

# **NORTHEASTERN UNIVERSITY**

## **EECE5554: Robotic Sensing and Navigation**

**Instructor:** Dr Thomas R. Consi

**Date:** 04/19/2024

### **Final Project Report**

**Title:**

**Simulation of Two-Wheeled Robot Stability: A Comparative  
Study of PID vs LQR Controller Systems**

**Group no:** 2

**Group members**

Kausalya Sankaranarayanan

Kaviarasu Annadurai

Rohit Gurusamy Anandakumar

Sharmila Sivalingam

***Abstract—*** This work performs a comparative analysis of the efficiency of Proportional-Integral-Derivative (PID) and Linear Quadratic Regulator (LQR) controllers in stabilizing inverted pendulums. The study investigates the utilization of linear acceleration and angular rate data obtained from an MPU as inputs to the controllers, which subsequently output torque to the motor. This torque determines both the speed and direction of the wheel, critical for maintaining balance in the inverted pendulum system. Through simulations conducted in MATLAB, various scenarios are examined, including the effect of different loads on the system's stability and performance. The results reveal insights into the effectiveness of PID and LQR controllers in different conditions, shedding light on their respective strengths and limitations in stabilizing inverted pendulums.

## I. BACKGROUND

### *A. Inverted Pendulum:*

Balancing an inverted pendulum is crucial for stability, control, and efficiency in systems like self-balancing robots<sup>[1]</sup>. Stability prevents toppling, ensuring consistent performance. Advanced control systems use sensor data to detect deviations and apply precise corrections, optimizing energy use. Efficient balance extends battery life, enhancing functionality and safety by reducing collision risks. Balanced systems minimize accidents, enhancing operational safety and reliability. Managing balance in inverted pendulum systems is essential for stable, controlled, and safe operation in dynamic environments<sup>[2]</sup>, where precise control and efficient resource use are critical for success.

### *B. Controller logic:*

In self-balancing robots, PID controllers offer simplicity and effectiveness in maintaining stability by adjusting motor speeds in response to deviations from the upright position<sup>[3]</sup>. However, PID control may not achieve optimal performance due to its limitations in handling complex dynamics. LQR controllers, on the other hand, excel in optimizing control actions for Multiple Input, Multiple Output (MIMO) systems like self-balancing robots, offering superior stability and performance<sup>[4]</sup>. Using an MPU (Microprocessor Unit) over light sensors enhances control precision and responsiveness. MPU provides real-time data on the robot's orientation, enabling more accurate adjustments to maintain balance compared to light sensors, which may be less reliable in varying lighting conditions. Utilizing these controller logics is crucial for achieving stable and precise control in self-balancing robots, ensuring safe and efficient operation in dynamic environments.

### *C. Real-life products:*

Self-balancing robots are prominent in real-life applications like Segway personal transporters, enhancing urban mobility. They're also utilized in warehouse logistics for automated material handling, increasing efficiency. Additionally, they find use in educational platforms and research labs, fostering innovation in robotics and control systems.

## II. METHODS

### A. Nomenclature

Symbol	Parameter	Value[unit]
$x_r$	Straight line position	[m]
$\theta_p$	Pitch angle	[rad]
$M_r$	Mass of rotating masses connected to the left and right wheel.	0.4[kg]
	$M_{RL} = M_{RR} = M_r$	
$M_p$	Mass of body	2.1[kg]
$R$	Radius of wheel	0.05[m]
$L$	Distance between the z axis and the center of gravity of the vehicle	0.3[m]
$g$	Gravity constant	9.8[ms <sup>-2</sup> ]
$C_L, C_R$	Input torque for right and left wheels accordingly	0.147[Nm]

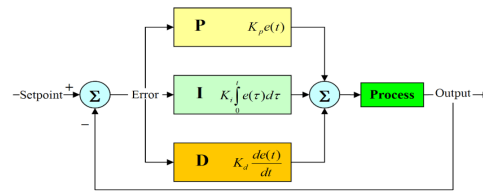
**Fig.1** Symbols and Parameters Used

### B. Sensor Readings

In Simulink, our model incorporates Multibody modeling to simulate the dynamic behaviors of mechanical systems comprising rigid and/or flexible bodies connected by joints. Prismatic joints enable movement along the Z-axis, while Revolute joints facilitate rotation. This approach allows for a highly realistic representation of real-world conditions, including gravity and adjustable friction damping<sup>[5]</sup>. To capture the system's dynamics, we utilize Transfer sensors between the joints. Specifically, we employ a Transfer sensor as a 3-axis Accelerometer to measure linear acceleration and another as a 3-axis Gyroscope to measure angular rate, providing essential sensor values for our simulation.

### C. PID

The Proportional-Integral-Derivative (PID) control loop is a continuous modulated control system<sup>[6]</sup>. As shown in Fig 3, it calculates the error  $e(t)$  as the difference between the set point ( $SP = r(t)$ ) and the measured process variable ( $PV = y(t)$ ), then adjusts the output  $u(t)$  based on PID coefficients ( $K_p, K_i, K_d$ ) to minimize  $e(t)$ . By tuning these coefficients, the controller's stability, steady-state error, and oscillations can be adjusted according to specific requirements. Increasing the proportional term (P) in a PID controller can improve stability by reducing the settling time and overshoot. Increasing the integral term (I) reduces steady-state error and helps the system reach the setpoint more accurately. Increasing the derivative term (D) reduces oscillations and improves the system's response to changes. Here input for the loop is angle(rad) from MPU, the output from the loop is actuator torque which is given to the model motors.



**Fig.2** Symbols and Parameters Used

### D. State-Space Modelling

Dynamic systems can be represented as differential equations<sup>[7,8]</sup>. From below Equations (1,2,3,4,5) internal states are connected through matrix A, which defines how the states change over time. Matrix B describes how external inputs affect these states. Matrix C determines how the states are combined to produce the system's output. Matrix D allows inputs to bypass the system and directly influence the output, providing a feedforward path. This representation in state space equations is crucial for understanding and analyzing the behavior of dynamic systems like the inverted pendulum.

$$\begin{bmatrix} \dot{x}_r \\ \dot{x}_r \\ \dot{\theta}_p \\ \ddot{\theta}_p \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & A_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & A_{43} & 0 \end{bmatrix} \begin{bmatrix} x_r \\ \dot{x}_r \\ \theta_p \\ \dot{\theta}_p \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \\ 0 \\ B_4 \end{bmatrix} [C_L \quad C_R] u \quad y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \Phi \\ \dot{\Phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (1,2)$$

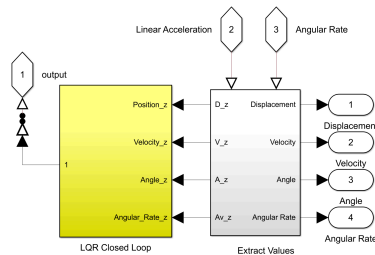
$$X = \frac{1}{3} \frac{M_p(M_p + 6M_r)L}{M_p + \frac{3}{2}M_r}, \quad Y = \frac{M_p}{(M_p + \frac{3}{2}M_r)R} + \frac{1}{L}, \quad A_{23} = g(1 - \frac{4}{3}L \frac{M_p}{X}), \quad A_{43} = \frac{gM_p}{X} \quad B_2 = \left( \frac{4LY}{3X} - \frac{1}{M_p L} \right), \quad B_4 = -\frac{Y}{X} \quad (3,4,5)$$

### E. LQR

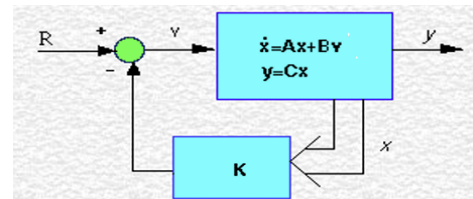
The LQR (Linear Quadratic Regulator) controller computes optimal K values using Q and R parameters<sup>[9]</sup>. Q = C\*C (reduce for aggressive control) and R = 1 (reduce for slower response). Optimal values are found through trial and error. Linear acceleration and angular rate from MPU are transformed into displacement, velocity, angle, and angular rate, then sent to the LQR block for K multiplication. Gain K is calculated separately using the MATLAB function K = lqr(A, B, Q, R) with the state space model and system parameters. The eigenvalues of Matrix A determine system poles, which are initially positive indicating instability. After controller application, marginal poles suggest marginal stability<sup>[10]</sup>.

**Table.1 System Parameters**

System Parameters		
Mass of the Chassis	M	1 kg
Mass of Wheels	m	0.2 kg
Friction coefficient	b	0.1
Inertia	I	0.0005
Gravity acceleration	g	9.8
Length of COM	L	0.125



(a) LQR model from Simulink

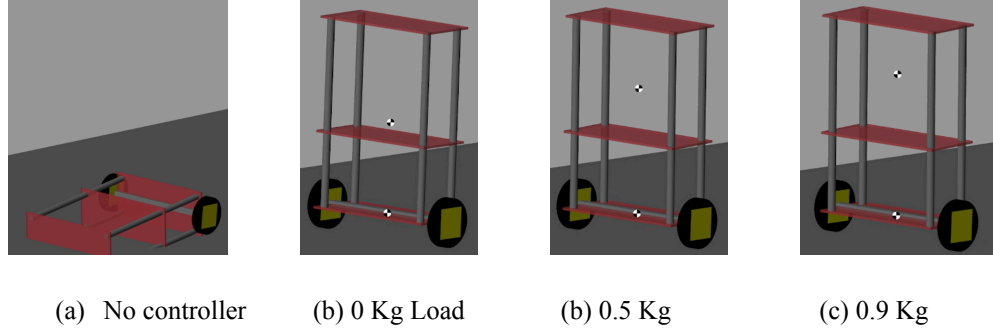


(b) LQR controller loop

**Fig.3 LQR block**

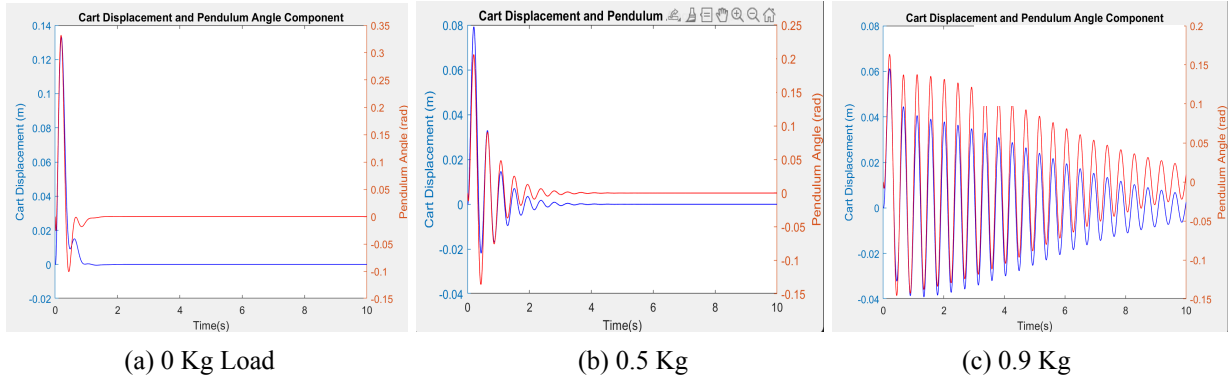
### III. RESULTS

The performance of the mobile inverted pendulum robot under different weights on top +0,+0.5kg,+0.9kg as the center of gravity shifts up as seen in the fig 4(b,c,d), and controller types were evaluated through simulations in MATLAB.



**Fig4** Cart with no controller (a) and cart with weights (b,c,d) indicating the center of gravity

#### A. LQR Controller Performance



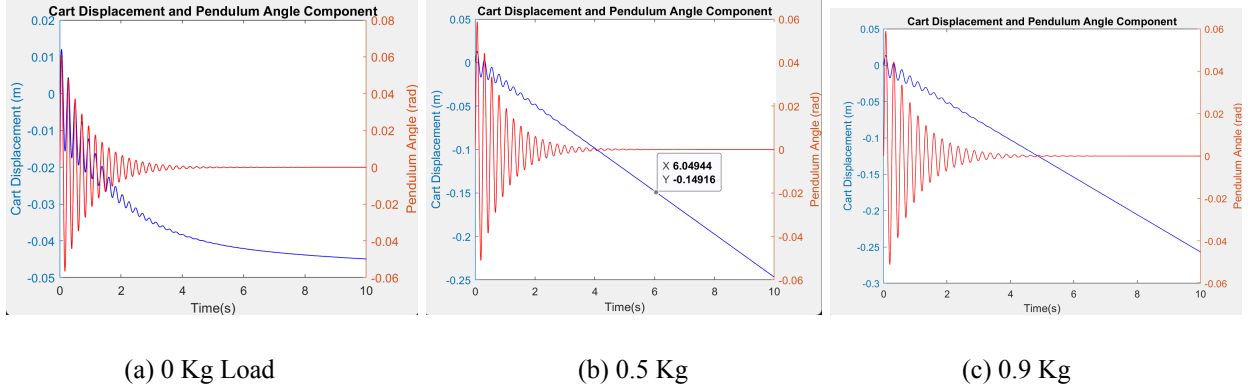
**Fig.5** The LQR controller's Cart displacement and pendulum angle component graph

Figure 5(a) depicts the robot's performance without load using an LQR controller. The pendulum angle (orange curve) and the cart's position (blue curve) indicate rapid stabilization. The robot achieves balance within a second, demonstrating minimal oscillation. The overshoot, rise time, and settling time are all within the desired specifications. In Figure 5(b), the robot is loaded with 0.5 kg. The stabilization process under this condition shows fewer oscillations but a delayed response compared to the no-load scenario. The shift in the center of gravity affects the robot's dynamics, leading to a steady-state error and deviations in rise and settle times from the no-load condition. Figure 5(c) illustrates the scenario with a 0.9 kg load. Here, the robot struggles to stabilize, exhibiting significant oscillations that decrease only gradually over time. This plot indicates unsatisfactory performance, with necessary improvements in settling and rise times for better stability.

#### B. PID Controller Performance

Figure 6(a) shows the robot's response to no load using a PID controller. The stabilization

process takes approximately 5 seconds, with the oscillation curve flattening over time. The metrics of overshoot, rise, and settling times are maintained consistently across tests. Figure 6(b) presents results under a 0.5 kg load with the PID controller. Similar to the LQR scenario, the added mass induces a shift in the center of gravity, influencing the stabilization time and oscillation behavior, albeit with fewer oscillations compared to the LQR controller under similar conditions. Finally, Figure 6(c) examines the robot loaded with 0.9 kg under PID control. The stabilization is delayed, with noticeable oscillations that diminish slowly. The shift in the center of gravity from the no-load and 0.5 kg conditions is evident, affecting overall stability.



**Fig.6** The PID controller's Cart displacement and pendulum angle component graph

### C. Comparative Analysis and Material Considerations

The comparative analysis of LQR and PID controllers under varying load conditions shows that heavier loads significantly affect the robot's stabilization time, leading to increased instability. This insight points out how important it is to choose strong and lightweight materials for the body of the robot to minimize its impact on stability. Furthermore, optimizing the robot's height relative to its center of mass is also very important. While a taller robot would be simpler to balance, a lower center of mass indicates that the wheels can stay below it and keep the robot balanced.

## IV. CONCLUSION

In conclusion, the work aims to emphasize the importance of selecting appropriate control algorithms and sensor inputs for stabilizing inverted pendulum systems. The simulations demonstrate that heavier loads pose challenges to the system's stability, resulting in longer settling times and increased oscillations. Along with this, the choice of materials for the pendulum is crucial, with lightweight yet strong materials being preferred to optimize stability and minimize settling time. Furthermore, the design of the robot's height and center of mass plays a significant role in achieving balance, necessitating careful consideration during the design phase. Overall, the findings provide valuable insights for designing and optimizing mobile inverted pendulum robots, highlighting the importance of control algorithm selection and system design considerations for achieving stable and efficient operation.

## V. REFERENCE

- [1] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proceedings of the Institution of Electrical Engineers*, vol. 121, no. 12, pp. 1585-1588, Dec. 1974.
- [2] K. J. Astrom and T. Hagglund, "The future of PID control," *Control Engineering Practice*, vol. 9, no. 11, pp. 1163-1175, Nov. 2001.
- [3] B. C. Kuo and F. Golnaraghi, *Automatic Control Systems*, 9th ed. Hoboken, NJ, USA: Wiley, 2010.
- [4] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed. London, UK: Springer, 2009.
- [5] K. J. Astrom and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1997.
- [6] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.
- [7] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157-173, Mar. 2004.
- [8] J. Wang, S. M. Savaresi, and J. E. Riberio, "The control of linear inverted pendulum system via sliding mode and LQR techniques," in *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, USA, Dec. 2002, pp. 2856-2861.
- [9] R. A. Valenzuela and L. R. Garcia, "Stabilization of an inverted pendulum using LQR control," in *Proceedings of the American Control Conference*, San Francisco, CA, USA, Jun. 2011, pp. 3200-3205.
- [10] P. W. Gibbens and Y. Zhang, "PID controller tuning for desired closed-loop response using root locus," in *Proceedings of the IEEE International Conference on Control Applications*, Anchorage, AK, USA, Sep. 2000, pp. 123-128.