Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 2, No.1, February : 2025

IMPACT OF ROTATING ARM ON BALANCING ROBOT

Dr. Prabin Kumar Jha, Associate Professor, Department of Robotics and Automation Engineering, The Neotia University, Kolkata, India Email: prabinkumar.jha@tnu.in

ABSTRACT

The huge application of balancing robot needs its better dynamic steady state performances. The paper presents the design of a balancing robot in Coppeliasim environment. Two control strategies PID control and a hybrid PID and LQR (Liner Quadratic Regulator) control are incorporated through API Python in Coppeliasim. Dynamic behavior is studied based on two controllers. The control algorithms are designed when the disturbances are addressed to the robot to keep it in the vertical equilibrium. The novelty of work is to study the joint torque on the wheels due to the impact of rotating arm. The simulation is carried out on two different surfaces of coefficient of frictions 0.2 and 0.8 respectively. Proper values of parameters for PID and LQR are determined.

Keywords:
Balancing robot; PID; LQR; Joint torque; Coppeliasim

I. **Introduction**

The two-wheeled balancing robot has become an interesting topic due to its unstable dynamic performance and non-linearity in the evaluation of various control theories. Many works have been studied on the balancing robot but no unique solution found.

The control strategy in balancing robot is to stabilize its position, velocity, angular velocity and tilt angle. The aim is to approach robot's state variables in the shortest time. An accelerometer, speed encoders, and a gyroscope are integrated to robot to measure its state variables [1].

Linear and nonlinear control techniques have been implemented to study the performance of balancing robot [2]. Further, the study includes sliding mode controls, fuzzy logic control, artificial neural network, and deep learning [3]. The PID and LQR schemes are the most popular methods to control the robot stability [4]. The performance of PID and LQR schemes have been compared and analyzed. A hybrid PID-LQR scheme is then employed to improve the performance of robot. The experiment is performed on the balancing robot with control to understand its performance [5]. The real model of balancing robot is made to evaluate its stability [6]. Pole placement technique proves the best performance as compared to the LQR technique [7].

A comparative study between Model-Based Control (MBC) and Data-Based Control (DBC) strategies, are implemented for stabilizing balancing robot [8].

A kinematic model of the robot with a PI controller is simulated and further an avalanched of two PIDs is compared [9].

A robot is designed to minimize its movement with a control algorithm of gyroscope module and disturbance observer [10]. The experiment on the self-balancing robot is
conducted with PID controller and its dynamics and control robustness are evaluated [11]. Simulations and experimental results of the Perceptual Control algorithm is compared to the LQR controller [12]. Just-in-Time learning based is offered to optimize the dynamic performance under disturbances [13]. Nonlinear dynamics and stability of two wheeled auto balancing is experimented and compared with the simulation results [14]. The PID controller and Kalman filter are implemented in the balancing robot to reduce the noises [15]. Adaptive tracking control with periodic disturbances and uncertainties for a mobile-wheeled inverted pendulum is formulated [16]. A cascade control algorithm and the appropriate reference trajectories are incorporated in the system to maintain its stability [17]. A sliding mode control is proposed to optimize the error of balancing robot [18]. An adaptive trajectory tracking controller is proposed to study the stability of robot [19]. Nonlinear disturbance observer is designed to enhance the tracking performance of robot [20]. An intelligent control consists of online learning

and neural network of two wheel-based mobile inverted pendulum system is tested [21]. Decoupled multi-loop control of linear and rotational motions of the robot is analyzed [22]. Strategies of Mamdani-Like Fuzzy and Fusion-Based fuzzy for balancing robot is compared and it is found that PD-Like fuzzy has better performance [23]. Robust adaptive observer is proposed to the inverted pendulum with variable mass [24]. Integral sliding-mode controller on a two-wheeled mobile robot is addressed to carry out the uncertainty of its dynamics [25]. Two sliding mode controllers are implemented to control velocity and braking of the mobile wheeled inverted pendulum [26].

The simulation of self-balancing robot on various terrains are studied [27]. Dynamic performance of balancing robot on two different surfaces is analyzed [28]. A prototype of wheeled based inverted pendulum with PID to test on surfaces with different friction coefficients [29]. Disturbance of uncertain terrains is compensated to two wheels is simulated and experimented [30].

Two algorithms (basic Q-Learning and Deep Q-Networks (DQN)) are incorporated to the inverted pendulum to control with its high accuracy [31]. The performances of Advantage Actor-Critic and Deep Q-Network algorithm are tested on the inverted pendulum and compared [32]. Deep Reinforcement Learning (RL) controller is applied to the cart-pole and its sensitivity analysis is studied [33]. A comparative study of model-free RL and model based Nonlinear Model Predictive Control (NMPC) on the cart pendulum is performed [34]. The self-balancing robot is designed with Q learning and deep Q network (DQN) [35].

This paper presents the simulation of balancing robot in Coppeliasim software and the impact of a rotating arm on it is determined. The robot moves on two different surfaces is considered for the study. Section III describes the design of controllers to keep the balancing robot in its vertical position.

II. **Method**

The method section describes the mathematics of balancing robot. The dynamics of robot covers robot chassis, and the forces on the wheels. The balancing robot is designed in Coppeliasim with proper specification. The velocity to each wheel of robot is considered. Low friction and high friction surfaces are addressed in the modeling. PID is added to make the balancing robot controllable. Then PID and LQR controller are designed in the model. The LQR control determines the optimal gains to meet desired cost specifications. Control algorithms of the research methodology is expressed in detail.

A. Dynamic Model of Balancing Robot

The governing equations of balancing robot to define its behavior is expressed applying the Lagrangian dynamics based as shown in Figure 1 [11, 14]. The kinetic energy of the system due to its wheel's angular motions can be represented as:

$$K.E_r = \frac{1}{2} m_w \dot{x}^2 \qquad (1)$$

Let xj and yj coordinates is the position of the center of gravity of the robot:

$$\begin{aligned} x_j &= x + l.\sin\emptyset_r \\ y_j &= l.\cos\emptyset_r \end{aligned} \qquad (2)$$

Taking the derivatives of the equation (2) with respect to time, the respective velocities were found as:

$$\begin{aligned} v_{x_j} &= \dot{x} \dot{+} l.\cos\emptyset_r \\ v_{y_j} &= -l.\sin\emptyset_r \end{aligned} \qquad (3)$$

Considering the squares of the balancing robot velocities:

$$\left|v_j^2\right| = v_{x_j}^2 + v_{y_j}^2 = \dot{x}^2 + 2l\dot{x}\dot{\emptyset}_r\cos\emptyset_r + l^2\dot{\emptyset}_r^2 cos^2\emptyset_r + l^2\dot{\emptyset}_r^2 sin^2\emptyset_r \quad (4)$$

Rearranging the equation (4):

$$\left|v_j^2\right| = \dot{x}^2 + 2l\dot{x}\dot{\emptyset}_r\cos\emptyset_r + l^2\dot{\emptyset}_r^2 \quad (5)$$

Then the kinetic energy of the robot due to its linear displacement is obtained as:

$$K.E = \frac{1}{2}m_r\dot{x}^2 + m_r l\dot{x}\dot{\emptyset}_r\cos\emptyset_r + \frac{1}{2}m_r l^2\dot{\emptyset}_r^2 \quad (6)$$

Adding equations (2) and (6) to get Total Kinetic Energy (TE) of balancing robot, we have

$$T_E = \frac{1}{2}(m_r + m_w)\dot{x}^2 + m_r l\dot{x}\dot{\emptyset}_r\cos\emptyset_r + \frac{1}{2}(J_r + m_r l^2)\dot{\emptyset}_r^2 \quad (7)$$

Applying Lagrange equation for the velocity of the rotation system and the main body of the balancing robot, we get:

$$\frac{d}{dt}\left(\frac{\partial T_E}{\partial \dot{x}}\right) - \frac{\partial T_E}{\partial x} = Q_w \quad (8)$$

$$\frac{d}{dt}\left(\frac{\partial T_E}{\partial \dot{\emptyset}}\right) - \frac{\partial T_E}{\partial \emptyset} = Q_r \quad (9)$$

Qw and Qr are derived:

$$\frac{\partial T_E}{\partial \dot{x}} = (m_r + m_w)\dot{x} + m_r l\dot{\emptyset}_r\cos\emptyset_r \quad (10)$$

$$\frac{d}{dt}\left(\frac{\partial T_E}{\partial \dot{x}}\right) = (m_r + m_w)\ddot{x} + m_r l\ddot{\emptyset}_r\cos\emptyset_r \quad (11)$$

$$\frac{\partial T_E}{\partial x} = 0 \rightarrow Q_w = T_w - f\dot{x} \quad (12)$$

$$\frac{\partial T_E}{\partial \dot{\emptyset}_r} = m_r l\dot{x}\cos\emptyset_r + (J_r + m_r l^2)\dot{\emptyset}_r \quad (13)$$

$$\frac{d}{dt}\left(\frac{\partial T_E}{\partial \dot{\emptyset}_r}\right) = m_r l\ddot{x}\cos\emptyset + (J_r + m_r l^2)\ddot{\emptyset}_r - m_r l\dot{x}\dot{\emptyset}_r^{\,2}\sin\emptyset_r \quad (14)$$

$$\frac{\partial T_E}{\partial \emptyset_r} = -m_r g l\sin\emptyset_r - f\dot{\emptyset}_r \quad (15)$$

Substituting the values of equations (10) to (15) in the Lagrange equations (8) and (9) is obtained:

$$\ddot{x} = -\frac{m_r\, l}{(m_r + m_w)}\left(\ddot{\emptyset}_r\, cos\emptyset_r + \dot{\emptyset}_r^{\,2}\ \sin\emptyset_r\right) + \frac{1}{(m_r + m_w)}(T - f.\dot{x}) \tag{16}$$

$$\ddot{\emptyset}_r = -\frac{m_r l}{(J_r + m_r l^2)}(\ddot{x}\cos\emptyset_r + g\sin\emptyset_r) - \frac{f}{(J_r + m_r l^2)}\dot{\emptyset}_r \tag{17}$$

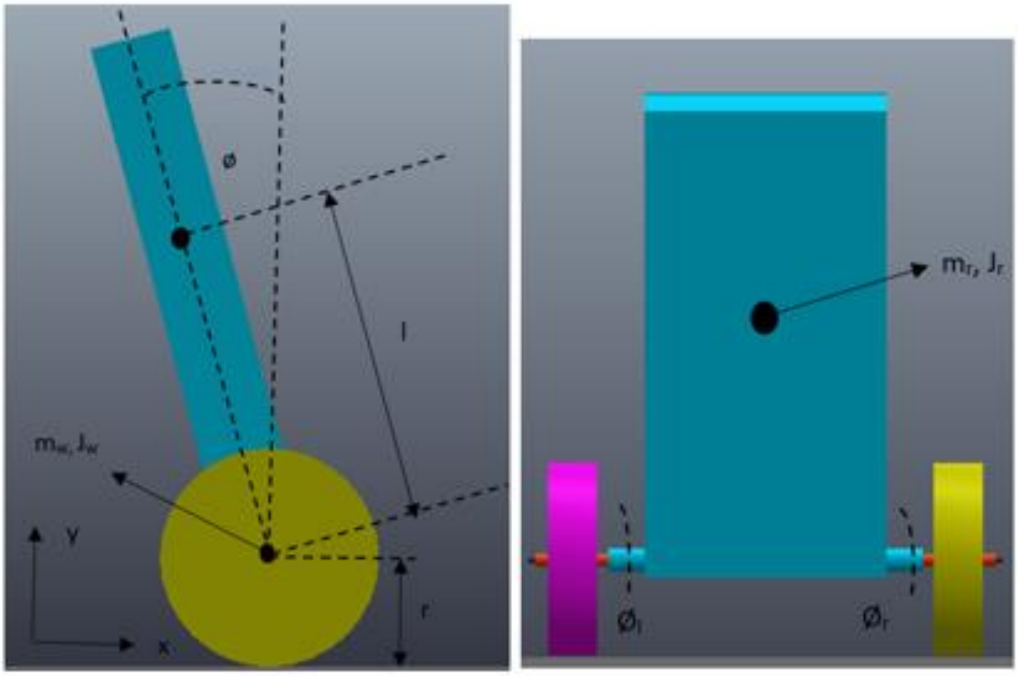

Fig. 1 Free Body Diagram of Balancing Robot in Coppeliasim.

TABLE I. Comparative Study on Methods of Self Balancing Robot

S. No.	***Principle***	***Software***	***Parameter***	***Result***	***Remarks***	***Ref[]***
1.	PID and LQR	Matlab/Simulink	Mathematical expression	Application for step, ramp and uneven terrain	Better performance achieved.	[27]
2.	PID	Matlab/Simulink	Mathematical expression and experiment	Low friction and no slip	Low-traction terrain smoother	[28]
3.	PID and LQR	Matlab/Simulink	Mathematical expression and experiment	Different coefficient of frictions	Experimental tested with good stability	[29]

4.	PID	Simscape Multibody Toolbox	Uncertain terrains and experiment	environmental disturbances is addressed	Greater functionality	[30]

Equations (16) and (17) define the non-linear dynamic behavior of the balancing robot. Equation (16) was obtained from the complete rotation system, while (17) was obtained from the main body. The torque (T) as an input is applied to the left and right wheel of robot by the DC motors in the simulation.

B. Representation of Linearized Balancing Robot Model in the State Space

The equations (16) and (17) are linearized around the equilibrium point of the system at tilt angle ø = 0, the desired operating position is considered when the system is in a stable vertical condition. The linearized model is made with the assumption sinø ≅ ø; cosø ≅ –1; and neglecting higher order of ø ≅ 0.

$$\ddot{x} = \frac{m_r\,l}{(m_r + m_w)}\ddot{\emptyset}_r + \frac{1}{(m_r + m_w)}(T - f.\dot{x}) \qquad (18)$$

$$\ddot{\emptyset}_r = \frac{m_r l\ddot{x}}{(J_r + m_r l^2)} - \frac{f}{(J_r + m_r l^2)}\dot{\emptyset}_r \qquad (19)$$

Substituting from (19) in (18), we have:

$$\ddot{x} = \frac{(T - f.\dot{x})(J_r + m_r l^2) + m_r^2 l^2 g \emptyset_r}{J_r(m_r + m_w) + m_w m_r\, l^2} \qquad (20)$$

Substituting from (19) in (18), we have:

$$\ddot{\emptyset}_r = \frac{m_r l(T - f.\dot{x})(J_r + m_r l^2) + m_r g(m_r +}{J_r(m_r + m_w) + m_w m_r\, l^2} \qquad (21)$$

$$Suppose\ h = J_r(m_r + m_w) + m_w m_r\, l^2$$

Equations (20) and (21) can be formed in State Space Equation. The state variables of the balancing robot are:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{f(J_r + m_r l^2)}{h} & \frac{m_r^2 l^2 g}{h} \\ 0 & 0 & 1 \\ 0 & -\frac{f m_r l}{h} & \frac{m_r g(m_r + m_l)}{h} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{J_r + m_r l^2}{p} \\ 0 \\ \frac{m_r l}{p} \end{bmatrix} [T] \qquad (22)$$

The output C and feedforward D matrices:

$$\boldsymbol{Cx} + \boldsymbol{Du} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \emptyset_r \\ \dot{\emptyset}_r \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} [T] \qquad (23)$$

III. Simulation Set for Modeling of Balancing Robot

A proper rigid body of balancing robot is built in Coppeliasim with its version 4.7.0. It is free software for academic uses. The software offers remote API interface with Python software. The ODE physics engine is considered for the simulation of robot dynamics with 0.001s time step.
The robot is developed in the scene of Coppeliasim as shown in Figure 2.

Fig. 2 Simulation of Balancing Robot in Coppeliasim.

The titling angle of robot is 12.5 deg. A rotating arm of mass 2 kg and has angular velocity of 180 deg/s. The robot strikes the rotating arm in approximate 8s.

The simulation of robot for two different surfaces having coefficient of frictions (0.2 and 0.8) are performed. Its effects on the joints of robot is studied. The PID controller and PID+LQR are included in the model to make the movement of the robot with tilt angle 0.
The parameters are taken in the design of the model described in Table II.

TABLE II. Physical Parameters for the robot

Parameter		***Value***	***Unit***
Mass of balancing robot	m_r	2	kg
Mass of wheel	m_w	0.2	kg
Moment of inertia of the robot	J_r	0.005	kgm^2
Wheel Radius	r	0.05	m
Moment of inertia of the wheel	J_w	0.002148	Kgm^2
Friction Coefficient of the surface	f	0.2 & 0.8	$Nrad^{-1}s^{-1}$
Length measured from the wheel axis and mass center of the robot.	l	0.1	m
Gravitational acceleration of the Earth	g	9.81	m/s^2

A. PID Controller Design of Balancing Robot

The gains of PID controller has: the proportional gain (Kp), the differential gain (Kd) and the integral gain (Ki) are designed to the system [14]. The state of the actual tilt angle measured by the gyro sensor of the system in Coppeliasim. The error is formed due to the difference between the desired state of the tilt angle ø = 0. The PID is integrated to system to minimize the error. The Kp reduces the angular position error of the system, advoiding the falling of robot. The integral gain Ki minimizes the steady-state error. The differential gain Kd acts on the system to compensate the overshoots in the transient response. The eigen values of the system are [-20+0.2j, -30.0+0.5j, -10.48+1.6j, -10.48-1.6j] who shows the system stable. The schematic diagram of PID in the system is shown in Figure 3 and respective gains are presented in Table III.

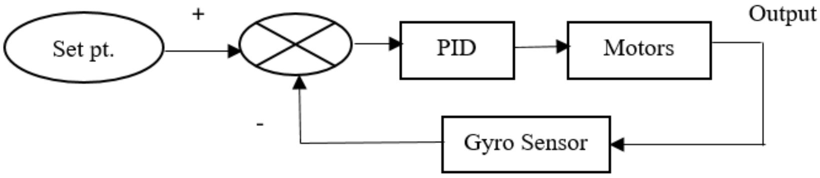

Fig. 3 PID Controller Design in the system.

TABLE III. PID Gain Values

Gain Values		
Proportional Gain (kp)	***Integral Gain (ki)***	***Derivative Gain (kd)***
10	0.02	2

B. Combination of PID Controller and LQR in Balancing Robot

The LQR (Linear Quadratic Regulator) technique is proposed to the system. The LQR determines the optimal gains which multiply each state variable of balancing robot. The gain matrix K= [0.79, 0.98, 9.66, 1.15] are made by minimizing a cost function. The controller finds the rotational position of system to make the tilt angle from the vertical position to zero (ø = 0).

The updated PID parameters while integrating with LQR is presented in Table IV.

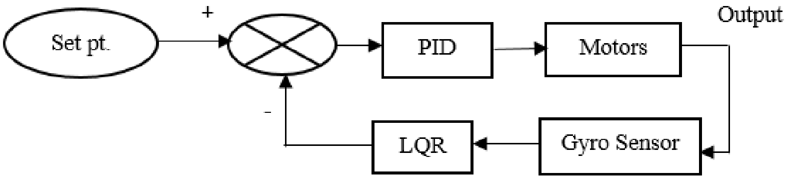

Fig. 4 PID Controller plus LQR Design in the system.

The values of Q and R are assigned in the system as:

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{R} = [0.1]$$

Using **lqr** command in Python to get the best fit of these values and above mentioned gain matrix K is evaluated.

TABLE IV. Updated PID Gain Values

Gain Values		
Proportional Gain (kp)	***Integral Gain (ki)***	***Derivative Gain (kd)***
6	0.2	0.1

The movement of balancing robot center of mass after striking the rotating arm is shown in Figure 5.

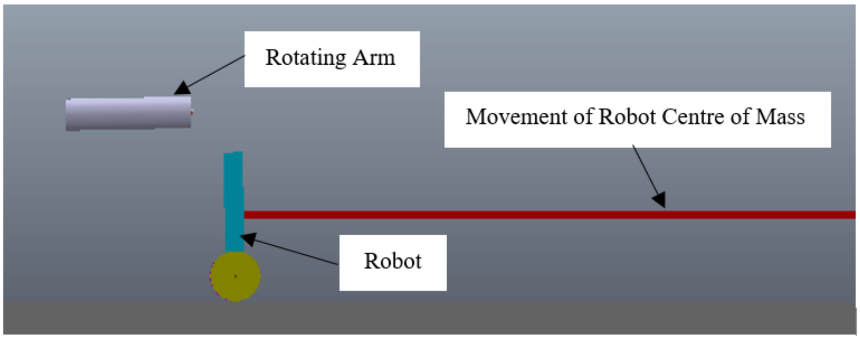

Fig. 5 Trajectory of robot center mass after striking the rotating arm.

IV. Simulation and Discussion

The proposed work covers the design of PID algorithm and combination of PID and LQR algorithm for two wheeled balancing robot. The simulation environment of robot is successfully structured in Coppeliasim. The complete control algorithms are developed in Python API. The behavior of robot has studied at a satisfactory level and the results of the simulation are described. Simulations with different PID gains and LQR parameters are iterated to achieve the suitable results. The response of joint torques at each wheel on two different surfaces are analyzed.

The balancing robot system with PID controller produced responses for tilt angle ø and joint torques on surfaces having coefficients 0.2 and 0.8. The gyro sensor is adopted to calculate the tilt angle. The Kp, Ki, and Kd values are obtained by trial and error on the behavior of the self-balancing robot. The PID helps the robot to balance its position without falling to move. The tilt angle of the robot also remained near 2 deg.

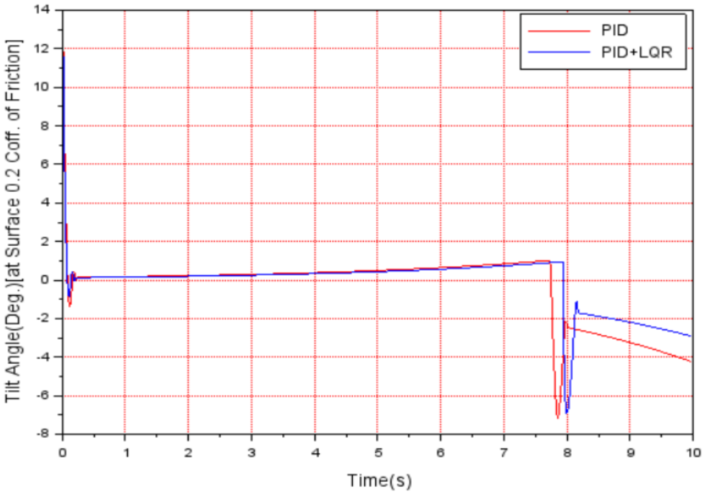

Fig. 6 PID and Hybrid PID-LQR control on the surface 0.2.

As mentioned earlier, the initial value of the title angle ø of the balancing robot was 12.5 degree.

A low pass filter is applied to the gyro sensor of robot to reduce the noise of high frequency vibration produced by the motors of the robot.

The performance of balancing robot on the surface of coefficient of friction 0.2 is shown in Figure 6. A significant improvement of robot with both control schemes is observed.

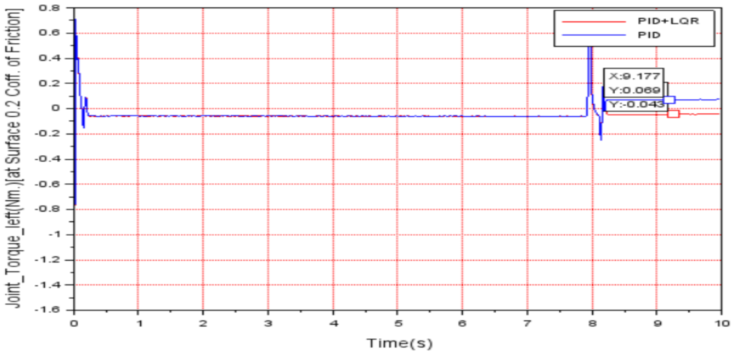

Fig. 7 Joint Torque of Left Wheel(Nm) on the surface 0.2.

A suitable value of combined LQR and PID are set to the model and smooth responses are obtained as compared to PID controller.

The joint torque (Nm) on left wheel are shown in Figure 7. The torque is increased while striking the arm at 8s. It becomes stable after the strike. The spikes in both control algorithms are shown due to impact of rotating arm.

The robot is able to withstand shock that was produced when it bumped with the arm. It oscillates more for a very few seconds and later return to its initial movement. Similar performances are observed in joints.

It is noticed that the dynamics of robot is changed during the strike of rotating arm in 8s. The tilting angle of the robot is increased.

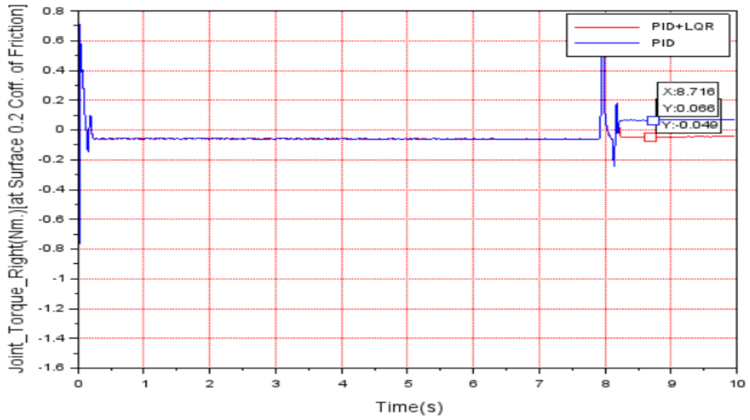

Fig. 8 Joint Torque of Left Wheel(Nm) on the surface 0.2.

The joint torque (Nm) on right wheel is shown in Figure 8. Similar behavior is noticed as compared to Figure 7.

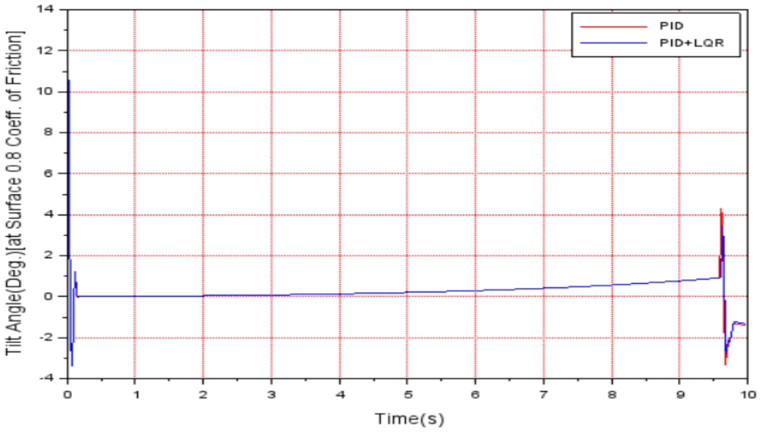

Fig. 9 PID and Hybrid PID-LQR control on the surface 0.8.

In both wheels, the joint torque reached at its maximum value (0.7Nm) while colliding with the rotating arm.

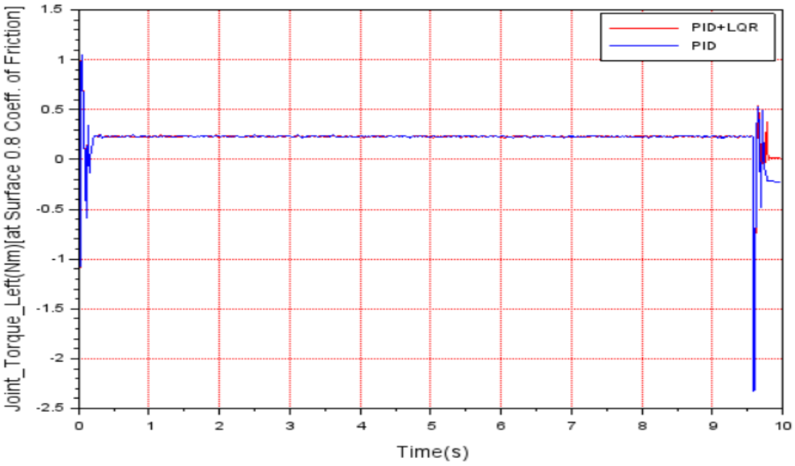

Fig. 10 Joint Torque of Left Wheel(Nm) on the surface 0.8.

The simulation study of robot on surface of coefficient of friction 0.8 is analyzed. The tilting angle is varying below 1.5 deg as shown in Figure 9.

The joint torque (Nm) on both wheels during the movement on the surface of friction coefficient 0.8 are shown in Figures 10 and 11 respectively. These are more on the hard surface in the strike for a small time. Immediately after the collision, the robot becomes stable.

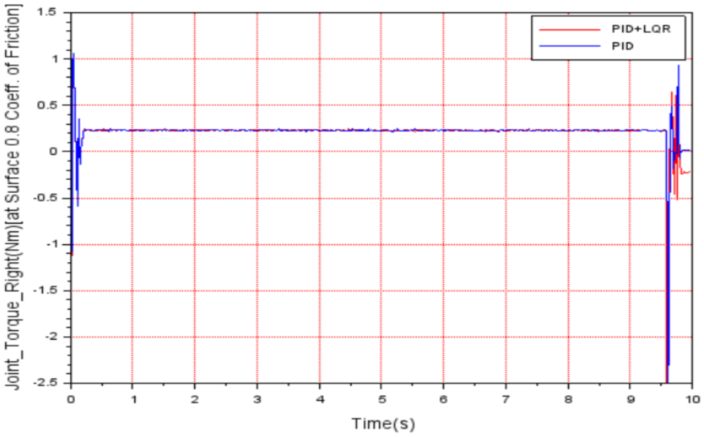

Fig. 11 Joint Torque of Right Wheel(Nm) on the surface 0.8.

V. Conclusion

This study presents the impact of rotating arm on the wheel joints of balancing robot. A comparative literature on balancing robot is expressed. The model is developed in Coppeliasim. Further simulation is incorporated with Python API where control algorithms is made. Two different surfaces of coefficient of frictions (0.2 and 0.8) are considered in the simulation. The tilt angle of balancing robot is studied successfully in association with control system of PID and hybrid of PID and LQR.

A rotating arm affects a balancing robot's dynamics and design requirements. It introduces challenges such as increased control complexity and stability management but also offers opportunities for enhanced utility and functionality.

In future, Reinforcement Learning and PID controller will be incorporated in the model that can handle uncertainties and nonlinearities effectively. An experiment will be performed extending to control speed or trajectory of robot.

This study will help to understand the impact of dynamic object on the balancing robot during the collision.

Acknowledgment

No fund is provided to pursue this research work.

References

[1]Lau, Lu & Ahmad, Nur & Goh, Patrick. “Self-balancing robot: modeling and comparative analysis between PID and linear quadratic regulator,” International Journal of Reconfigurable and Embedded Systems (IJRES), 2023, Vol.12, pp351-359.

[2]A. Loganathan and N. S. Ahmad, “A systematic review on recent advances in autonomous mobile robot navigation,” Engineering Science and Technology, an International Journal,2023, vol. 40, p. 101343, doi: 10.1016/j.jestch.2023.101343.

[3]R. Alika, El Mehdi Mellouli, El Houssaine Tissir,
“A modified sliding mode controller based on fuzzy logic to control the longitudinal dynamics of the autonomous vehicle”, Results in Engineering, 2024, Vol. 22, 102120, ISSN 2590-1230, https://doi.org/10.1016/j.rineng.2024.102120.

[4]M. O. Asali, F. Hadary, and B. W. Sanjaya, “Modeling, simulation, and optimal control for two-wheeled self-balancing robot,” International Journal of Electrical and Computer Engineering (IJECE), 2017, vol. 7, no. 4, pp. 2008–2017, doi: 10.11591/ijece. v7i4.pp2008-2017.

[5]Das, Avishek & Rahman, Mizanur & Adib, Abidur & Anam, Mohammad., “Self-Balancing Robot with PID Controller,” Inproceedings, 2021.

[6]A. S. Shekhawat and Y. Rohilla, "Design and Control of Two-wheeled Self-Balancing Robot using Arduino," International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, pp. 2020,1025-1030, doi: 10.1109/ICOSEC49089.2020.9215421.

[7]A., Ahmed & Mohammed, Akram. “Design and Implementation of Effective LQR Controller for Self-Balancing Robot”, 7th International Conference of Computer Science and Information Technology, Khartoum, Sudan, 2020, pp.1-5.

[8]A. Abdelgawad, Tarek Shohdy, Ayman Nada,” Model- and Data-Based Control of Self-Balancing Robots: Practical Educational Approach with LabVIEW and Arduino”, IFAC-Papers On Line, 2024, Vol.58. 9, pp. 217-222, https://doi.org/10.1016/j.ifacol.2024.07.399.

[9]M. Majczak and P. Wawrzyński, "Comparison of two efficient control strategies for two-wheeled balancing robot," 20th International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 2015, pp. 744-749, doi: 10.1109/MMAR.2015.7283968.

[10]P. J-H, Cho B-K. “Development of a self-balancing robot with a control moment gyroscope”. International Journal of Advanced Robotic Systems. 2018;15(2). doi:10.1177/1729881418770865.

[11]Jimenez, Andres & Jiménez L., Fabián & Ruge, Ilber. “Modeling and Control of a Two Wheeled Self-Balancing Robot: a didactic platform for control engineering education”, 18th LACCEI International Multi-Conference for Engineering, Education, and Technology: Engineering, Integration, And Alliances for A Sustainable Development” “Hemispheric Cooperation for Competitiveness and Prosperity on A Knowledge-Based Economy, 2020. 10.18687/LACCEI2020.1.1.556.

[12]Johnson, T., Zhou, S., Cheah, W. et al. “Implementation of a Perceptual Controller for an Inverted Pendulum Robot”, J Intell Robot Syst., 2020, pp683-692. https://doi.org/10.1007/s10846-020-01158-4

[13]Ch. Zhang, Xin Xu, Xinglong Zhang,” Dual heuristic programming with just-in-time modeling for self-learning fault-tolerant control of mobile robots”, Optimal Control Applications and Methods, 2021, https://doi.org/10.1002/oca.2791

[14]Johnson, Aja and Abu Nasar. “Design and Development of a Two Wheeled Self Balancing Robot for an Application of Object Carrying.” *International Journal of Engineering Research and* 6, 2017.pp.:1-12.

[15]B. H. Prasetio, and Wijaya Kurniawan,” Disturbance rejection using feed-forward control system on self-balancing robot”, MATEC Web Conf. 154 03002 (2018) DOI: 10.1051/matec conf/201815403002

[16]W. Sun, S. -F. Su, J. Xia and Y. Wu, "Adaptive Tracking Control of Wheeled Inverted Pendulums with Periodic Disturbances," in IEEE Transactions on Cybernetics, vol. 50, no. 5, pp. 1867-1876, May 2020, doi: 10.1109/TCYB.2018.2884707.

[17]A. Ghahremani and A. K. Khalaji, "Simultaneous regulation and velocity tracking control of a two-wheeled self-balancing robot," International Conference on Control, Automation and Diagnosis (ICCAD), Rome, Italy, 2023, pp. 1-6, doi: 10.1109/ICCAD57653.2023.10152339.

[18]M. Zhang, Y. Cao and J. Huang, "Soft Prescribed Performance Control of Mobile Wheeled Inverted Pendulum Systems Subjected to Unknown Disturbances,"IEEE International Conference on Cyborg and Bionic Systems (CBS), Wuhan, China, 2023, pp. 356-361, doi: 10.1109/CBS55922.2023.10115343.

[19]Hu, Chuan, Minhao Liu, Lei Wang, and Hui Pang. "Optimization-based adaptive trajectory tracking controller design of self-balanced vehicle with asymptotic prescribed performance." ISA transactions 148 (2024): 114-127.

[20]Hou, Ming, Xuedong Zhang, Du Chen, and Zheng Xu. "Hierarchical sliding mode control combined with nonlinear disturbance observer for wheeled inverted pendulum robot trajectory tracking." Applied Sciences 13, no. 7 (2023): 4350.

[21]Jung, Seul, and Sung Su Kim. "Control experiment of a wheel-driven mobile inverted pendulum using neural network." IEEE Transactions on Control Systems Technology 16, no. 2 (2008): 297-303.

[22] Wang, F.-C.; Chen, Y.-H.; Wang, Z.-J.; Liu, C.-H.; Lin, P.-C.; Yen, J.-Y. Decoupled Multi-Loop Robust Control for a Walk-Assistance Robot Employing a Two-Wheeled Inverted Pendulum. Machines 2021, 9, 205.

[23] Moness, M.; Mahmoud, D.; Hussein, A. Real-time Mamdani-like fuzzy and fusion-based fuzzy controllers for balancing two-wheeled inverted pendulum. J. Ambient. Intell. Humaniz. Comput. 2020, 13, 3577–3593.

[24] Jmel, I.; Dimassi, H. Hadj-Said, S.; M'Sahli, F. "An adaptive sliding mode observer for inverted pendulum under mass variation and disturbances with experimental validation". ISA Trans. 2020, 102, 264–279.

[25] J. -X. Xu, Z. -Q. Guo and T. H. Lee, "Design and Implementation of Integral Sliding-Mode Control on an Underactuated Two-Wheeled Mobile Robot," in IEEE Transactions on Industrial Electronics, vol. 61, no. 7, pp. 3671-3681, July 2014, doi: 10.1109/TIE.2013.2282594.

[26] J. Huang, F. Ding, T. Fukuda and T. Matsuno, "Modeling and Velocity Control for a Novel Narrow Vehicle Based on Mobile Wheeled Inverted Pendulum," in IEEE Transactions on Control Systems Technology, vol. 21, no. 5, pp. 1607-1617, Sept. 2013, doi: 10.1109/TCST.2012.2214439.

[27] A. Mathew, R Ananthu, P Binsy, Abdul Vahid, Ciby Thomas and S Sidharthan," Design and control of a two-wheel self-balancing robot
" IOP Conf. Ser.: Mater. Sci. Eng. 1114 012058DOI 10.1088/1757-2021,899X/1114/1/012058

[28] Jones, Daniel, Stol, Karl. "Modelling and Stability Control of Two-Wheeled Robots in Low-Traction Environments" 2010.

[29] R. Petcu and G. -D. Andreescu, "Two-Wheeled Inverted-Pendulum Self-Balancing Robot with LQR or PID Control," IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC), Reykjavík, Iceland, 2022, pp. 000345-000350, doi: 10.1109/ICCC202255925.2022.9922862.

[30] Kim, Y.; Kwon, S. Robust Stabilization of Underactuated Two-Wheeled Balancing Vehicles on Uncertain Terrains with Nonlinear
Model-Based Disturbance Compensation. Actuators 2022, 11, 339.

[31] Israilov S, Fu L, Sánchez-Rodríguez J, Fusco F, Allibert G, Raufaste C." Reinforcement learning approach to control an inverted pendulum: A general framework for educational purposes". PLoS ONE 2023,18(2): e0280071. https://doi.org/10.1371/journal.pone.0280071

[32] Zheng, Y., Li, X. & Xu, L. "Balance Control for the First-order Inverted Pendulum Based on the Advantage Actor-critic Algorithm". Int. J. Control Autom. Syst. 2020,18, 3093–3100. https://doi.org/10.1007/s12555-019-0278-z

[33] Manrique Escobar, Camilo Andrés, Carmine Maria Pappalardo, and Domenico Guida. "A Parametric Study of a Deep Reinforcement Learning Control System Applied to the Swing-Up Problem of the Cart-Pole" Applied Sciences,2020,10, no. 24: 9013. https://doi.org/10.3390/app10249013

[34] I. Koryakovskiy, Manuel Kudruss, Robert Babuška, Wouter Caarls, Christian Kirches, Katja Mombaur, Johannes P. Schlöder, Heike Vallery, "Benchmarking model-free and model-based optimal control,
Robotics and Autonomous Systems", 2017, Vol 92,2017, pp.: 81-90, ISSN 0921-8890, https://doi.org/10.1016/j.robot.2017.02.006.

[35] Rahman, M.M., Rashid, S.M.H. & Hossain, M.M. Implementation of Q learning and deep Q network for controlling a self-balancing robot model. Robot. Biomim. 2018,5, 8. https://doi.org/10.1186/s40638-018-0091-9.