The BNO055 IMU sensor, integrated into the Arduino motor carrier board, is utilised to capture the angular displacement and angular acceleration crucial for the implementation of the balancing control algorithm. To monitor the bike's orientation, a selector block is employed to extract the y-axis orientation data from the sensor. To ensure that the balancing algorithm can rapidly react to changes in the bike's orientation, no filtering is applied to the sensor data. While filtering can help smooth out noise, it also introduces a delay in signal processing, potentially slowing the bike's response to orientation changes caused by shifts in inertia. Additionally, a bias value is introduced to the angular displacement data to counteract any inherent sensor bias and improve accuracy. To calibrate the sensor and refine the balancing algorithm, the bike is initially stabilised in an upright equilibrium position using a plastic support. This setup allows for the identification and adjustment of any errors in the angular displacement readings. Specifically, the observed error is used to fine-tune the bias, ensuring that the sensor's zero-degree point aligns with the bike's true equilibrium position.

The decision-making process for the execution of the balancing control mechanism in the system relies heavily on precise calibration of the Inertial Measurement Unit (IMU) sensor, which comprises a gyroscope, accelerometer, and magnetometer. The IMU sensor within the BNO055 module reaches full calibration when the calibration values for each vector attain a value of 3. Calibrating the IMU sensor required balancing mode to be switched off, holding the bike still with the plastic mount for 10 seconds, then rotating the bike in all 3 axes slowly until the IMU status responded fully calibrated. In addition to the IMU's calibration status, the system also monitors the battery voltage through the motor carrier. The control mechanism for balancing is permitted to engage only when two conditions are simultaneously met: the battery voltage exceeds 3.0 volts, indicating sufficient power levels, and the IMU sensor is fully calibrated, assuring reliable orientation and motion data.

The control system's initial setup incorporates the use of the motor's rotary encoder to determine the rotational speed, which is calculated by dividing the encoder's count by a ratio relative to π. To address the challenge of signal noise, which can significantly affect the accuracy of speed measurements and subsequently the performance of the control system, a low-pass filter is employed. This filter effectively smooths out high-frequency noise, ensuring that only relevant speed data influences the control decisions. Initially, this measurement was incorporated into the balancing algorithm as a feedback component. However, due to the system's high sensitivity to this parameter, which often led to instability, it was decided to exclude this metric from the algorithm. Eliminating this parameter resulted in a more stable and predictable system performance.

However, the system's operational integrity also accounts for the bike's physical orientation. If the bike tilts beyond a predefined safe threshold range of angular displacement, indicating a potential fall or unsafe operating condition, the system is designed to halt promptly. This immediate stop is a safety measure, preventing the balancing inertia wheel from causing harm to the bike or to the surroundings. It is a crucial feature that ensures the bike does not continue to attempt to balance in a scenario where it could lead to further instability or damage.

In preparation for designing a controller for the bike, a foundational understanding of the dynamics involved was established through simulation models. A Full-State Feedback controller was developed to balance the bike utilising the inertia wheel, incorporating feedback from all system states: angular displacement, angular velocity, and the integral of angular displacement. Each parameter was scaled using gains, with the system's behaviour monitored through scopes, starting with the proportional gain for theta. The simulation graphs indicated that introducing this gain caused the flywheel to oscillate in the upright position. A parameter sweep illustrated a direct correlation between the proportional gain and the oscillations observed within the simulation. The objective was to reduce these oscillations to zero, a task that presented challenges, as the gain needed to remain sufficiently high to provide the torque necessary for stabilising the bike. The derivative gain was then applied as a dampening factor to angular velocity, effectively minimising unwanted oscillations. Through continuous iterative tuning, it was observed that reducing the proportional gain improved the system's response time but reduced precision. The tuning process was finalised by adjusting the inertia wheel's angular velocity with another proportional gain. This meticulous tuning of all three gains allowed the bike to achieve balance with minimal oscillations, demonstrating the intricacy and precision required in the control system's design and calibration for optimal performance.

The simulated control system is then deployed to the controller on the bike hardware, clear objectives for performance benchmarks were established to guide the development process. The primary goal was to enable the bike to maintain balance for extended periods, achieving consistent stability when stationary. Furthermore, it was essential for the controller to maintain the bike's upright position even when subjected to external noise, ensuring robustness against disturbances. Lastly, the bike was required to have the capability to drive in a straight line, execute steering manoeuvres, and move in reverse. The tuning of the controller was meticulously carried out with these performance benchmarks in mind, ensuring that the final system met all the defined criteria for stability, noise resilience, and manoeuvrability.

A full state feedback controller with PID control shown in Figure 3 was implemented to balance the bike using the inertia flywheel. This approach enables precise management of the bike's stability by adjusting the control actions based on the feedback received from the system's current state. Utilising a feedback controller offers significant advantages, including the ability to respond dynamically to disturbances and deviations from the desired state, thus enhancing the system's robustness and adaptability. Furthermore, feedback control facilitates the achievement of desired performance and stability criteria, even in the presence of uncertainties and external changes. The angular displacement of the wheel relative to the equilibrium position serves as the error value for the proportional control parameter, which dictates the response's magnitude to the error. This strategy ensures that the control efforts are directly proportional to the deviation, promoting a swift return to equilibrium. The derivative of the error, although not fully elaborated upon, plays a crucial role in predicting the system's future behaviour, thereby enabling anticipatory adjustments to the control actions, enhancing the system's responsiveness and stability. The integral component of the PID controller is added to improve the controller architecture, addressing any residual steady-state error that the proportional and derivative components cannot eliminate. This integration ensures that even minor discrepancies between the current state and the desired state are corrected over time, leading to a more accurate and stable control outcome. The Ziegler-Nichols method is used for tuning the PID gain values to achieve minimal oscillations in bike balancing. This approach commenced with the determination of the system's ultimate gain and oscillation period by incrementally increasing the proportional gain until the bike exhibited consistent, sustained oscillations. The PID gains were initially set to the default values of [80, 0.01, 10]. These values were incrementally adjusted, using a parameter sweep approach, until the bike displayed consistent behaviour, resulting in a new setting of [83, 0.01, 10]. Following this, fine-tuning of the derivative and integral gains was undertaken to optimise the system's response time, aiming for a quick reaction while maintaining minimal overshoot and undershoot. The tuning process and test result of response is documented in Table 3. This approach yields an aggressive gain setting, facilitating the bike's rapid attainment of a balanced state; however, it also introduces a minor overshoot. Despite this, the slight overshoot is deemed acceptable, as it does not compromise the bike's overall balance. Consequently, the parameters selected, as detailed in the Figure 3, reflect a strategic compromise, optimising the bike's response time while maintaining its stability.

The optimal number of inertia flywheels for balancing the bike was determined through testing configurations with one, two, or three discs, focusing on the interplay between inertia and torque. Inertia, crucial for resisting changes in rotational motion, depends on the flywheel's mass and radius, influencing the generated counter-torque essential for stabilisation, as the magnitude of torque is directly proportional to the magnitude of inertia, and the magnitude of inertia is proportional to mass ($I = k \cdot m \cdot r^2, \tau = I \cdot \alpha$). A single disc proved insufficient in providing the necessary inertia for effective balance, the bike tilted over and failed to stabilise as the wheel cannot provide enough torque. The three-disc setup emerged as the optimal choice compared to two-disc setup, offering a balanced compromise between generating sufficient counter-torque for stability and maintaining a manageable weight, resulting to be faster to response compared to two-disc for every test conducted in Table 2, thereby optimising the bike's overall performance and manoeuvrability without excessive energy consumption.

To further evaluate the stability of the PID controller, additional noise was introduced during the testing phase. Additionally, the starting angle of the bike was varied across five tests, with each test increasing the starting angle by 5 degrees. The results of these tests are documented in Table 1. In every instance, the bike successfully returned to a balanced and stable state within a maximum time frame of 6 seconds. It was observed that the bike is capable of regaining balance from a maximum tilt angle of 25 degrees. This demonstrates the robustness of the PID controller in handling disturbances and its effectiveness in restoring equilibrium under varied initial conditions.

Following the successful balancing of the bike, a feed-forward locomotion controller was developed to initiate movement. The first step involved integrating a "M1 M2 DC Motor" controller block into the motorcycle's hardware model. This addition enables the control system to transmit voltage signals directly to the pins connected to the motor, thereby initiating locomotion. To safeguard the motor against potential damage, a saturation block limits the maximum input voltage. Furthermore, a stopwatch is used to calculate the bike's velocity in relation to the applied voltage input. It was observed that the bike's stability improves with an increase in velocity, yet it becomes prone to tilting beyond a certain velocity threshold. Through extensive testing, a safe velocity range was established, leading to the setting of a reasonable minimum and maximum saturation voltage between 30 and 67 volts. This careful calibration ensures the motor operates within a safe and efficient range, optimising the bike's performance while preventing damage.

The open-loop locomotion controller is detailed in Figure 2. It utilises a slider mechanism to regulate the input voltage delivered to the motor. Despite the advantages offered by closed-loop systems in terms of adaptability and precision, the choice was made in favour of an open-loop system. This decision was based on several key factors. Primarily, the open-loop system's simplicity and lower complexity make it more accessible and easier to implement, especially in scenarios where rapid development and deployment are crucial. Additionally, the operational environment or application for which the bike is designed may not necessitate the high level of adaptability that closed-loop systems provide. For applications with predictable conditions and less variability, an open-loop system can efficiently fulfil the requirements without the additional complexity and cost associated with sensors and feedback mechanisms. A similar control system was adapted for managing the steering servo motor, ensuring that as the bike moves forward, the servo motor automatically adjusts to a default position. This adjustment counteracts any unintended steering of the front wheel, a phenomenon typically induced by the inertia from the balancing wheel. Carrying out a system response test using a sinusoidal input resulted in the bike moving forwards and backwards.

One notable improvement being the application of Root-Locus analysis to enhance system stability. By modelling the bike as an inverted pendulum plus flywheel model (IPFM) and linearizing this model using Equation 1, a foundational step is established. Following this, the determination of the system's transfer function allows for the identification of its poles and zeros. With this information, a Root-Locus plot can be generated using MATLAB, providing a visual and analytical means to assess the system's stability. The Root-Locus method not only aids in stabilising the system but also enhances the understanding of the dynamics at play, guiding the design towards achieving desired stability and response characteristics. Another enhancement could be the integration of a supervisory locomotion system. This can be achieved by utilising the rear DC motor's rotary encoder. Converting the encoder count readings into revolutions and then translating them into angular distance, theta, finally converting theta into linear distance. An essential part of this enhancement involves implementing a conditional logic, specifically an if statement, designed to cease the operation of the DC motor once the bike reaches a predetermined location.

At the conclusion of the project, the bike demonstrated the capability to move forward more than 2 metres while maintaining balance at a velocity of 1.12 m/s, and it showcased the ability to stay balanced on the ground for more than 1 minute. This performance is indicative of a robust and well-calibrated control system that effectively integrates feedback from the bike's sensors to maintain stability and forward momentum under varying conditions. Despite meticulous design and testing, the control system and hardware model were subject to persisting uncertainties. A notable issue emerged with the DC motor overheating during extended operation periods, adversely affecting its performance. Furthermore, the bike's performance was significantly influenced by battery power levels; as the battery drained, the bike's behaviour varied noticeably from the beginning to the end of the sessions. Manufacturing discrepancies among the bikes also presented challenges, necessitating individual PID tuning for each bike to achieve optimal performance. Given these variables, the development of an auto-tuning PID system becomes imperative.

Appendix

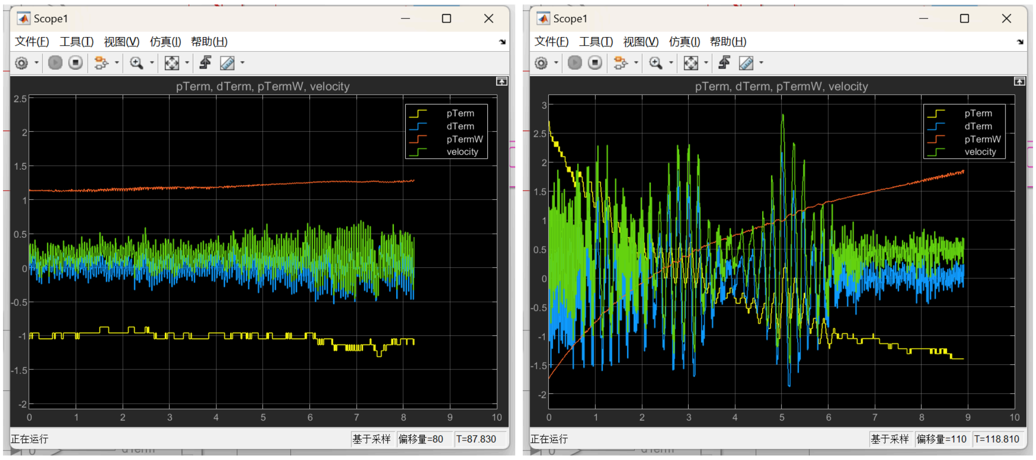


Figure 1:(L) PID and Velocity Curve After Tuned (R): PID and Velocity Curve with start noise

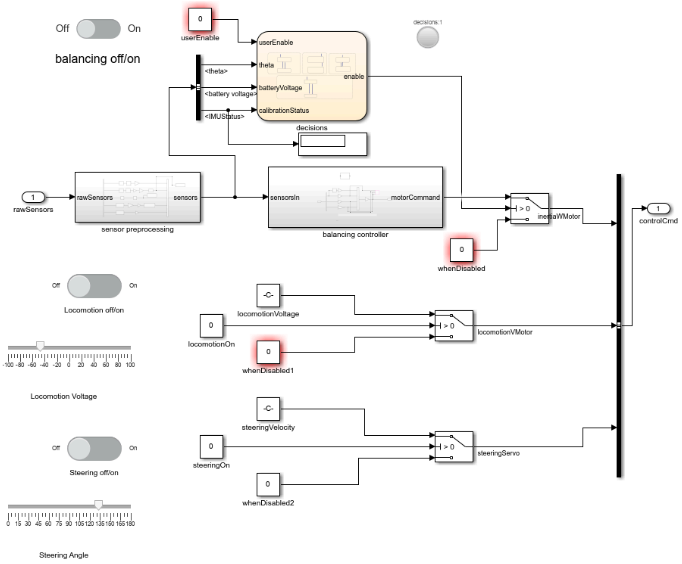


Figure 2: Control System Overall

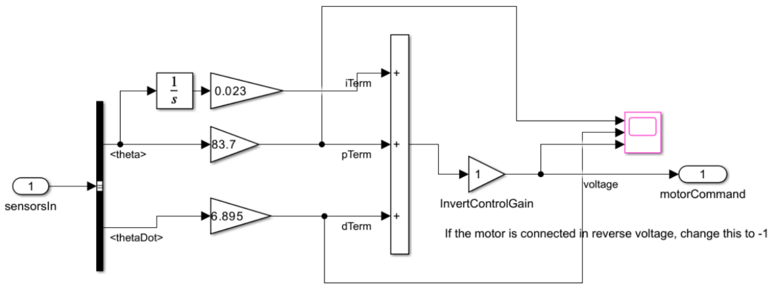


Figure 3: PID Controller

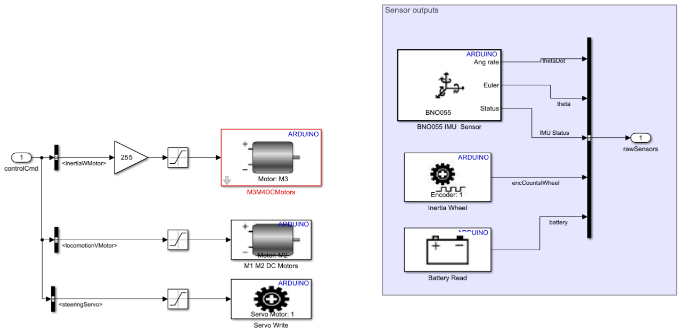


Figure 3: Hardware Model

Start Angle (deg)	Time take to be balanced
0	1
5	1.67
10	2.35
15	3.56
20	5.12
25	6

Table 1: Noise Test

Flywheel Disc Count	Mass (g)	Time take to be balanced (s)
1	30	Failed
2	60	7
3	90	4.3

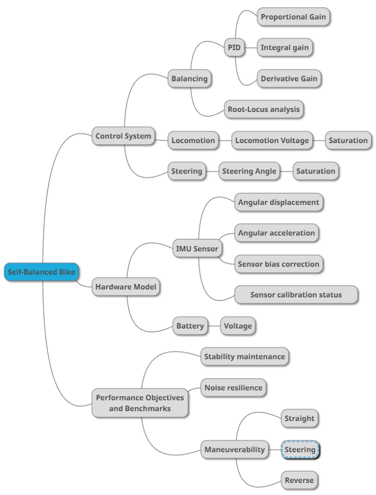
Table 2: The Bike's Response Time vs Flywheel Disc Count

P	I	D	Time take to be balanced (s)	Undershoot %	Overshoot %
80	0.01	10	8	3	0.78
82	0.01	10	9.25	3.32	0.33
83	0.01	10	10.7	4.5	0.85
83.5	0.02	9	9.8	8.5	0.45
84	0.03	7	Not Balanced		
83.6	0.02	6	8.87	8.5	0.78
83.7	0.023	6.5	7.6	7.5	0.55
83.7	0.023	6.895	6.35	5.56	0.56

Table 3: PID Tuning and Response of the System

$$m\ddot{x} = \frac{mg}{z_c}x - \frac{1}{z_c}\tau_F \qquad I\ddot{\theta}_F = \tau_F$$

Equation 1: Linear IPFM Equation



Design Parameters