**Enhancing Stability: Novel Control Techniques for Two-Wheeled Self-Balancing Robots**

Dr. Vinod Kumar P1, Kamala Narayanan2, Praveen Kumar M3,

Amruth Ramesh Thelkar4\*

1,3Department of Electrical and Electronics Engineering, ATME College of Engineering, Mysuru, Visvesvaraya Technological University, Belgaum, Karnataka India

2Department of Electrical Engineering, University of Nevada Las Vegas, Las Vegas, United States of America

4Electrical & Electronics Engineering Department, Nitte Meenakshi Institute of Technology

Bangalore, Visvesvaraya Technological University ,Karnataka,India & Faculty of Electrical & Computer Engineering, Jimma Institute of Technology,Jimma University, Ethiopia.

Corresponding Author Email: [amruth.ramesh@ju.edu.et](mailto:amruth.ramesh@ju.edu.et), [amruth.ramesh@nmit.ac.in](mailto:amruth.ramesh@nmit.ac.in)

**Abstract**

The article focuses on developing a TWSBR (Two-Wheeled Self-Balancing Robots) controller aimed at enhancing the performance of underperforming robotic systems, particularly in maintaining stability and precision during movements. It underscores the importance of a novel control approach to address specific performance metrics such as balance, agility, and responsiveness. The paper outlines the establishment of a non-linear system model to capture the intricate dynamics of robotic movements. It briefly discusses the incorporation of non-linearity within the model, potentially involving factors like frictional forces or dynamic load variations. This non-linear framework helps in addressing inherent complexities, thereby improving the performance of two-wheeled robotic systems, especially in scenarios requiring precise balance and regulation. The paper evaluates the performance of various controllers, such as LQR (Linear Quadratic Regulator), ANN (Artificial Neural Network), and PID (Proportional-Integral-Derivative), within the context of the TWSBR system. This analysis provides insights into the effectiveness of different control approaches in improving system stability and precision.

**Keywords:** Underactuated system, Linear Quadratic Regulator (LQR), Artificial Neural Networks (ANN), Two-Wheeled Self-Balancing Robot (TWSBR), Proportional Integral Derivative (PID).

**1. Introduction**

The development of two-wheel self-balancing robots is a significant advance in the field of mechanical technology. The design of the system relies on the Inverted pendulum hypothesis, which indicates that a framework composed of sensors and microcontrollers is needed to balance the robot. Due to their small size and force requirements, these robots are ideal for various applications, such as inventory management, surveillance, and autonomous delivery within indoor environments.

This paper aims to design electro-mechanical systems that can control the movement of a robot. In addition to this, the paper also focuses on the control calculations that are required to enable the system to perform continuously. The Inverted Pendulum Hypothesis models the dynamics of a system where a rigid body is balanced on a single pivot point, resembling an inverted pendulum. In the context of two-wheel self-balancing robots, this hypothesis simplifies the representation of the robot's dynamics, considering the robot as an inverted pendulum with the wheel axis serving as the pivot point. The system's equilibrium is analogous to the upright position of the inverted pendulum. The robot's stability is inherently tied to the management of the inverted pendulum dynamics. Deviations from the equilibrium position trigger control mechanisms, such as the application of torques through the wheels or adjustments in the robot's center of mass, to restore balance. The adoption of these controllers is motivated by their proven effectiveness in addressing specific challenges in robotics. LQR provides a robust framework for linear control, ensuring stability and precision in controlled movements [1]. Simultaneously, ANN contributes to the adaptability and learning capabilities of the robot, allowing for real-time adjustments based on environmental conditions [2]. Additionally, the simplicity and effectiveness of PID control are harnessed for regulating specific aspects of the system, contributing to overall system stability [3]. The paper is structured into several sections, including a Review of Literature, Mathematical Modelling, Controller Design, and Comparison of Controller Performance to assess stability and performance in the presence of disturbances.

**2. Related works**

The control process is performed in a virtual environment, and the results of the simulation are shown to demonstrate its efficiency and performance. The system is controlled using the Arduino Due Microcontroller, which features a 9-DoF, DC motors, and a Bluetooth module [[1,2]. The LQR and PID controllers are used for the stability of the system. The performance of the system is shown by its sensitivity to the noise generated by the sensors. Two-wheeled robots with a gyroscope module can operate independently and have better mobility. Due to its size and movement, the movement of one or two wheels can affect the balance of the system [3]. To compensate its body tilt, the control moment gyroscope module can generate torque to reduce the disturbance caused by its movement. The system is also supported by a pmod and onboard accelerometer. The free PD and filter pendulum controllers from Lab View are additionally utilized for testing the system [4]. The most challenging issues in designing a self-balancing system are the weight and balance of the robot. In order to prevent the system from getting unstable, a dynamic model is developed using the Kalman filter and the PID controller [5]. The complexity of the Kalman filter is considered when it comes to filtering [6]. A complimentary filter is also utilized for this purpose. The non-linear equation for the constraint is then derived. The control methods used for the system are LQR, and LQG [7]. The mathematical modeling of the system is performed to verify its stability [8]. The performance of the LQG, LQR, and PID controllers is analyzed after the implementation of the mathematical modeling of an inverted pendulum robot [9]. A double loop control scheme is also implemented to regulate the robot's pitch angle and speed [10]. The development of a robotic system with a fuzzy adaptive control method is carried out due to the deficiency of the LQR and the PID controllers in terms of overshoot and settling time [11]. The system is composed of a Kalman filter, a signal processing unit, and a control algorithm [12]. The PD controller ensures that the motor achieves its velocity [13]. The development of a wheel and frame structure for a self-balancing system is also carried out. This process can lead to the creation of a 2-DoF unstable system [14]. To enable the system to perform in real time, the design of a controller that can handle position, disturbance rejection, and vertical balance needs to be analyzed. The development of real-time implementations of self-balancing systems is also carried out. The components used in this process include a single-axis gyroscope, a 2-axis accelerometer, and an Arduino microcontroller[15]. To prevent the system from getting unstable, a dynamic model is developed using the Kalman filter and the PI-PD controller. The kinematic dynamic model utilizes the LQR and PID controllers for regulating the Yaw rotation of the robot[16]. The latter is used for controlling the two subsystems while the former is utilized for the self-balancing system. Simulation results show that the angle of the robotic position and the SMC controller is stable[17]. The development of sliding mode functions for the system improves the quality of the system. The feedback from the three sensors is also taken into account to control the system. The parameters of the model, such as the tilt angle, velocity, and angular position, are then measured using the gyroscope and accelerometer. The Kalman filter then provides a state space representation of the data collected by the sensors[18]. To summarize the research challenge in simulating self-balancing robots include:

**Developing accurate models:** Building accurate models of self-balancing robots can be a challenge, as they involve complex mechanical and control systems. Creating accurate models requires a deep understanding of the robot's design and operation.

**Simulating external disturbances:** Simulating external disturbances such as bumps, obstacles, and wind can be challenging. These disturbances can affect the robot's stability and control and simulating them accurately can help researchers test and improve their algorithms.

**Validating simulation results:** Ensuring that the simulation results accurately reflect the real-world behavior of self-balancing robots is crucial. This involves developing validation methods to verify that the simulation results are accurate and reliable.

**3. Mathematical modelling**

Following are the equations of motion [19] [20] because of the moment and forces which is acting on the left wheel:

(1)

(2)

Similar to the left wheel, the right wheel has:

(3)

(4)

Moments among the pendulum and the platform about the y-axis and forces seriatim at the pendulum within the direction of the x-axis are balanced to supply

(5)

(6)

(7)

(8)

When the moment of the pendulum and platform stage around the z-axis are stable, the following results are produced.:

(9)

The displacements of the wheel alongside the x-axis and the rotational perspective of the wheel about the y-axis have the subsequent courting

(10)

On the other hand, the following equation describes the relationship between the robot's heading angle () with respect to the z-axis and x-axis wheel movement is

(11)

When (1) and (2) are subtracted from (3) and (4), respectively, and then added to the previous equation, the following results:

(12)

(13)

(14)

(15)

When we combine (1) and (2) with (3) and (4), respectively, and then add it to (10), we get:

(16)

(17)

Adding (16) and (17) we have:

(18)

By substituting (18) to (5) and (6) we obtain:

(19)

By substituting (7) to (8) result in:

(20)

The dynamics equation of motion for the 2-wheeled self-balancing On the assumption that the robotic best capabilities round a tiny operational point, the dynamic model of the robot is then linearized around the points ; ; . After adding on equation (20), we get the following:

(21)

Let’s define the state variable as

In our study, we adopted a Taylor series expansion to linearize the non-linear system model. The above linearized system of equation can be represented in state space form as follows (22).

Where:

According to the state-space model that was described earlier, the system possesses a wide variety of dynamic properties that will lead the state matrices to go through a state transition at some point. The following table 1 presents the values of the system's parameters to make simpler the problematic at hand and to provide the state matrix and input matrix with a distinct and understandable physical meaning for the purpose of developing a controller [20][21][22][23].

Table-1:System Parameters values

|  |  |  |
| --- | --- | --- |
| **Particular** | **Unit** | **Value** |
|  |  | 2.1 |
|  |  | 0.0313 |
|  |  | 0.25 |
|  |  | 14 |
|  |  | 9.8 |
|  |  | 1 |
|  |  | 0.5 |
|  |  | 5 |
|  |  | 0.0385 |
|  |  | 1.8569 |

# **Design of Controllers**

In this paper, we have design and simulated 3 controllers for two wheeled self-balancing robots. We have started with Linear–quadratic regulator (LQR) controllers , Artificial neural networks (ANN) controllers and also design Proportional Integral Derivative (PID) Controllers. Lastly we have compared the performance of the three controllers with respect to settling time, overshoot and steady state error which determines the system stability. Both the simulation without disturbance and the simulation with disturbance are included in this section. The selection of controllers in our two-wheeled robot design is a strategic decision based on their unique strengths and applications within the field of robotics. The combination of Linear Quadratic Regulator (LQR), Artificial Neural Network (ANN), and Proportional-Integral-Derivative (PID) controllers was chosen to address specific challenges and enhance the overall performance of the system.

**A. LQR Controllers:**

LQR controllers were chosen for their ability to provide robust and optimal linear control. This is particularly advantageous in scenarios where precise and stable movements are essential. Real-world applications of LQR controllers range from aerospace systems to industrial automation, showcasing their effectiveness in achieving high-performance control [1]. In our two-wheeled robot, LQR ensures stability and precision in controlled movements, contributing to the overall efficiency of the design.

**B. ANN Controllers:**

The adoption of Artificial Neural Network (ANN) controllers stems from their capacity for adaptability and learning. This is particularly beneficial in dynamic environments where the robot needs to make real-time adjustments based on changing conditions. Real-world applications of ANN controllers extend to fields such as image and speech recognition, where learning from data is paramount [24]. In our design, ANN enhances the adaptability of the robot, allowing it to respond effectively to diverse environmental factors.

C **PID** Controllers:

PID controllers were chosen for their simplicity and effectiveness in regulating specific aspects of the system. With applications ranging from temperature control in ovens to speed control in motor systems, PID controllers are widely employed in various industries [13]. In our two-wheeled robot, PID control contributes to overall system stability, ensuring reliable performance in different operating conditions.

The combination of these controllers in our design reflects a comprehensive approach, leveraging the strengths of each to create a versatile and high-performing robotic system.

**4.1 LQR Controller**

The LQ problem refers to a scenario in which a set of linear differential equations are used to explain the dynamics of the system, while a quadratic function is used to represent the cost of the system

The LQR controller approach used by the two-wheeled self-balancing robot machine provided output responses: the robotics’ pitch angle (α) and the heading angle (θ). For the purposes of this take a look at, the initial cost of the pitch angle (α) of the balancing robotic became programmed to be -1 rad so that the overall performance of the controller may be simulated. It indicates that the preliminary domain of the robotic is one that is vulnerable to first-rate instability. On the alternative hand, the preliminary state of the heading angle became given a unit step because the reference sign and changed into set to 0 radians when it first initialized. The Control architecture of LQR controller is shown in figure-1.

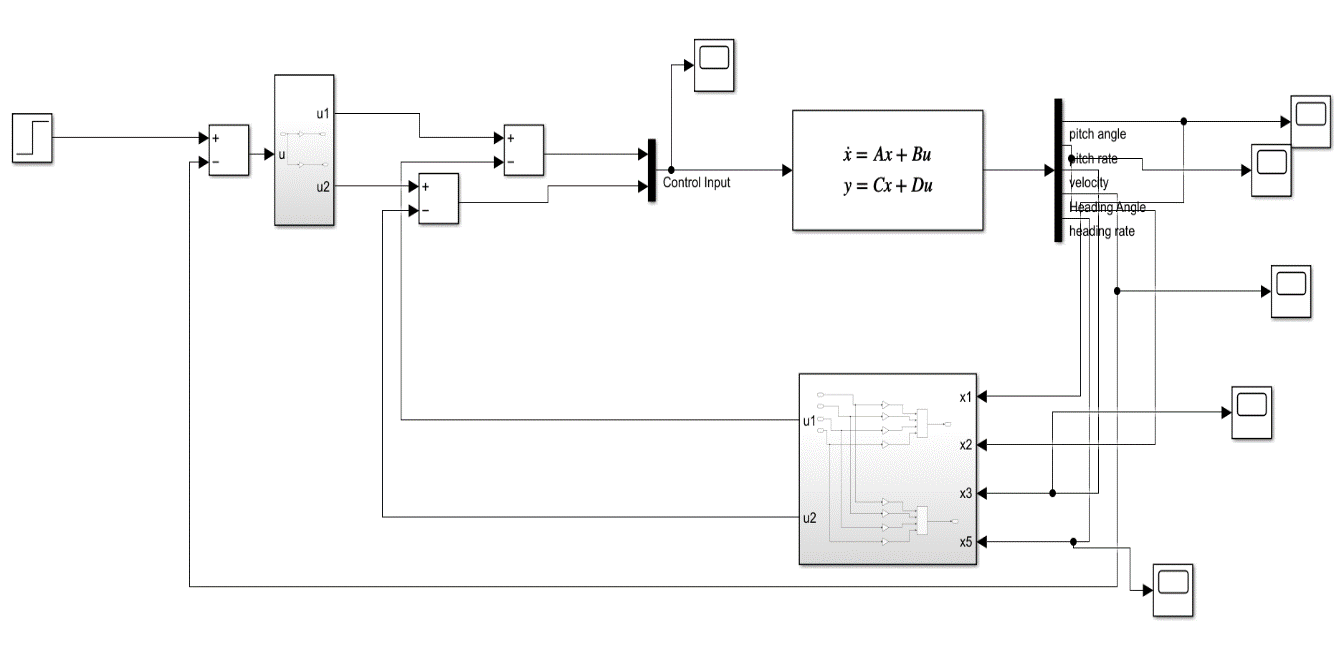
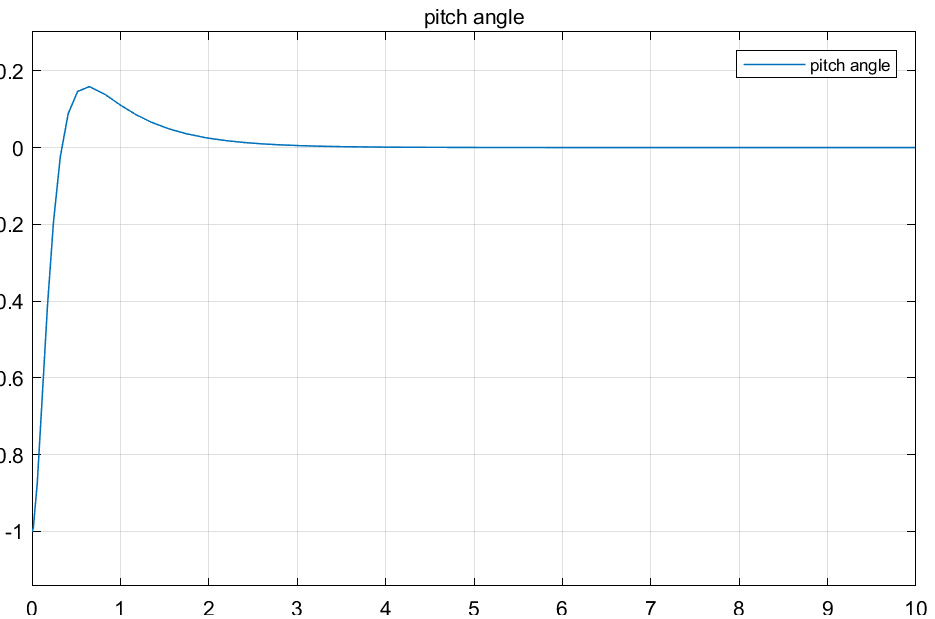


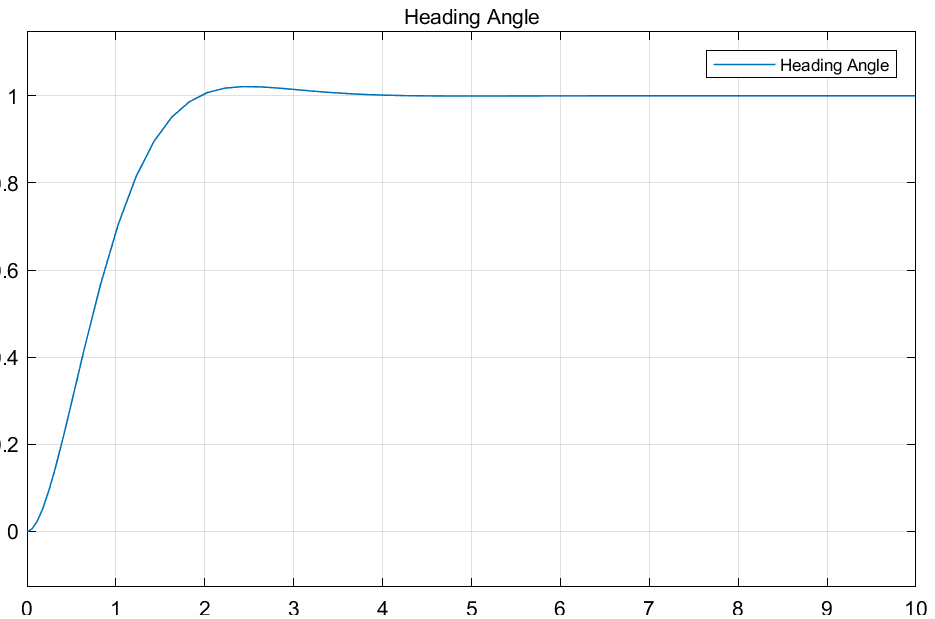
Figure 1 Control architecture of LQR controller of TWSBR



Pitch Angle (degree)

Time in (Seconds)

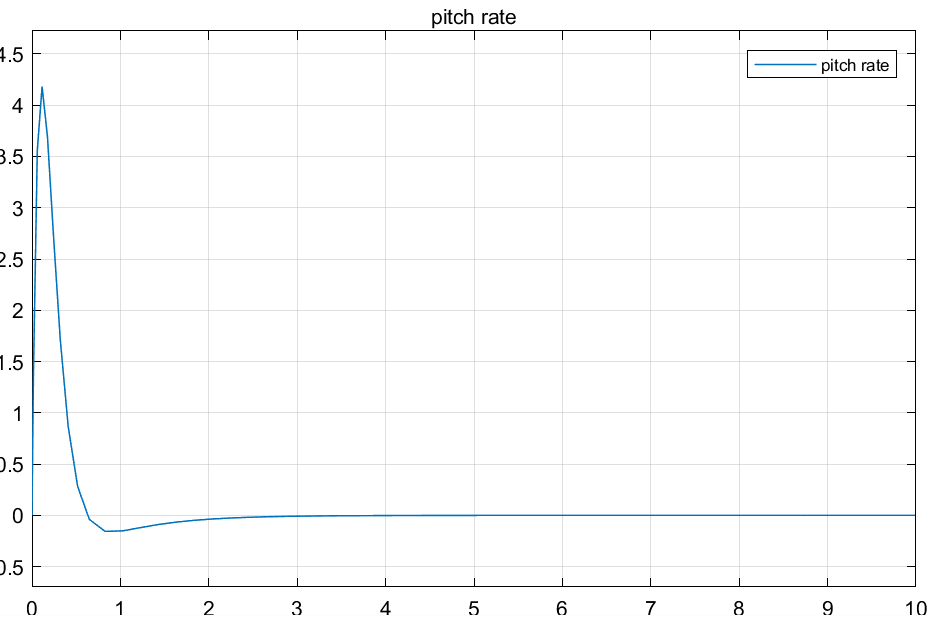
Figure 2 TWSBR Pitch Angle Response of LQR controller in Normal condition



Time in (Seconds)

Heading Angle (degree)

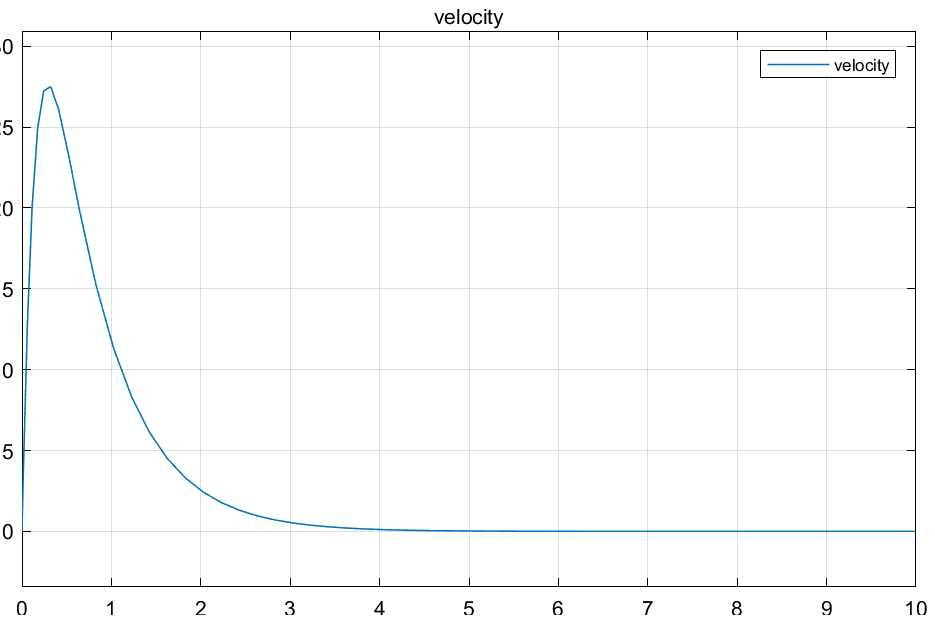
Figure 3 TWSBR Heading Angle response of LQR controller in normal condition



Pitch Rate (rad/sec)

Time in (Seconds)

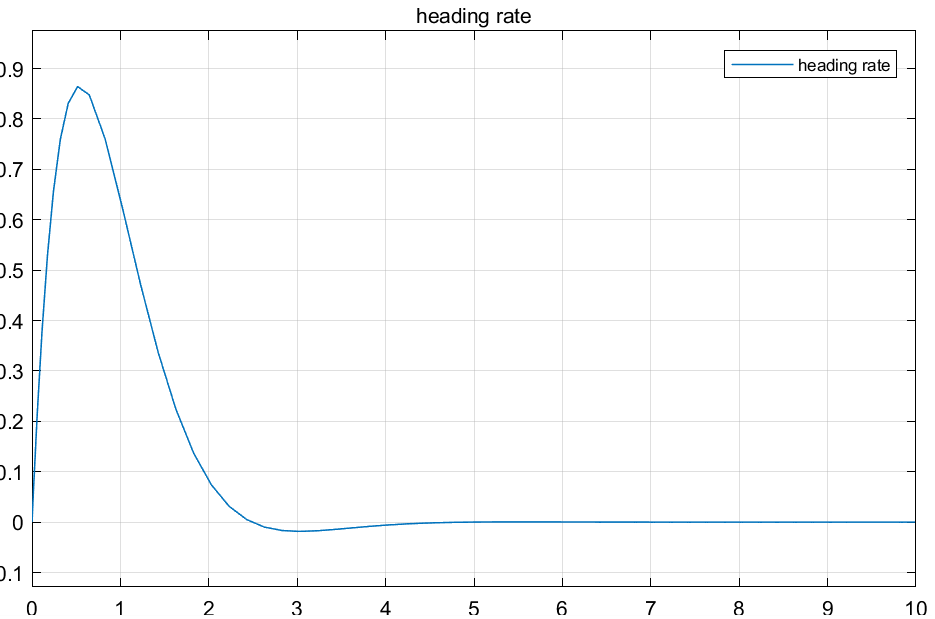
Figure 4 TWSBR Pitch Rate of LQR controller



Velocity (m/sec)

Time in (Seconds)

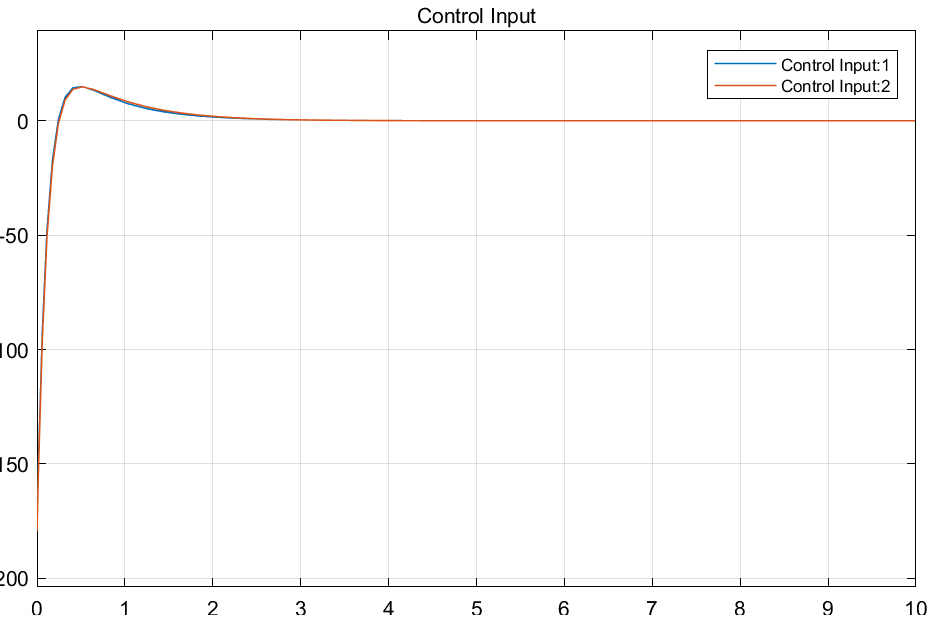
Figure 5 TWSBR Velocity of LQR Controller



Heading Rate (rad/sec)

Time in (Seconds)

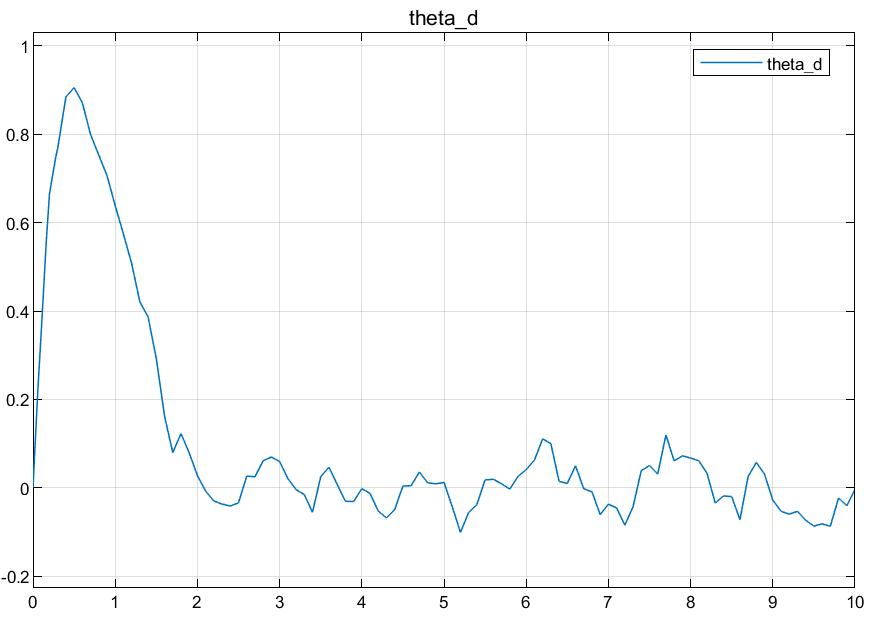
Figure 6 TWSBR Heading Rate of LQR Controller in normal condition.



Control (Degree)

Time in (Seconds)

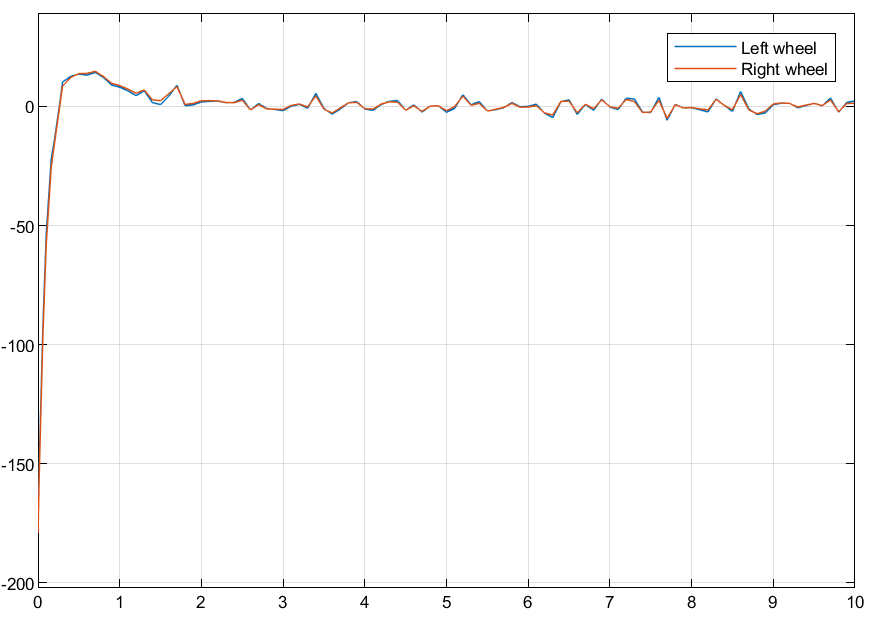
Figure 7 Control effect of TWSBR of LQR Controller in Normal Condition



Time in (Seconds)

Heading rate (rad/sec)

Figure 8 TWSBR heading rate response in normal condition for disturbance signal



Time in (Seconds)

Control (Degree)

Figure 9 Control effort of TWSBR in normal condition for disturbance signal.

Figure 2 indicates the response of TWSBR pitch angle (α) in regular condition at the same time as figure 3 indicates the responses TWSBR heading angle (θ) in ordinary circumstance. The LQR controller response for various parameters like Pitch angle(α), heading angle(θ) are represented in blue color line weight as shown in figure 4, 5 and 6. The graph clearly indicates the LQR controller capability of handling the robot in phased manner towards stability with settling time of 2.23 sec and stability error around 0.0086 rad/sec. The control effort of the device’s response is as shown in Figure 7. The figure- 8 & 9 shows the LQR controller performance for disturbance signal. The torque applied to both wheels show slight deviation from initial heading angle of 1 rad/sec as the robot try to follow the referred value.

**4.2 ANN based Controller**

Artificial intelligence is the field of developing machines that can go beyond human capabilities and take intelligent decisions themselves. Such systems can analyze data and automatically trigger actions without human interface. Today Artificially intelligent systems are able to perform tasks without human intervention by identifying patterns, making decisions and improving themselves through experience and data. AI has now become the heart of technology. AI is a much broader field with a deeper domain; however, we are only interested in covering a very small subdomain of it which is the Machine Learning.

In machine learning, any system, either software or hardware, is trained through some available data and required information is predicted using the relations drawn by the neural network

Mostly AI and Machine Learning is studied under computer sciences however with the advancements in science and technology and the rate at which machine learning is dominating over giving faster solutions than the conventional methods, people from other fields have adopted machine learning to proceed in their own fields.

The difference between using machine learning and traditional programming is that in traditional programming, we use data and process it to find output, however in machine learning we provide data and its output to the model and train it to predict the relation between the two and then use the trained model to predict results for further data.

The main unit of machine learning model is the neural network known as Artificial Neural Network (ANN) is the replica of human brain cell known as Neuron, in terms of machine. It functions in the same way a neuron of human body does.

A Neural Network consists of node layers containing 1 input layer, 1 output layer and one or more hidden layers. Each node connects with the other node having an associated weight and threshold. Neural Networks need to be trained and have their weights and thresholds found, by giving them training data, before they can be brought to use. In supervised learning, the training data includes the inputs and desired outputs and the model trains itself to give the desired outputs when fed by those inputs and accurately predict for other inputs too[25]. Whereas in unsupervised learning, the training data is only limited to having inputs and the algorithm itself identifies the pattern in the inputs contributing to some output and predicts further related inputs.

The most basic neural network consists of the following parts:

* Forward Propagation
* Optimizer
* Backward Propagation
* Activation function

We have used a special type of ANN for this work, this is called the LSTM, which has an additional layer known as long short-term memory[24]. This layer is a special layer which remembers both long- and short-term responses and non-linearities are handled carefully. LSTM has that advantage that it can process sequences of data easily.

The characteristics of built neural network are shown in table 2.

Table-2: Characteristics of Neural Networks

|  |  |
| --- | --- |
| **Characteristic Name** | **Characteristic** |
| Type of Output | Regression |
| Number of Layers | 1 Hidden Layer  1 Input Layer  1 Output Layer  1 LSTM Layer |
| Number of Inputs | 4 |
| Number of Outputs | 12 |
| Dropout percentage | 50% |
| Number of nodes in Hidden layer | 500 |
| Optimizer | Adam |

## **4.2.1 Training of ANN:**

The ANN has been trained using a dataset of 10000 possibilities having defined gains.

The learning curve is shown in figure-10.

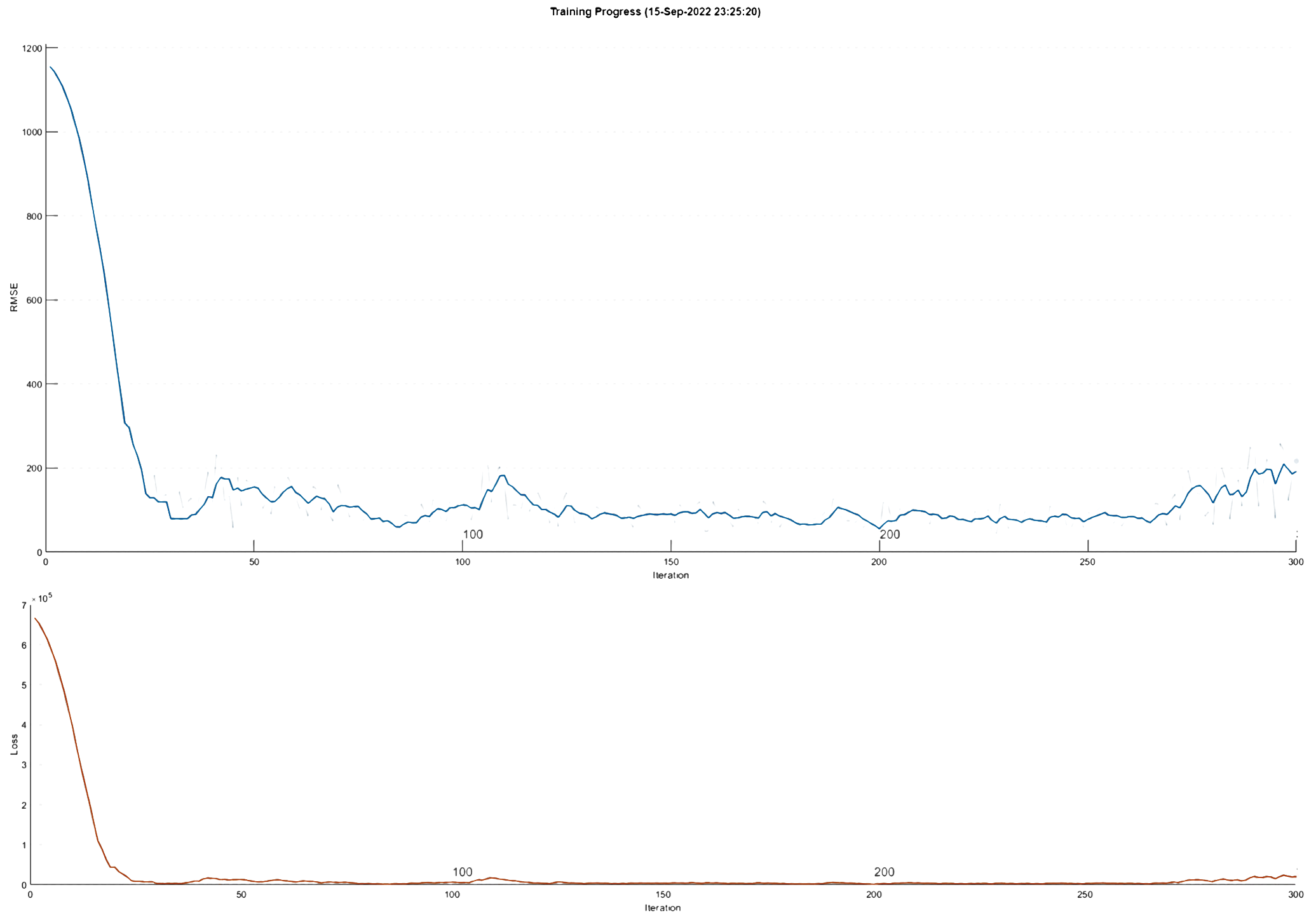


Figure 10 Data Training for ANN controller Design

The training took almost 30 hours to complete. With this kind of training, self-balancing robot is likely to be very stable and perform well in a variety of situations. An ANN can be utilized to train self-balance robots.

To start gathering data on our robot's behavior need to obtained. Utilizing sensors enables the collection of data concerning aspects such as the robot's angle, velocity, and acceleration. Subsequently, machine learning algorithms come into play to train the ANN in forecasting the robot's most stable movements. We can see that with each iteration, the model loss and root mean square error decreases substantially. And the training stopped when the loss has minimized[24].

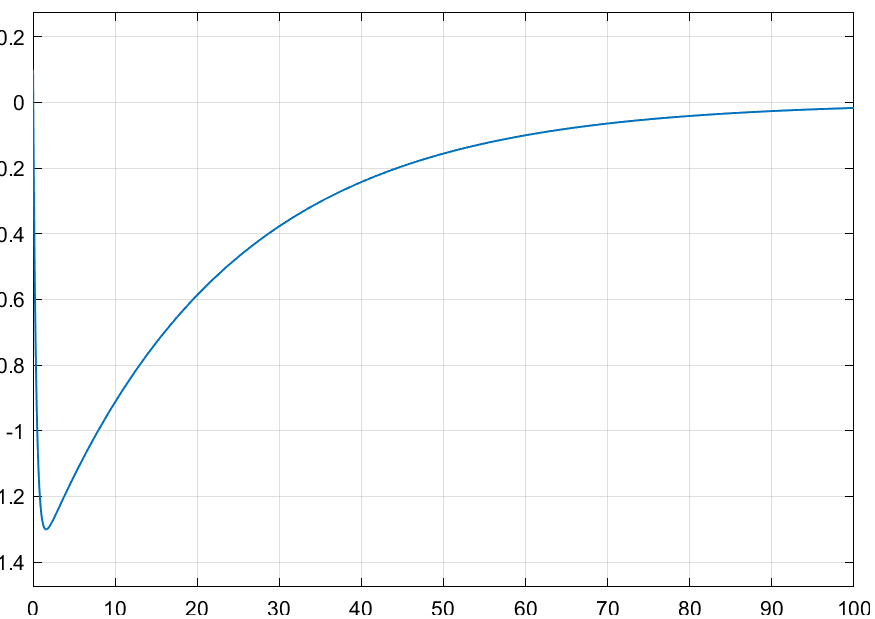
Now if we give some random input and with the help of the neural network, we will find the gains of the neural network and see our responses. The gains are updated in following block diagram as shown in figure 11.

Diagram

Description automatically generated

Figure-11: ANN architecture

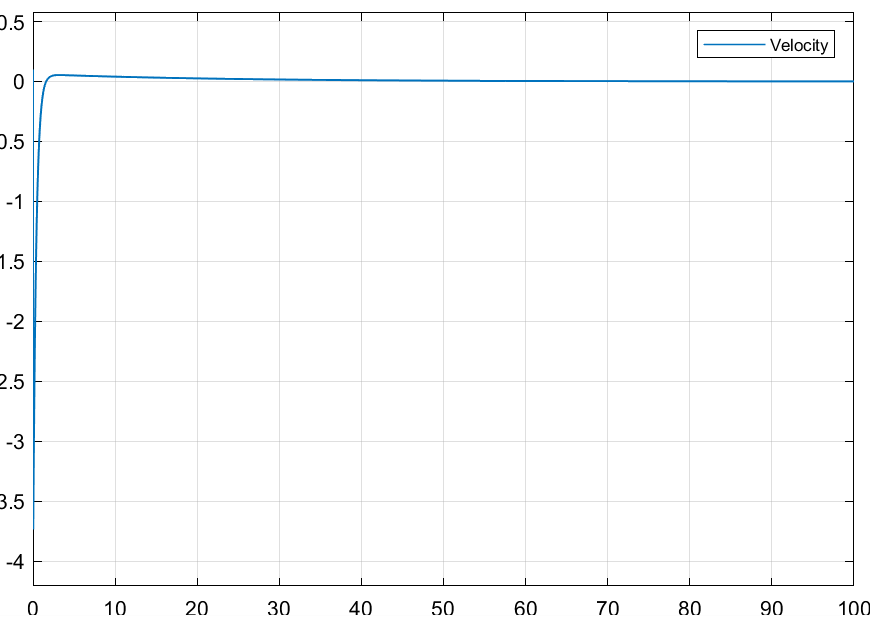
The responses of TWSBR similar to LQR controllers for different parameters is shown in figure from 12 to 17.



Time in (Seconds)

Position

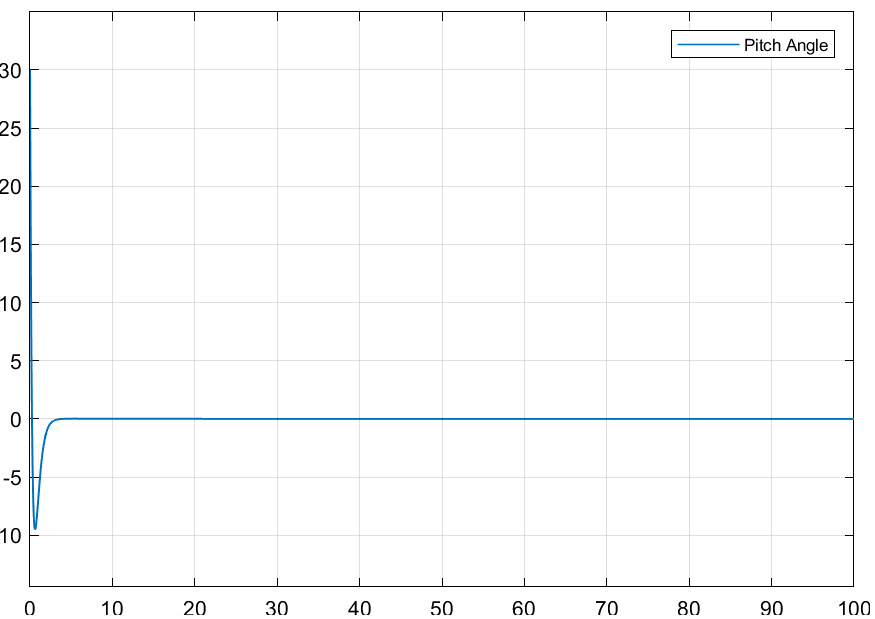
Figure 12 TWSBR Position using ANN Controller



Time in (Seconds)

Velocity (m/sec)

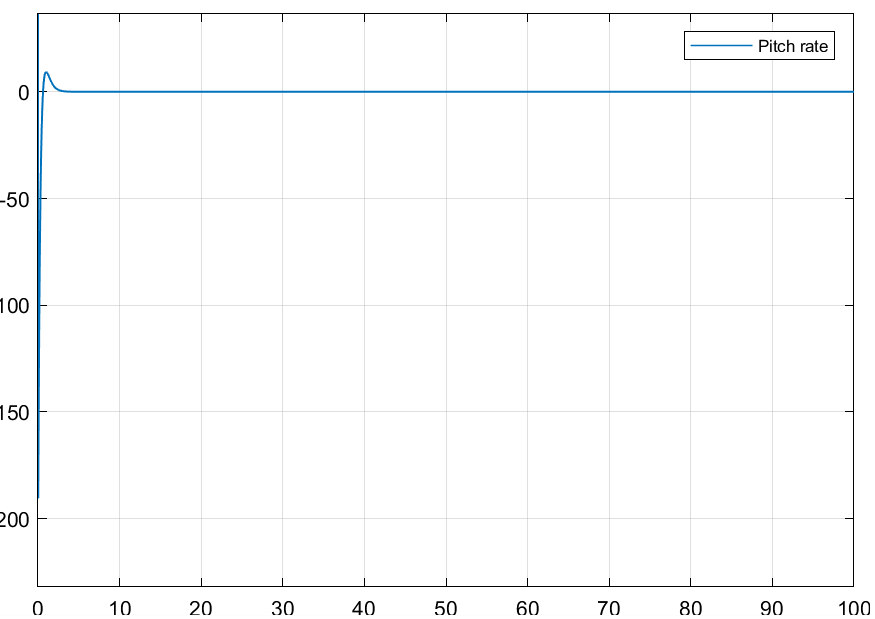
Figure 13 TWSBR Velocity using ANN Controllers



Time in (Seconds)

Pitch Angle (degree)

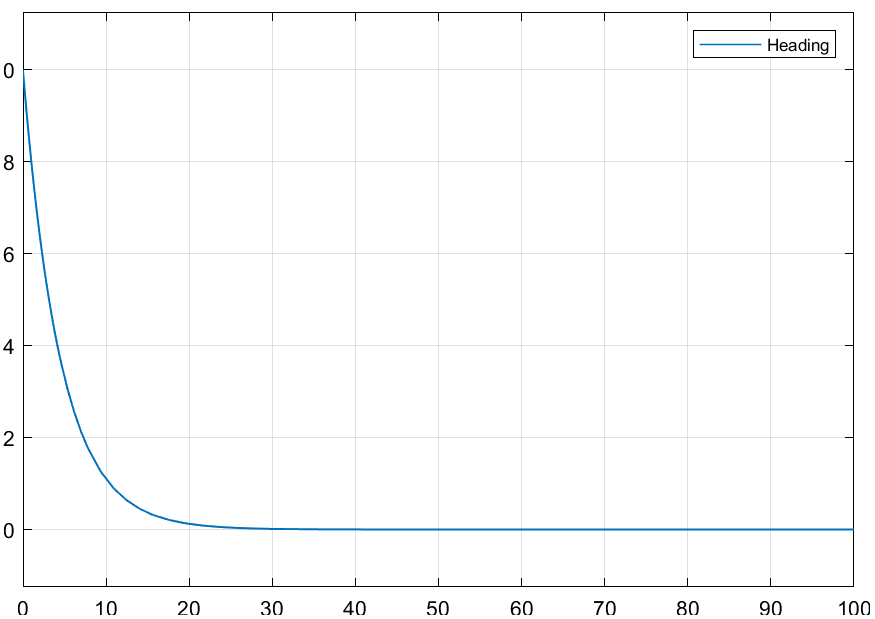
Figure 14 TWSBR pitch angle using ANN Controller



Pitch Rate (rad/sec)

Time in (Seconds)

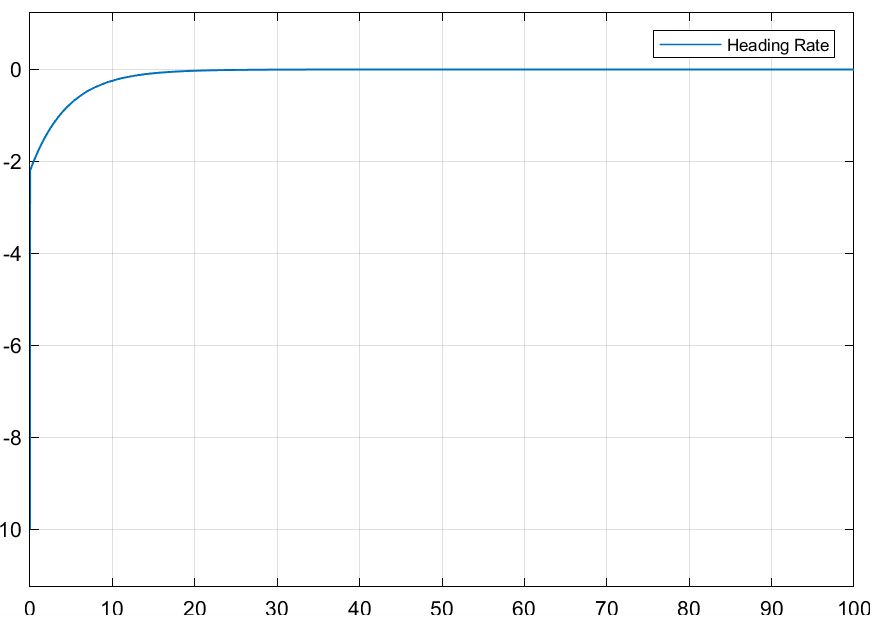
Figure 15 TWSBR Pitch Rate using ANN Controller



Time in (Seconds)

Heading Angle (degree)

Figure 16 TWSBR Heading Angle using ANN Controller



Heading Rate (rad/sec)

Time in (Seconds)

Figure 17 TWSBR Heading rate using ANN Controller

Although the responses are a little different from the responses, we received from the controller without neural network but we can see that all the responses are in acceptable limits. The results can further be enhanced by training the neural network in such a way that the loss and RMSE minimize further.

# **4.3 PID Based Controller**

The simulation was performed using the controller that was proposed before. MATLAB/Simulink is the simulation tool that is used in order to develop and stimulate the PID Controller for the system. This part also devotes considerable attention to analyzing the controller's performance characteristics in minute detail. The PID controller approach used by the TWSBR system provided the same initial conditions as that of LQR in which two output responses pitch angle (α) and the heading angle (θ) shows great instability. The initial angle was set to 0 and response are obtained. The Control architecture is shown in figure 18. The response of TWSBR for PID controllers is shown in figure 19 to 24 for different parameters.

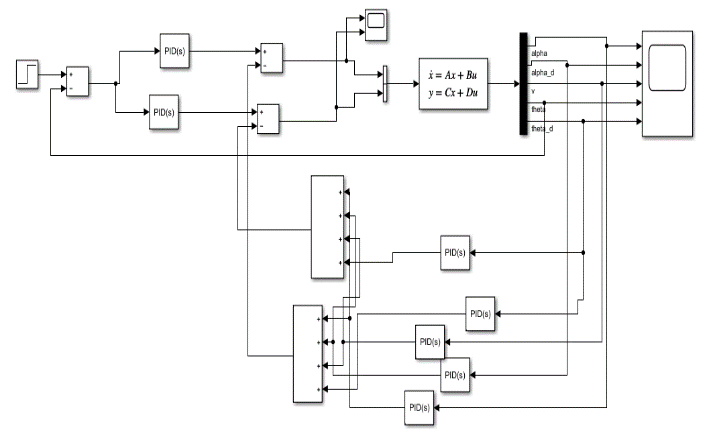
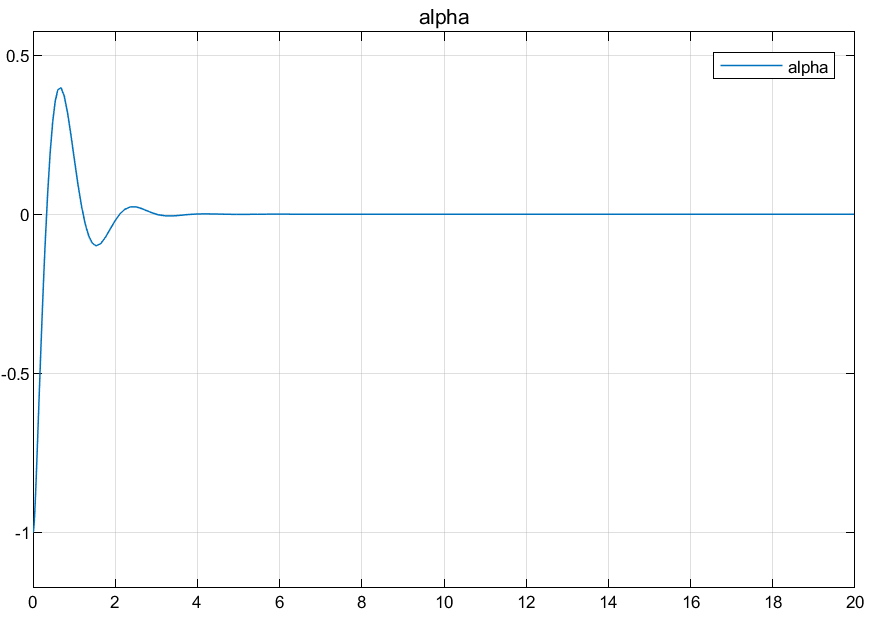


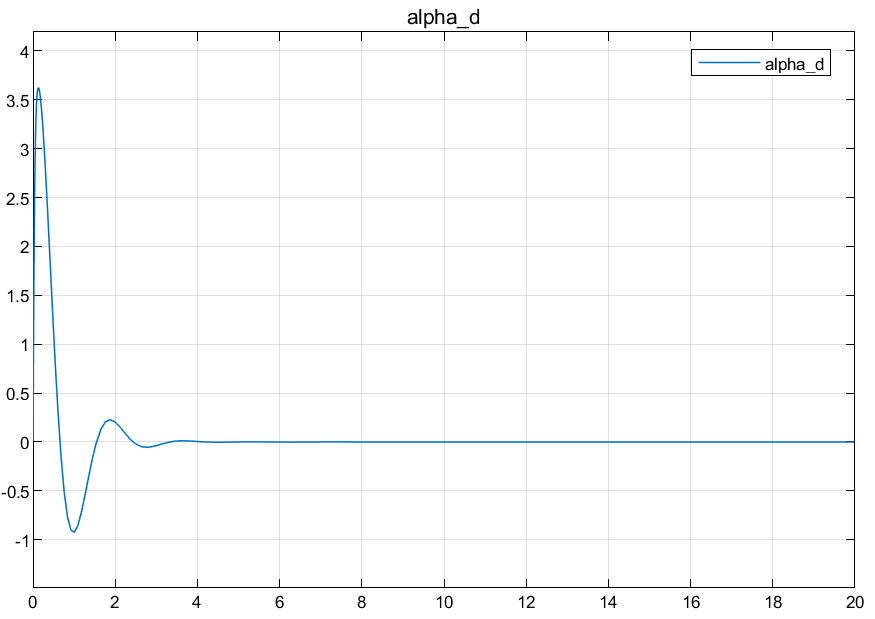
Figure 18: PID architecture



Alpha (degree)

Time in (Seconds)

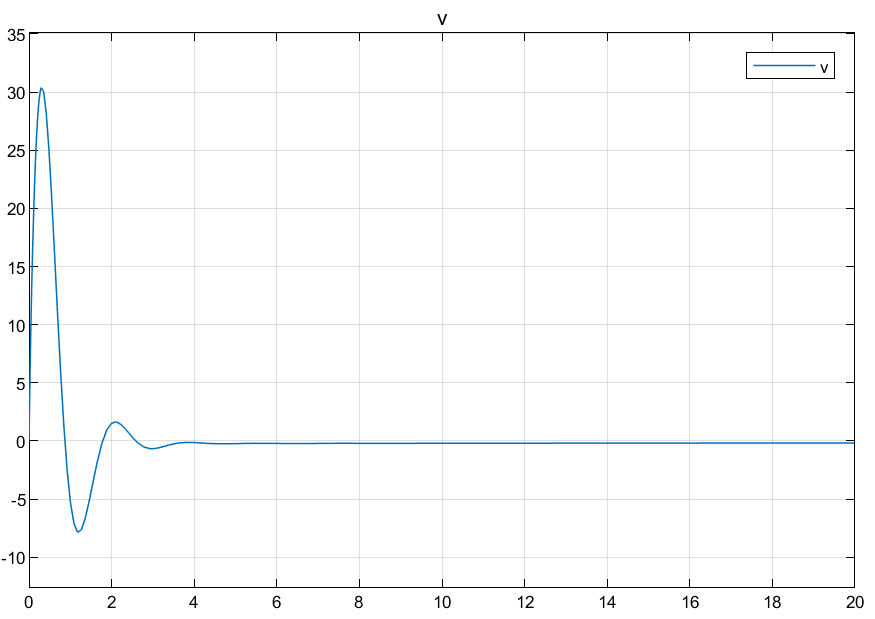
Figure 19 TWSBR pitch angle using PID Controller



Alpha dot (rad/sec)

Time in (Seconds)

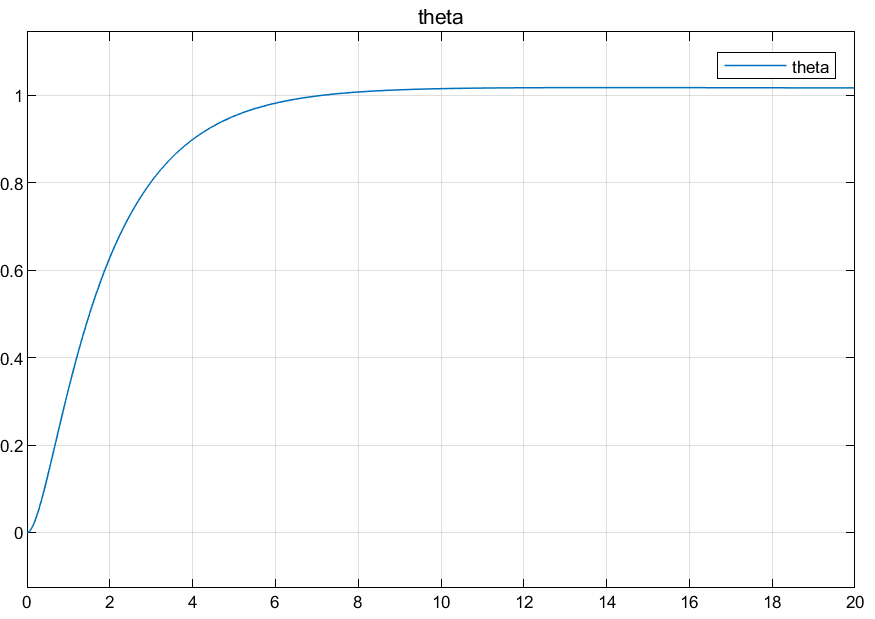
Figure 20 TWSBR pitch rate using PID Controller



Velocity (m/sec)

Time in (Seconds)

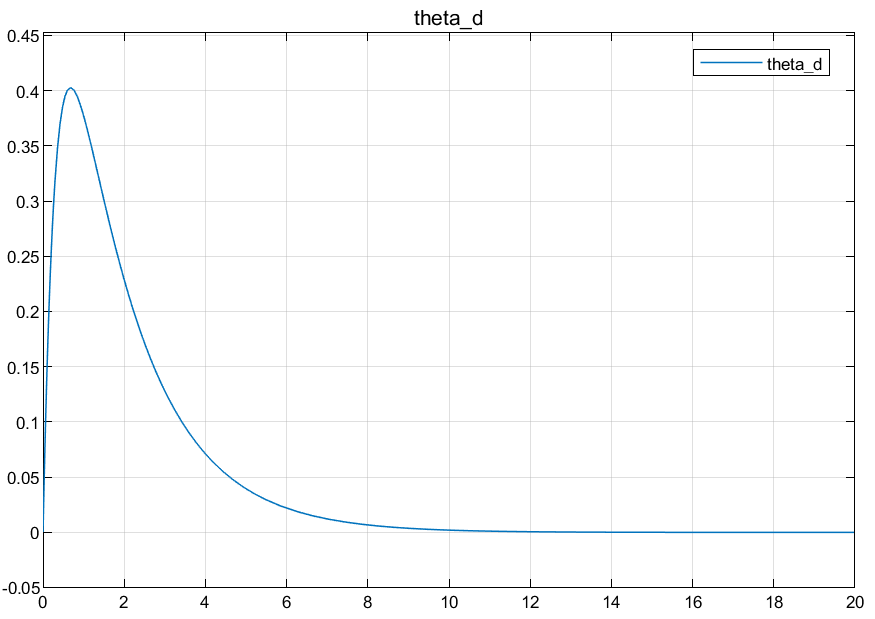
Figure 21 TWSBR Velocity using PID Controller



Heading angle (degree)

Time in (Seconds)

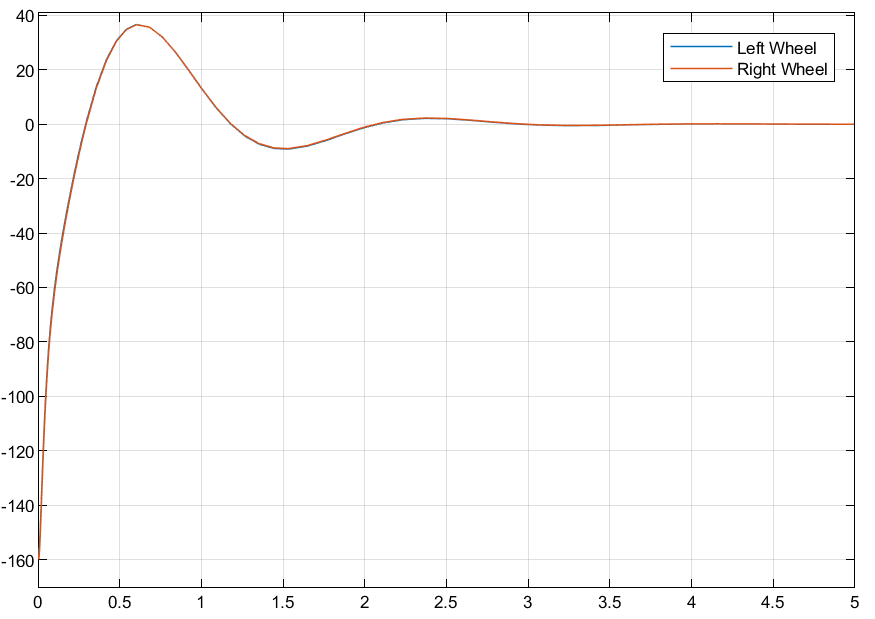
Figure 22 TWSBR heading angle in normal condition using PID Controller



Time in (Seconds)

Heading rate(rad/sec)

Figure 23 TWSBR heading rate response in normal condition using PID Controller



Time in (Seconds)

Control (Degree)

Figure 24 Control effort of TWSBR in normal condition using PID Controllers

1. **Conclusion**

The comparison of all controllers is given below table -3.

Table-3: Stability Performance of Controllers

|  |  |  |  |
| --- | --- | --- | --- |
| **Controller** | **Settling Time** | **% Overshoot** | **Steady State Error** |
| **PID** | 5 sec | 20% | 0 |
| **LQR** | 1.8 sec | 2% | 0 |
| **ANN** | 1.5 sec | 0.25% | 0 |

Initially we designed LQR controller which is quite optimized controller, and we can give weights to desired states. The results of the LQR are shown above. The states calculated using LQR has less overshoot and they settled very fast. Also, the control input required in LQR is very reasonable. We have given it unstable initial position and the controller extra efficiency that it stabilizes it. After that we have designed the same system using PID. The result plotted shows overshoot which is greater than LQR. Similarly, the values of gains are quite high which requires a lot of torque or control input from the motors, so a PID controller is less efficient as compared with the LQR. Now move on to our final controller which is based on neural network, we have trained our system based on the data provided which consists of different values of controllers. The control architecture use is that of the LQR one. We have trained the system such a way based on the data that whatever the inputs of the system, whatever the size of the system, it automatically calculates best gains for our system. The TWSBR controller introduces a pioneering method for achieving stability and precision in two-wheeled robotic systems, distinguishing itself from conventional control strategies. By combining the strengths of Linear Quadratic Regulator (LQR), Artificial Neural Network (ANN), and Proportional-Integral-Derivative (PID) controllers, the TWSBR controller offers a holistic and adaptive solution to the identified performance challenges. The TWSBR controller leverages the benefits of a carefully designed linearization method, enabling the application of classical control techniques to a non-linear system model. This strategic integration results in enhanced control precision, adaptability, and real-time responsiveness. Furthermore, the TWSBR controller showcases simplicity and efficiency, making it applicable to a wide range of robotic platforms. The proposed TWSBR controller holds promise for various applications in the field of robotics, particularly in scenarios demanding precise balance and regulation. Potential applications include but are not limited to autonomous vehicles, delivery drones, and humanoid robots. Its adaptability to different operating conditions positions the TWSBR controller as a versatile solution for dynamic and unpredictable environments.

**Data Availability**The data used to support the findings of this study are available from the corresponding author upon request.

**Conflict of Interest**

Authors have declared no conflict of interest.

**Acknowledgments**

We declare this work is an independent work and no financial assistance has been received for the work.

##### **References**

1. Long Chen , Hai Wang , Yunzhi Huang , Zhaowu Ping , Ming Yu , Xuefeng Zheng , Mao Ye , Youhao Hu” Robust hierarchical sliding mode control of a two-wheeled self-balancing vehicle using perturbation estimation” , *2019 Mechanical Systems and Signal Processing* Volume 139, May 2020, 106584, Elsevier.
2. Ines Jmel, Habib Dimassi, Salim Hadj-Said, Faouzi Msahli , “An adaptive observer for two wheeled self-balancing robot with a varying center of mass”, 2019 19th *international conference on Sciences and Techniques of Automatic control & computer engineering (STA)*, Sousse, Tunisia, March 24-26, 2019.
3. Vincent Y. Philippart, Kristian O. Snel, Antoine M. de Waal, Jeedella S.Y. Jeedella, Esmaeil Najafi, “Model-based Design for a Self-balancing Robot using the Arduino Micro-controller Board”, *2019 IEEE*.
4. Ji-Hyun Park and Baek-Kyu Cho, “Development of a self-balancing robot with a control moment gyroscope”, *2018 International Journal of Advanced Robotic Systems*.
5. Shyamala Sarathy, Mariyam Hibah M M, Anusooya S, S.Kalaivani “Implementation of Efficient Self Balancing Robot” *2018 IEEE*.
6. Md. Iman Ali and Md. Modasser Hossen “A Two-Wheeled Self-Balancing Robot with Dynamics Model” *Proceedings of the 2017 4th International Conference on Advances in Electrical Engineering (ICAEE),*28-30 September, Dhaka, Bangladesh.
7. Magdi S. Mahmoud and Mohammad T. Nasir “Robust Control Design of Wheeled Inverted Pendulum Assistant Robot” *IEEE/CAA JOURNAL OF AUTOMATICA SINICA*, VOL. 4, NO. 4, OCTOBER 2017.
8. P. FRANKOVSKÝ, L. DOMINIK, A. GMITERKO, I. VIRGALA P. KURYLO O. PERMINOVA, “Modeling of Two-Wheeled Self-Balancing Robot Driven by DC Gearmotors” *Int. J. of Applied Mechanics and Engineering*, 2017, vol.22, No.3, pp.739-747.
9. Keerthi Prakash, Koshy Thomas “Study of Controllers for a Two Wheeled Self-balancing Robot” 2016 *International Conference on Next Generation Intelligent Systems (ICNGIS)*.
10. WANG Xin, CHEN Songlin\*, CHEN Ting and YANG Baoqing “Study on control design of a two-wheeled self-balancing robot based on ADRC” *Proceedings of the 35th Chinese Control Conference* July 27-29, 2016, Chengdu, China.
11. Congying Qiu and Yibin Huang “The Design of Fuzzy Adaptive PID Controller of Two-Wheeled Self-Balancing Robot” *International Journal of Information and Electronics Engineering*, Vol. 5, No. 3, May 2015.
12. Fengxin Sun , Zhen Yu, Haijiao Yang “A Design for Two-Wheeled Self-Balancing Robot Based on Kalman Filter and LQR” 2014 *International Conference on Mechatronics and Control (ICMC)* July 3 - 5, 2014, Jinzhou, China.
13. Liangliang Cui, Yongsheng Ou, Junbo Xin, Dawei Dai, Xiang Gao “Control of a Two-Wheeled Self-Balancing Robot with Support Vector Regression Method” *2014 IEEE*.
14. Osama Jamil, Mohsin Jamil, Yasar Ayaz, Khubab Ahmad, “Modeling, Control of a Two-Wheeled Self-Balancing Robot” 2014 *International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)* Islamabad, Pakistan, April 22-24, 2014.
15. Hau-Shiue Juang and Kai-Yew Lum “Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino Microcontroller Board”, 2013 10th *IEEE International Conference on Control and Automation (ICCA)* Hangzhou, China, June 12-14, 2013.
16. Wei An¹ and Yangmin Li¹²\*, Senior Member, IEEE, “Simulation and Control of a Two-wheeled Self-balancing Robot”, *Proceeding of the IEEE International Conference on Robotics and Biomimetics (ROBIO)* Shenzhen, China, December 2013.
17. Junfeng Wu, Yuxin Liang, Zhe Wang, “A Robust Control Method of Two-Wheeled Self-Balancing Robot” 2011 The 6th *International Forum on Strategic Technology*.
18. Tao Feng, Tao Liu, Xu Wang, , Zhao Xu, Meng Zhang, Sheng-chao Han, “Modeling and Implementation of Two-wheel Selfbalancing Robot Equipped With Supporting Arms” *2011 IEEE*.
19. David Morin, “*Introduction to Classical Mechanincs with Problems and Solutions*”, New York: Cambridge University Press, 2008.

[20] H. D. Young, R. A. Freedman, “*University Physics with Modern Physics*”, Twelfth Edition. San Fransisco: Pearson Adisson-Wesley, 2007.

[21] Vinod Kumar P, Dr. N Kamala, “Design, Dynamic Modelling, and Control of a Two-wheeled Self-Balancing Robot (TWSBR)”, International Journal of Mechanical Engineering, , Vol. 7 (Special Issue 5, April-May 2022) pp 650-668.

[22] Vinod Kumar P, Dr. Kamala N, “Study of Path Finding and Constraints of Two-wheeled Mobile Robots”, International Journal of Mechanical Engineering (7) Special Issue 5, 501-506, 2022

[23] Vinod Kumar P, Dr. Kamala N, “Design of Robust Controller for Two-wheeled Inverted Pendulum System”, Elsevier Journal: Materials Today: Proceeding. (58) Part-1, 191- 198, 2022.

[24] Cortés-Antonio, Prometeo, et al.(2020) "Learning rules for Sugeno ANFIS with parametric conjunction operations." *Applied Soft Computing* 89: 106095.

[25] Nguyen, Duc-Minh, Nguyen Van-Tiem, and Trong-Thang Nguyen.(2021) "A neural network combined with sliding mode controller for the two-wheel self-balancing robot." IAES International Journal of Artificial Intelligence (10).3 : 592-601.