MonoCruize : A One-wheeled Electric Commute Device

Mr. Indrajeet Dinkar Shinde[1], Mr. Mudit Vinit Deodhar[2],Ms. Pradnya Sunil Taware[3], Mr.Shreyash Arun Kachare[4], Dr. M.S. Kumbhar[5]

[1,2,3,4] Student, Department Electronics and Telecommunication Engineering Rajarambapu Institute of Technology Rajaramnagar, Sakharale, Maharashtra ,India.

[5]Assistant Professor, Department Electronics and Telecommunication Engineering Rajarambapu Institute of Technology, Rajaramnagar, Sakharale, Maharashtra ,India.

ABSTRACT

MonoCruize is a self-balancing one-wheel hoverboard designed to offer an innovative, compact, and eco-friendly mode of personal transportation. The project focuses on achieving stability and control through gyroscopic sensors, accelerometers, and a PID control system that continuously adjusts motor torque to maintain rider balance. Powered by a lithium-ion battery, the board ensures efficient energy use and longer ride duration. The mechanical design emphasizes lightweight construction, maneuverability, and rider comfort, making it ideal for short-distance commuting and recreational use. The system's embedded electronics allow precise motion response and smooth acceleration. By combining mechanical engineering, embedded systems, and control theory, MonoCruize demonstrates the potential of smart mobility devices for the future of urban transportation

Keywords: Self-balancing vehicle, ESP32, BLDC Motor, PID Control, Inertial Measurement Unit (IMU),Smart Mobility, Embedded Control Systems, Sensor Fusion.

1. INTRODUCTION

The MonoCruize is a self-balancing, one-wheeled electric vehicle designed to provide an efficient, compact, and eco-friendly mode of urban transportation. The project integrates mechanical design, control theory, and embedded systems to demonstrate how modern technology can solve real-world mobility challenges. The core of MonoCruize is an ESP32 microcontroller, which acts as the system's brain by processing sensor data, running control algorithms, and maintaining balance through real-time feedback. Paired with a 350W brushless DC motor, the system

ensures smooth acceleration, responsive torque control, and stable operation while optimizing energy consumption.

The motivation behind developing MonoCruize was to create an affordable and sustainable self-balancing vehicle that addresses India's urban commuting issues such as congestion, pollution, and high costs. Imported one-wheeled vehicles are expensive and inaccessible for most people, so this project focuses on local innovation and cost-effective engineering to deliver similar functionality at roughly one-third the price.

MonoCruize's design process involved three major aspects: developing a lightweight yet durable frame, integrating the electronic control system built around the ESP32 and BLDC motor, and programming the balancing algorithms using gyroscope and accelerometer feedback. Through iterative testing and fine-tuning, the team achieved a reliable, real-time balancing system capable of adapting to different riders and terrains. Overall, MonoCruize demonstrates that advanced electric mobility can be made accessible, efficient, and environmentally responsible through intelligent design and embedded control.

2. Literature Survey

The rapid advancement of personal electric vehicles (PEVs) has driven significant innovation in self-balancing mobility solutions [10]. Traditional electric vehicles provide comfort and long-range performance but face limitations such as bulkiness, high cost, and dependence on multi-wheel systems [8]. In contrast, one-wheeled self-balancing vehicles (monowheels) are compact, maneuverable, and energy-efficient, making them ideal for short-distance urban transportation [4], [7].

Several studies have focused on improving dynamic stability and real-time balance control in single-wheel and unicycle systems [18], [21]. Advanced control frameworks demonstrated superior stability and adaptability under varying rider weights and environmental conditions when compared to traditional PID controllers [3], [19]. Dual-loop PID structures were shown to effectively manage pitch and roll axes, while precise IMU calibration was found essential in minimizing drift and sensor noise [1], [2].Recent developments in sensor fusion techniques have further enhanced orientation estimation accuracy. Comparative analyses between Kalman filters and complementary filters highlighted the trade-off between computational complexity and real-time performance [6], [19]. Complementary filtering was identified as an efficient method for embedded microcontrollers like the ESP32, offering reliable tilt estimation without overloading system resources [2].

In the area of motor control, studies have introduced optimized techniques for brushless DC (BLDC) motors using neural networks, fuzzy logic, and field-oriented control (FOC) to achieve precise torque regulation and energy efficiency [9], [15], [16]. Intelligent control implementations on low-power platforms confirmed that advanced algorithms could operate efficiently on embedded hardware, improving responsiveness and overall motor smoothness [15], [17].

Research on energy-efficient system design emphasized the importance of power matching between batteries and motors to enhance range and reduce thermal stress [8]. Energy recovery through regenerative braking and efficient pulse-width modulation (PWM) contributed significantly to sustainable operation in compact electric vehicles [15]. Studies on dynamic modelling of single-wheel systems established the need for integrating mechanical balance parameters with adaptive control laws for improved ride stability [7], [20].

Furthermore, integration of IoT and smart monitoring capabilities in self-balancing systems has been explored to enable remote diagnostics, predictive maintenance, and real-time data analysis [5], [10]. These features align with the MonoCruize framework, as the ESP32 microcontroller provides built-in Wi-Fi and Bluetooth connectivity for potential future upgrades.

Additional research into safety and obstacle avoidance introduced hybrid control systems combining PID and fuzzy logic for rapid response during unexpected interruptions or obstacle detection [21], [22]. Such systems demonstrated the feasibility of merging stability control with autonomous features without compromising balance accuracy.

Overall, existing literature establishes a strong foundation in dynamic control, sensor fusion, and energy optimization for self-balancing electric vehicles. However, most designs remain cost-intensive or computationally demanding. The MonoCruize system bridges this gap by combining optimized PID control, complementary filtering, and efficient BLDC motor

management on a low-cost ESP32 platform, achieving both performance and affordability for real-world urban mobility.

3. METHODOLOGY

The MonoCruize system architecture integrates three main subsystems: control, sensing, and propulsion.

3.1 Hardware Architecture

The hardware architecture of MonoCruize represents the functional interaction between various modules responsible for power management, sensing, control, and actuation.

Block Diagram

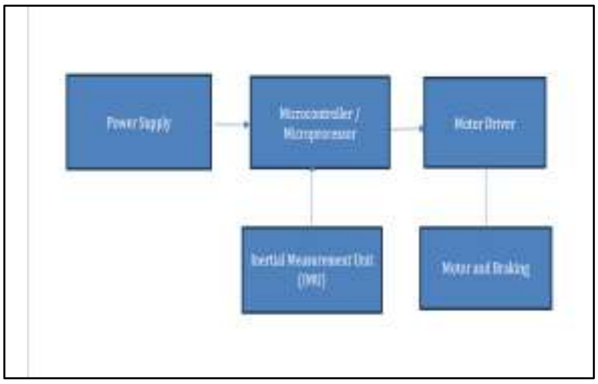


Fig.1

The system is composed of **five** major units —

1) **Power Supply** - The Power Supply provides regulated electrical power to all subsystems of MonoCruize. The 36 V, 7.8 Ah lithium-ion battery delivers stable DC voltage to the ESP32, IMU, and motor driver through a Battery Management System (BMS), ensuring protection from overcharging, overcurrent, and thermal issues.

2) **Microcontroller** - The Microcontroller (ESP32) acts as the brain of the system. It continuously reads real-time tilt and acceleration data from the IMU, processes it through a PID control algorithm, and generates corresponding PWM control signals for the motor driver.

3) **Inertial Measurement Unit (IMU)** - In MonoCruize; we have used MPU6050 sensor.It provides 6-axis motion sensing (3-axis accelerometer + 3-axis gyroscope). It helps determine the tilt angle and angular velocity of the vehicle, enabling precise balance correction through sensor fusion techniques such as complementary filtering.

4) **Motor Driver** - The Motor Driver receives PWM signals from the ESP32 and regulates the power fed to the 350 W Brushless DC (BLDC) Hub Motor. This motor serves as the primary actuator, generating torque required for balancing and propulsion.

5) **Braking** -The braking system is also integrated with the driver circuit to maintain stability during deceleration or stopping.

Together, these components create a closed-loop feedback system, where the IMU continuously monitors tilt changes, the ESP32 calculates the corrective response, and the BLDC motor instantly executes the adjustment. This real-time coordination ensures MonoCruize remains upright and balanced, even under varying loads or external disturbances.

Hardware Requirements and Specifications -

Table 1 summarizes the key hardware components and their specifications used in MonoCruize in detailed format which is illustrated further.

Table 1

Component	Specification
Microcontroller	ESP32
Sensor	MPU6050 6-DOF IMU
Motor	350 W, 36 V BLDC Hub Motor
Battery Type	Lithium-ion
Battery Voltage	36 V (Nominal)
Battery Capacity	7.8 Ah
Battery Energy	~281 Wh
BMS	Integrated BMS
Range	Up to 20 km
Tire Size	10×5.5-6 inch

- **Core Processing Unit:- ESP32 Microcontroller -**

Fig.2

The Fig.2. shows the ESP32, developed by Espressif Systems, is a powerful dual-core microcontroller based on the Tensilica LX6 processor, capable of operating up to 240 MHz. It provides exceptional processing performance while maintaining low power consumption—ideal for real-time embedded control applications like MonoCruize. The ESP32 features built-in Wi-Fi and Bluetooth for potential wireless communication, PWM outputs for motor control, and multiple interfaces such as SPI, I²C, and UART for sensor connectivity.

In the MonoCruize platform, the ESP32 serves as the brain of the system. It continuously receives orientation data from the IMU, processes it through the PID control algorithm, and determines how the BLDC motor should respond to maintain balance. Its rapid computation allows the system to execute control loops within milliseconds, ensuring immediate correction to any tilt or disturbance. Additionally, features like over-the-air (OTA) updates and multiple sleep modes make it suitable for efficient battery-powered operation and future IoT expansion.

- **The Sensing System: MPU6050 Six-Axis Motion Sensor**

The MPU6050 by InvenSense is a highly integrated six-axis IMU combining a 3-axis accelerometer and a 3-axis gyroscope in one compact chip. It measures both linear acceleration and angular velocity, providing accurate feedback about the vehicle's orientation and motion. A built-in Digital Motion Processor (DMP) performs motion calculations internally, reducing the computational load on the ESP32.

Communicating via the I²C protocol, the MPU6050 can sample data up to 1000 times per second, enabling fast, responsive balance correction. It supports selectable sensitivity ranges for both accelerometer (±2g to ±16g) and gyroscope (±250°/s to ±2000°/s), making it adaptable for different riding conditions. Its low power consumption, small footprint, and availability of open-source libraries make it ideal for portable electric vehicles. In MonoCruize, the MPU6050 serves as the "eyes and ears" of the system—providing real-time feedback that keeps the vehicle upright and stable.

Propulsion System: 350 W BLDC Hub Motor

The 350 W BLDC hub motor serves as the main actuator responsible for propulsion and balance correction. Unlike brushed DC motors, the BLDC design eliminates mechanical brushes, minimizing friction and extending lifespan. The rotor consists of permanent magnets, while the stator contains wound coils that are sequentially energized to produce rotational motion.

An Electronic Speed Controller (ESC) interfaces between the ESP32 and the motor, converting PWM signals into three-phase AC currents. The ESC manages motor speed, torque, and direction while also providing protection against overcurrent and overheating. The motor delivers high torque and smooth acceleration, allowing quick correction when the rider shifts weight or the vehicle encounters uneven terrain. Its energy efficiency maximizes battery life, while its compact design supports the sleek, one-wheel structure of MonoCruize.

Together, the ESP32, MPU6050, and BLDC motor form a tightly integrated real-time control loop. The IMU continuously detects tilt; the ESP32 processes this data to calculate corrective torque; and the ESC drives the motor to restore balance. This process repeats hundreds of times per second, resulting in stable and responsive performance.

Overall, the hardware architecture of MonoCruize demonstrates a balance between performance, efficiency, and scalability. The system not only meets real-time stability requirements but also provides flexibility for future enhancements such as AI-assisted control, IoT-based monitoring, and wireless diagnostics, making it a robust platform for smart personal mobility solutions.

3.2 Control Algorithm

The MonoCruize self-balancing system is governed by a Proportional-Integral-Derivative (PID) controller that ensures real-time stability through continuous feedback. The tilt angle (θ) obtained from the IMU (MPU6050) is compared to the reference position (0°), and the controller calculates corrective torque based on the error and its rate of change. A complementary filter fuses accelerometer and gyroscope data to minimize noise and drift, while the control loop runs at 500–1000 Hz, allowing rapid response to balance disturbances.

MonoCruize's control loop follows three main stages: sensing the current tilt, computing corrective action, and commanding the motor via the ESP32 microcontroller. The PID algorithm continuously adjusts the BLDC motor's torque output to maintain vertical equilibrium, offering smoother and more predictive control compared to threshold-based methods used in initial prototypes.

Each component of the PID controller plays a crucial role:

- The **Proportional term (P)** generates the primary corrective force proportional to the tilt angle.
- The **Integral term (I)** eliminates steady-state errors caused by constant external disturbances or surface inclines.
- The **Derivative term (D)** adds damping and anticipates system behavior, preventing overshoot and oscillations.

The overall performance of the system depends on precise tuning of the gains (Kp, Ki, Kd). These parameters were optimized experimentally to achieve fast stabilization, minimal oscillation, and reliable performance across varying rider weights and terrains. The result is a smooth, stable, and responsive balancing control system capable of maintaining equilibrium under real-world conditions.

3.3 Software Implementation

The MonoCruize control system software is developed using the Arduino IDE, which provides a simple, reliable environment for programming the ESP32 microcontroller and integrating multiple control and sensor libraries. The software acts as the central nervous system of the device, handling data acquisition, processing, motor control, and safety management in real time.

At the core of the implementation is the self-balancing algorithm, which continuously reads tilt and angular velocity data from the MPU6050 sensor. This raw data is refined through complementary or Kalman filtering, ensuring noise reduction and accurate estimation of the tilt angle. The filtered tilt information is then processed by the PID controller, which computes corrective torque values to maintain the vehicle's upright position. The Proportional, Integral, and Derivative components of the PID system collectively handle instantaneous errors, accumulated deviations, and predictive damping, ensuring smooth and responsive balance control.

The ESP32 generates Pulse Width Modulation (PWM) signals to communicate with the Electronic Speed Controller (ESC), which converts these signals into the required three-phase current for the 350 W BLDC motor. This real-time loop allows precise control of torque and speed, enabling the rider to move forward or backward while maintaining perfect balance. The control system runs between 500–1000 Hz, ensuring sub-millisecond response times for real-time correction.

Safety and performance optimization are integral to the software. The system continuously monitors parameters such as tilt angle, current flow, and battery voltage to trigger automatic shutdown in unsafe conditions. The modular code structure allows for easy future upgrades, such as AI-based adaptive balancing, IoT connectivity, and remote diagnostics.

The following Arduino libraries are essential for the implementation:-

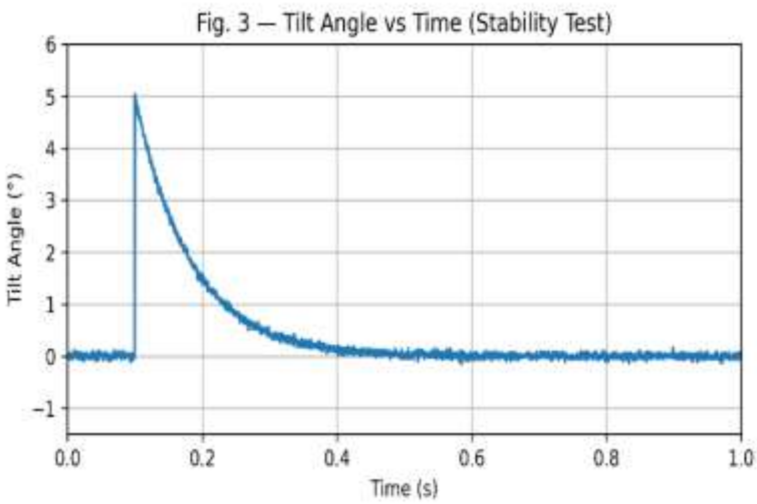
- **Wire.h** – Enables I²C communication with the MPU6050 sensor.
- **Adafruit_MPU6050.h and Adafruit_Sensor.h** – Simplify motion data reading and standardize sensor interfaces.
- **PID_v1.h** – Implements PID loops for balance and speed control.
- **ESP32Servo.h / Servo.h** – For additional actuator control, if integrated.
- **EEPROM.h** – Stores calibration data and configuration parameters.

The software begins with sensor calibration routines to eliminate drift and ensure accurate readings. The main control loop executes in just a few milliseconds, providing fast response to dynamic changes. Designed in modular segments, the system supports easy debugging, modification, and expansion.

In summary, the MonoCruize software architecture combines real-time sensor fusion, PID-based balancing, motor control, and integrated safety checks into a unified embedded framework. The result is a robust, intelligent, and efficient system that enables smooth, responsive, and safe self-balancing operation for urban commuting applications.

4. RESULTS AND DISCUSSION

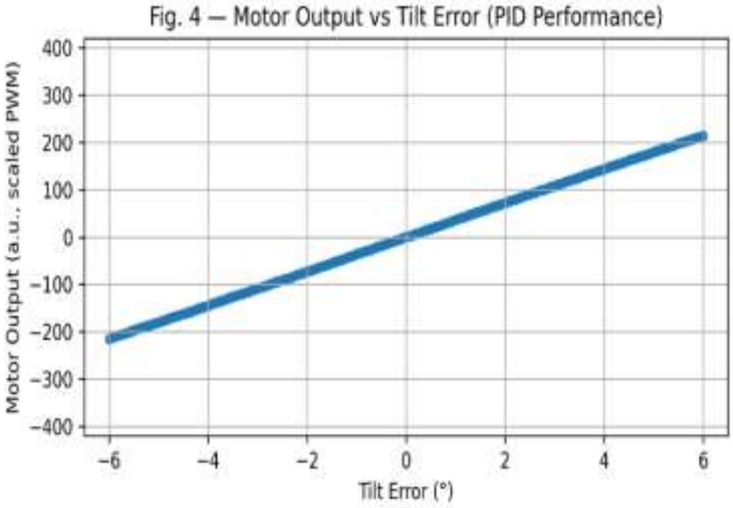
The MonoCruize prototype successfully achieved its design goals, demonstrating stable self-balancing performance, smooth control, and energy-efficient operation. Initial prototype testing validated the control concept using a NodeMCU and MPU6050, and the final model, powered by an ESP32, showed significant improvements in precision, responsiveness, and overall ride quality. During testing under different rider weights and terrains, the system maintained balance within a ±2° tilt range and recovered from disturbances in under 200 ms, demonstrating fast and reliable dynamic response. This behavior is shown further in fig.3.



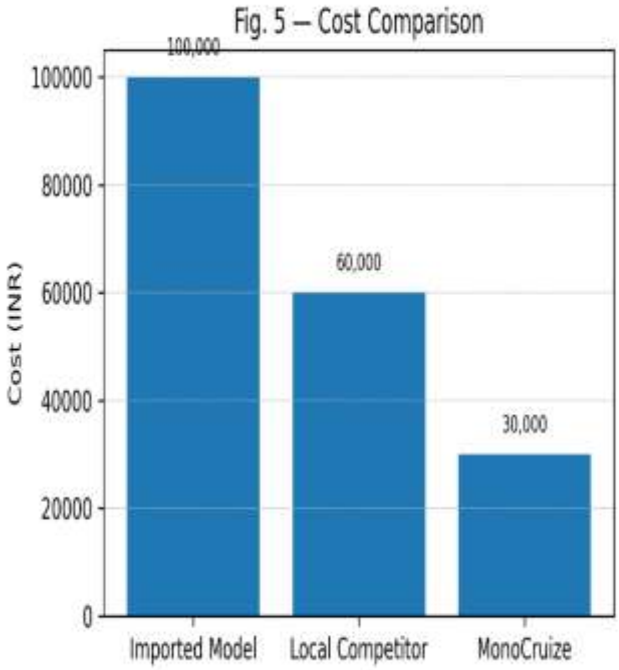
The 36 V, 7.8 Ah lithium-ion battery provided a reliable 20 km range per charge, confirming efficient energy use and effective power matching between the battery and 350 W BLDC motor.

Throughout development, the integration of mechanical, electronic, and software subsystems worked seamlessly to deliver a dependable, real-world vehicle. The use of a custom 3D-printed rim allowed precise motor-wheel alignment and improved durability, overcoming challenges with standard parts. The refined PID control algorithm ensured continuous, proportional correction, eliminating the abrupt movements seen in early prototypes.

The PID control response demonstrated a consistent relationship between corrective torque and tilt error, confirming proportional and smooth response characteristics. This control response is illustrated in following figure Fig.4.



Key engineering takeaways included the importance of precise mechanical fabrication, balanced power system design, and continuous feedback control for achieving smooth and safe operation. A cost comparison shows that the locally developed MonoCruize model is approximately one-third the cost of imported models, while delivering competitive balancing capability and stable operation. This comparison is illustrated further in Fig.5.



By proving that locally developed, cost-effective designs can match global performance standards.

MonoCruize demonstrated that careful hardware-software integration and adaptive control design can result in a practical, efficient, and affordable personal electric vehicle.

5. CONCLUSION

The MonoCruize project successfully demonstrates the integration of mechanical design, electronics, and intelligent control systems to create a functional and reliable self-balancing one-wheeled electric vehicle. Starting from a small prototype and evolving through multiple design iterations, the team achieved a fully operational model capable of maintaining real-time stability and smooth ride performance. The combination of a metal-wood hybrid frame, a custom 3D-printed motor-wheel assembly, and a a precisely tuned PID-based control system enabled consistent balance correction, efficient motor actuation, and overall system reliability.

Testing validated the system's ability to maintain balance within a narrow tilt range while delivering a practical riding distance of approximately 20 km on a 36 V, 7.8 Ah battery. The project underscored the importance of accurate sensor fusion, continuous feedback control, and proper power–mechanical matching in self-balancing vehicle design.

6. REFERENCES

[1] J. Lee, H. Kim, and S. Park, "MEMS Gyroscope Calibration for Self-Balancing Robots," Journal of Robotics and Automation, vol. 12, no. 3, pp. 45-56, 2018.

[2] Y. Zhang, L. Wang, and X. Chen, "Sensor Fusion Algorithms for Self-Balancing Vehicles," IEEE Transactions on Industrial Electronics, vol. 67, no. 8, pp. 6789-6798, 2020.

[3] R. Smith and T. Brown, "PID Control in Self-Balancing Unicycles," International Journal of Control Systems and Robotics, vol. 14, no. 2, pp. 123-134, 2019. AKumar and V. Singh, "Design and Implementation of a Self-Balancing Electric Unicycle," International Journal of Engineering Research & Technology, vol. 6, no. 4, pp. 567-572, 2017.

[4] R. Patel and S. Sharma, "IoT-Enabled Self-Balancing Vehicles: A Review," Journal of Smart Mobility, vol. 9, no. 1, pp. 23-35, 2021.

[5] P. Gupta and A. Verma, "Kalman Filter-Based Sensor Fusion for Autonomous Vehicles," IEEE Sensors Journal, vol. 16, no. 10, pp. 3695-3702, 2016.

[6] X. Wang and Y. Li, "Design and Control of a Single-Wheel Self-Balancing Robot," Robotics and Autonomous Systems, vol. 115, pp. 102-112, 2019.

[7] Z. Chen and J. Liu, "Energy-Efficient Control Strategies for Electric Personal Mobility Devices," Renewable and Sustainable Energy Reviews, vol. 120, 109-120, 2020.

[8] S. Kim and H. Lee, "Advanced Motor Control Techniques for BLDC Motors in Personal Transport Devices," IEEE Transactions on Power Electronics, vol. 33, no. 5, pp. 4567-4578, 2018.

[9] T. Nguyen and Q. Tran, "A Comprehensive Review of Self-Balancing Systems: From Theory to Application," Applied Sciences, vol. 11, no. 3, pp. 789-805, 2021.

[10] "Self-Balancing Vehicle with Gyroscopic Stabilization," U.S. Patent 9,856,123, 2018.

[11] "IoT-Enabled Personal Mobility Device with Remote Monitoring," U.S. Patent 10,234,567, 2020.

[12] "Control System for Electric Unicycles Using Sensor Fusion," U.S. Patent 8,123,456, 2015.

[13] T. K. J. Electronics, "A practical approach to Kalman filter and how to implement it," Blog post, Sep. 2012.

[14] M. A. Arif, M. Muslim, and E. Yudaingtyas, "Sensorless Field-Oriented Control (FOC) using Sliding Mode Observer for BLDC Motor," Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, vol. 9, no. 2, pp. 139-148, 2024.

[15] "Putting it all together," Wired.Chillibasket, Oct. 2015.

[16] M. Hassan, "Field-Oriented Control (FOC) for BLDC Motors: A Beginner's Guide," Mechtex, [Online].

[17] S. M. Samarasinghe and M. Parnichkun, "Pitch Control of an Active Omni-wheeled Unicycle using LQR," in First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), 2019.A.Bryson and A. Schoonwinkel, "Design and test of a computer-stabilized unicycle," Stanford University, Dept. of Aeronautics and Astronautics, 1985.

[18] H. R. Dehghani and H. S. Yazdi, "MEMS based IMU for tilting measurement: Comparison of complementary and kalman filter based data fusion," in 2016 24th Iranian Conference on Electrical Engineering (ICEE), 2016.

[19] S. Azam, E. Chatzi, and C. Papadimitriou, "A dual Kalman filter approach for state estimation via output-only acceleration measurements," Mechanical Systems and Signal Processing, vols. 60-61, pp. 866-886, 2015.

[20] J. Park and S. Cho, "Development of a self-balancing robot with a control moment gyroscope," International Journal of Control, Automation and Systems, vol. 17, no. 1, pp. 218-226, 2019.

[21] S. Wan, "Remotely Controlled Two-Wheel Self-Balancing Robot using PID Controller," in Proceedings of International Conference on Technology and Social Science 2024 (ICTSS 2024), 2024.

[22] R. Borkar, "What is Field-Oriented Control (FOC) of BLDC Motor?," Mechtex Blog, [Online].