4i1.81

[22] Tandan, N., and Swarnkar, K. K. Tuning of PID Controller using Modified Particle Swarm Optimization, *International Journal of Electronics, Electrical and Computational System*, 2015. 4, 62-66.

[23] Alkhafaji, F., Wan Hasan, W., Sulaiman, N. & Mohd, M. A Novel PID Robotic for Speed Controller Using Optimization Based Tune Technique. *IntechOpen* 2021. DOI: https://doi.org/10.5772/intechopen.95892

[24] NUCLEO-F401RE development board datasheet https://www.st.com/en/evaluation-tools/nucleo-f401re.html Accessed 2nd February 2024.

[25] Kankhunthod, K, Kongratana, V, Numsomran, A, & Tipsuwanporn, V "Self-balancing Robot Control Using Fractional-Order PID Controller", *International Multi-Conference of Engineers and Computer Scientists*, vol. 7, no. 4, (2019).