STM JOURNALS

Journal of
VLSI Design Tools & Technology

https://journals.stmjournals.com/jovdtt

ISSN: 2249-474X (Online)
ISSN: 2321-6492 (Print)
Volume 16, Issue 1, 2026
January–April
DOI (Journal): 10.37591/JoVDTT

A Review of Two-Wheeled Self-Balancing Robot Using Spartan-3E FPGA for Sensor Fusion and Real-Time Motor Control

Sagar Laxmikant Senad[1,*], Tejaswini Madan Chaudhari[1], Varsha Kshirsagar[2], Prerana Dharmendra Patil[1], Rohan Narendra Patil[1]

Abstract

Two-wheeled self-balancing robots (TWSBR) are a popular application of embedded control and robotics because they operate on the inverted pendulum concept, which is naturally unstable. The main objective of such robots is to continuously maintain balance by estimating the tilt angle and applying corrective motor action in real time. In most practical systems, low-cost inertial sensors such as accelerometers and gyroscopes are used for tilt measurement. However, accelerometer readings are affected by vibration and noise, while gyroscope-based angle estimation produces drift due to integration errors. Therefore, sensor fusion becomes an important requirement for obtaining a stable and reliable tilt estimate. This review paper discusses dynamic modeling methods based on Lagrange formulation and feedback linearization, along with commonly used balancing techniques such as Proportional Integral Derivative (PID) and fuzzy logic control. Complementary filter and Kalman filter-based fusion approaches are reviewed with respect to accuracy, computational complexity, and drift handling. The importance of Field Programmable Gate Array (FPGA)-based implementation is also highlighted, particularly using the Spartan-3E Field Programmable Gate Array (FPGA), since hardware parallelism can reduce delay between sensing and actuation and support efficient Pulse Width Modulation (PWM) generation for motor driving. From the reviewed literature, it is concluded that Spartan-3E-based implementation using complementary filtering and PID control provides a practical and low-cost solution for stable balancing and effective motor control.

Keywords: Complementary filter, inverted pendulum, Kalman filter, PID control, PWM, sensor fusion, Spartan-3E FPGA, two-wheeled self-balancing robot

***Author for Correspondence**
Sagar Laxmikant Senad
E-mail: sagarsenad.rmdstic.entc@gmail.com

[1]Student, Department of Electronics and Telecommunication Engineering, RMD Sinhgad School of Engineering, Warje, Pune, Maharashtra, India
[2]Assistant professor, Department of Electronics and Telecommunication Engineering, RMD Sinhgad School of Engineering, Warje, Pune, Maharashtra, India

Received Date: March 02, 2026
Accepted Date: March 9, 2026
Accepted Date: March 20, 2026

Citation: Sagar Laxmikant Senad, Tejaswini Madan Chaudhari, Varsha Kshirsagar, Prerana Dharmendra Patil, Rohan Narendra Patil. A Review on Two-Wheeled Self-Balancing Robot Using Spartan-3E FPGA for Sensor Fusion and Real-Time Motor Control. Journal of VLSI Design Tools & Technology. 2026; 16(1): 17–27p.

INTRODUCTION

A two-wheeled self-balancing robot (TWSBR) is a robotic platform that maintains an upright position by continuously adjusting the motion of two wheels. The robot can be treated as an inverted pendulum mounted on a moving base. Because the center of gravity lies above the wheel axle, the system is unstable and will fall quickly without continuous feedback and corrective control action. Owing to this challenging behavior, self-balancing robots are widely considered an effective educational and research platform for understanding real-time control systems, sensor integration, embedded hardware design, and robotics [1, 9, 16].

In practical implementation, the performance of balancing mainly depends on (i) accurate tilt angle estimation, and (ii) fast execution of the control loop. Most designs use MEMS-based Inertial Measurement Unit (IMU) sensors, such as MPU6050, which provide accelerometer and gyroscope measurements. Accelerometers provide a good steady-state reference but are disturbed by vibrations and external accelerations during motion. Gyroscopes provide smooth angular velocity signals; however, when integrated to obtain the angle, small biases and errors accumulate, resulting in long-term drift [3]. Therefore, sensor fusion methods are used to combine both signals and improve the accuracy of tilt estimation. Several balancing control strategies, such as Proportional Integral Derivative (PID), Linear Quadratic Regulator (LQR), nonlinear control, and fuzzy logic, have been discussed in existing studies. Among them, PID control is widely used because of its simplicity, stable response, and easy tuning for practical applications [6]. Field Programmable Gate Array (FPGA)-based embedded platforms have also been explored to improve real-time performance and reduce computation delays. The Spartan-3E is one such Field Programmable Gate Array (FPGA) platform that supports hardware parallelism and provides an efficient implementation for Pulse Width Modulation (PWM) generation and motor control [13, 14].

This review paper focuses on the study of inverted pendulum modeling, sensor fusion techniques for tilt estimation, balancing controller design, and Spartan-3E FPGA-based architecture for implementing the complete control loop in real time.

WORKING PRINCIPLE OF TWO-WHEELED SELF-BALANCING ROBOT

The operation of a self-balancing robot can be compared to balancing a stick on a finger, where continuous correction is required to prevent falling. When the robot tilts forward, its wheels must move forward so that the base comes under its center of gravity. Similarly, if the robot tilts backward, the wheels must move in the backward direction. This corrective motion is generated continuously using feedback control, where the sensor data are processed, and the motors are driven in real time [9].

Closed Loop Feedback Control

In the balancing loop:

- The sensor measures tilt angle θ.
- The controller compares it with the reference angle (0 degrees).
- A correction signal is generated.
- Motors are driven using PWM, and the robot maintains balance.

Practical Requirements for Balancing

For proper balancing, the system must satisfy:

- Fast sensor sampling and processing.
- Accurate tilt angle estimation.
- Stable control algorithm with low delay.
- Smooth PWM output for motors.

DYNAMIC MODELING OF INVERTED PENDULUM ROBOT

The dynamic model helps to understand robot motion and control requirements. In the literature, a two-wheeled robot is modeled as a rigid body with wheel dynamics and pitch angle dynamics. The model parameters include the wheel radius, body mass, inertia, and distance of the center of mass from the wheel axis [2].

Lagrange-Based Modeling

A commonly used method is the Lagrange formulation, where the wheel rotation and pitch angle are considered as generalized coordinates:

$q = [\alpha_1, \alpha_2, \beta]^T$ (1)

where α_1 and α_2 are the wheel angles, and β is the body pitch angle.

The overall dynamic equation is expressed as:

$J(q)\ddot{q} + f(q, \dot{q})\dot{q} + G(q) = \tau$ (2)

Figure 1. Inverted pendulum model of the two-wheeled self-balancing robot.

where τ contains the motor torque for the wheels [2]. This model helps design controllers such as PID or nonlinear feedback.

Importance of Modeling in Control

This modeling approach is important because it describes the relationship between body tilt and wheel motion, which helps in selecting the appropriate controller parameters. Although complete nonlinear models provide a better understanding, many practical implementations apply simplified linear approximations for designing PID controllers owing to easier tuning and reduced computational burden. Overall, the model confirms that the robot is highly unstable; therefore, fast control execution with correct sensor feedback is essential for balancing, as shown in Figure 1.

SENSOR SYSTEM AND TILT ANGLE ESTIMATION

Tilt angle estimation is one of the most important aspects of balancing. Most robots use IMU sensors, such as MPU6050, which provide accelerometer and gyroscope measurements [7]. The required tilt angle is usually the pitch angle.

Accelerometer-Based Tilt Angle

The accelerometer provides tilt information based on the direction of gravity. In this study, the accelerometer outputs are represented as $a_x(k)$, $a_y(k)$, and $a_z(k)$, where k indicates the current sample. It is accurate in the steady state but becomes noisy during movement because the robot's

acceleration mixes with the gravity acceleration. Vibrations from motors and wheel motion can also disturb the readings and introduce fluctuations in the estimated tilt angle.

$$\theta_{acc}(k) = tan^{-1}\left(\frac{a_y(k)}{a_z(k)}\right) \qquad (3)$$

Gyroscope-Based Tilt Angle

The gyroscope provides the angular velocity of the robot body, which is smooth and fast for sudden tilt changes. In this study, the gyroscope output is represented as $\omega_{gyro}(k)$, where k denotes the current sampling instant. However, when the angular velocity is integrated to obtain the tilt angle, small biases and measurement errors accumulate over time and cause drift [3].

$$\theta_{gyro}(k) = \theta_{gyro}(k-1) + \omega_{gyro}(k)\, dt \qquad (4)$$

Table 1. Typical complementary filter parameters used for tilt estimation in literature.

Reference work	α	τ (s)	Remarks
Complementary filter on an FPGA platform	0.95–0.99	Application dependent	Low-complexity, suitable for real-time FPGA
Self-balancing robot complementary filter	0.95–0.98	Based on dt	Fast response; reduces drift using accelerometer correction
Hybrid drift compensation approach	Prefilter	Tuned	Improves the short-term estimate before the Kalman update

Need for Sensor Fusion

Because each sensor has limitations, fusion is required. Sensor fusion combines accelerometer and gyroscope signals to generate a stable and accurate tilt estimate for the controller. From the reviewed literature, it is observed that improved tilt estimation reduces oscillations and improves recovery under disturbances during balancing.

COMPLEMENTARY FILTER FOR SENSOR FUSION

Complementary filters are widely used in self-balancing robots because they are simple and efficient. It combines accelerometer and gyroscope data using weights. It is also suitable for FPGA implementation because of its low computation [4, 6].

Complementary Filter Equation

$$\theta(k) = \alpha\, \theta_{gyro}(k) + (1-\alpha)\, \theta_{acc}(k) \qquad (5)$$

where $\theta(k)$ is the filtered tilt angle, $\theta_{gyro}(k)$ is the tilt angle obtained from gyroscope integration, $\theta_{acc}(k)$ is the tilt angle obtained from the accelerometer, and α is the filter constant (typically selected between 0.95 and 0.99) for a stable response.

Here, the gyroscope contribution provides a fast response to sudden changes, whereas the accelerometer contribution helps correct long-term drift. This makes the complementary filter a preferred choice for real-time balancing systems with limited hardware resources, as shown in Table 1.

Advantages

- Easy to implement and tune.
- Fast execution, suitable for real-time systems.
- Reduces the drift effect using accelerometer correction.

KALMAN FILTER AND GYRO DRIFT COMPENSATION

The Kalman filter is another sensor fusion technique that provides better accuracy than the complementary filter. It uses prediction and correction steps with a mathematical model [5]. The Kalman filter also helps in estimating the gyro bias, which reduces long-term drift [3].

Kalman Filter Approach

In gyro drift compensation studies, a complementary filter is sometimes used first, and then the Kalman filter is applied for better correction and stability [3]. This hybrid approach provides improved performance but requires higher computation than complementary filtering.

Kalman Filter Equations for Tilt Estimation

In the Kalman filter-based tilt estimation, the system state is generally defined as the tilt angle and gyroscope bias:

$$x\,(k) = \left[\frac{\theta\,(k)}{b\,(k)}\right] \qquad (6)$$

where $\theta(k)$ is the estimated tilt angle, and $b(k)$ is the gyroscope bias (drift).

The prediction step updates the angle using the gyroscope measurement:

$$\theta^{-}(k) = \theta(k-1) + \omega_{gyro}(k) - b(k-1)\,dt \quad (7)$$

The Kalman gain is computed by:

$$K(k) = P^{-}(k)\,H^{T}\,H\,P^{-}(k)\,H^{T} + R - 1 \qquad (8)$$

The correction step updates the state using the accelerometer tilt measurement

$$x(k) = x^{-}(k) + K(k)[z(k) - Hx^{-}(k)] \qquad (9)$$

Two-Stage Sensor Fusion and Control Flow

In many reviewed self-balancing robot systems, the tilt angle is estimated using a two-stage sensor fusion pipeline. First, the accelerometer and gyroscope outputs were used to calculate the individual tilt estimates. These two values are then combined using a complementary filter to obtain a fast and stable tilt estimate, as shown in Figure 2.

Next, a Kalman filter was applied for better drift compensation and long-term accuracy. In this review, the complementary filter output is considered the measurement input to the Kalman filter:

$$z(k) = \theta_{comp}(k) \quad (10)$$

The final tilt angle after fusion is then used for balancing control.

Comparison of Complementary and Kalman Filter

The complementary filter is simple and fast, whereas the Kalman filter provides a more accurate estimation but requires more computation and tuning. For FPGA implementation, complementary filtering is preferred in many systems because of its lower resource usage and fast response, as shown in Table 2.

CONTROL STRATEGIES FOR SELF-BALANCING ROBOTS

After obtaining the tilt angle, the robot requires a controller that generates motor corrections. Many control techniques have been reviewed in literature.

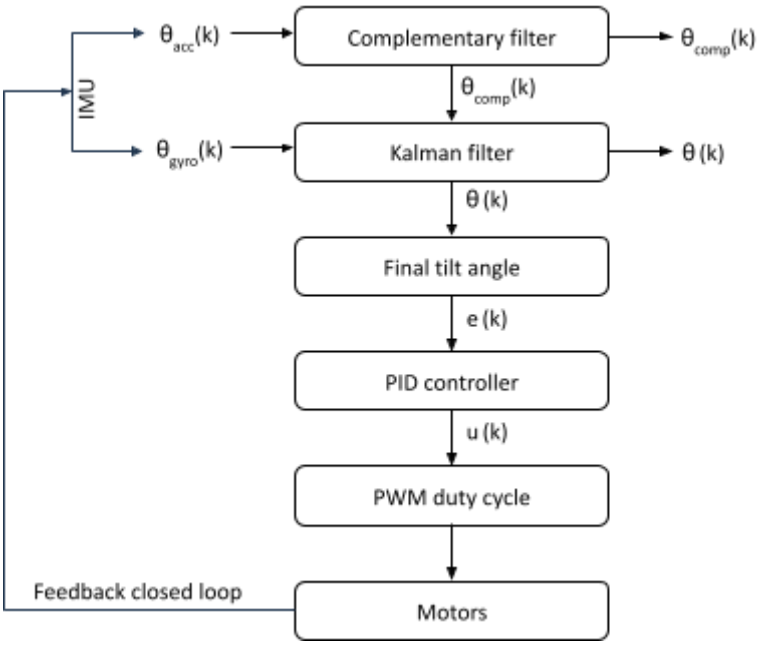

Figure 2. Two-stage sensor fusion and control flow.

Table 2. Complementary filter versus Kalman filter for tilt estimation in self-balancing robots.

Aspect	Complementary filter
Estimation accuracy	Moderate–high
Computation cost	Low
Drift handling	Partial drift reduction
Tuning	Easy (α or τ)
FPGA suitability	Very suitable (low resources)
Best fit for the project	Fast real-time balancing on Spartan-3E

PID Control

A PID controller is commonly used because it is easy to implement and provides a stable output for balancing. The PID parameters are tuned to reduce oscillations and improve the response [6]. FPGA implementations of PID for motor speed control have also been demonstrated [11, 12].

Fuzzy Logic Control

Fuzzy controllers are also used for motor control in real-time systems. FPGA-based fuzzy motor speed controllers have shown good tracking performance [10]. Fuzzy control is useful when system modeling is complex; however, it requires a rule-based design.

LQR and Nonlinear Control

For better optimal performance, LQR and nonlinear feedback linearization methods are used [2]. These methods require state-space modeling and are more complex than PID.

SPARTAN-3E FPGA-BASED IMPLEMENTATION

Real-time balancing requires high-speed data processing. FPGA-based platforms offer parallel execution, making them suitable for implementing filters and controllers in real time.

Why Spartan-3E FPGA?

Spartan-3E FPGA provides:

- Reconfigurable digital hardware logic.
- Parallel execution for filter and control blocks.
- Efficient implementation of PWM generation.

PWM generation and control using the Spartan-3E starter board have been demonstrated using rotary encoder-based methods [13, 14]. This proves that Spartan-3E can effectively handle motor control applications.

Proposed Control Pipeline on Spartan-3E

The proposed architecture includes:

- Reading IMU sensor values.
- Applying a complementary filter to estimate the tilt angle.
- PID control to generate a correction signal.
- PWM generation drives motors for balancing.

FPGA parallelism ensures a minimum delay between sensing and actuation, which improves the stability of the balancing loop.

Motor Control and PWM Generation

PWM generation with a variable duty cycle using a rotary encoder was implemented on a Spartan-3E starter kit [13, 14]. PWM output is a key requirement for DC motor speed control in self-balancing robots.

FPGA-Based PID and Fuzzy Controllers

PID controller implementation using LabVIEW FPGA on Spartan-3E was presented for DC motor speed control [11]. Another implementation used Hardware Description Language (HDL) and Xilinx ISE for PID control [12]. Real-time fuzzy logic motor control also showed better tracking performance than conventional PI control on Spartan-3E FPGA, as shown in Figure 3 [10].

Parallelism Benefit

In a balancing robot, filtering + PID + PWM tasks are continuously executed. FPGA allows:

- Sensor reading and offset calculation in parallel,
- Complementary filter and control computation simultaneously,
- Fast PWM updates with minimal delay.

Sensor Fusion Using Complementary Filter

Figure 3 shows the Spartan-3E FPGA-based sensor fusion and control pipeline. The IMU provides accelerometer and gyroscope data. The accelerometer provides a tilt angle estimate but contains noise, whereas the gyroscope provides a smooth angular velocity but suffers from drift. Therefore, a complementary filter was used to combine both signals and obtain a stable tilt angle for the balancing control.

The complementary filter-based tilt angle estimation is given by:

$$\theta(k) = \alpha\,\theta(k-1) + \omega_{gyro}(k)\,dt + (1-\alpha)\,\theta_{acc}(k) \quad (11)$$

The filtered tilt angle was then applied to the PID controller, and the PWM generator produced motor control signals to stabilize the robot.

NAVIGATION AND ADVANCED CAPABILITIES

Self-balancing robots can be extended for navigation and autonomous movements. ROS-based navigation and simulation environments, such as Gazebo and RViz, have been used to study balancing robot motion [8]. Sensor-based navigation with obstacle avoidance has also been

implemented using IMU and ultrasonic sensors [7]. These features can aid in the development of self-balancing robots as autonomous platforms for indoor applications (Table 3).

HARDWARE REQUIREMENTS AND SYSTEM COMPONENTS

The performance of a two-wheel self-balancing robot mainly depends on the selection of the mechanical and electronic components. In this project, the goal was to implement a stable balancing robot using a Spartan-3E FPGA with sensor fusion and PWM motor control.

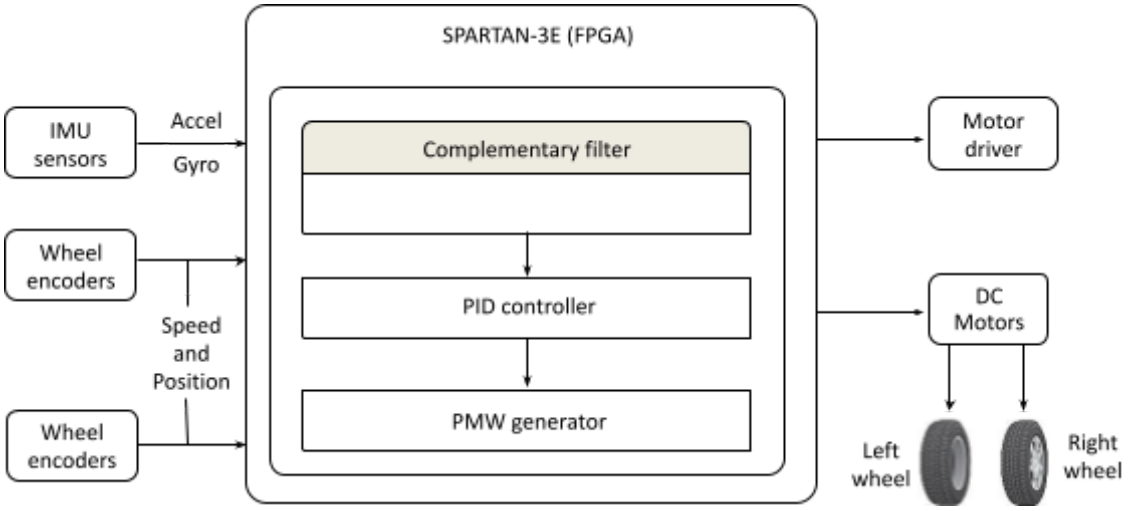

Figure 3. Proposed Spartan-3E FPGA-based architecture for sensor fusion, PID control, and PWM generation.

Table 3. Summary of key research contributions reviewed.

Category	Main contribution
Dynamic modeling	Lagrange-based inverted pendulum model.
Sensor fusion	Complementary and Kalman filter-based estimation methods.
Drift compensation	Gyroscope drift reduction using Kalman filtering.
FPGA control	PID and fuzzy control implemented on Spartan-3E FPGA.
PWM generation	Variable duty cycle PWM generation using Spartan-3E.
Navigation	ROS and sensor-based navigation methods.

Mechanical Components

The mechanical structure should be compact and symmetric to reduce the uneven load on the motors. The key mechanical parts required are as follows:

- Two DC geared motors with equal Revolutions Per Minute (RPM) ratings.
- Two wheels of the same diameter for uniform motion.
- Chassis frame to mount motors, battery, and electronic boards.
- Mounting rods or spacers to fix the FPGA board safely above the motor base.

A balanced mechanical design helps reduce tilting owing to weight imbalance and improves the controller's performance during operation.

Electronic Components

The main electronic components used are:

- *Spartan-3E FPGA board:* Used as the main controller because it supports parallel execution and fast PWM generation.

- *IMU sensor (MPU6050):* Provides accelerometer and gyroscope readings for tilt estimation.
- *Motor driver (H-bridge):* Used to control the direction and speed of DC motors using PWM signals.
- *Power supply:* Battery pack for motors and a regulated supply for the sensor and FPGA board.

Power and Safety Considerations

In practical implementation, separate power lines are recommended:

- The motor driver should be powered by a battery.
- The FPGA and sensor should be powered using a regulated 3.3 V/5 V supply.

This avoids sudden voltage drops caused by motors, which may reset the controller or disturb the sensor readings.

LIMITATIONS AND PRACTICAL CHALLENGES

Although self-balancing robots are an interesting control application, some practical challenges have been observed in most implementations.

Sensor Noise and Vibration Issues

Accelerometer readings are affected by the vibrations of motors and wheels. Sudden movements of the robot can also disturb the accelerometer angle calculation. This may cause fluctuations in the tilt angle and an unstable balance.

Gyroscope Drift

Gyroscope-based angle estimation produces drift owing to integration over time. Even a small bias in the gyro output can generate a large angle error during a long operation. Sensor fusion reduces drift, but tuning is still required for the best results.

PID Parameter Tuning

PID control is simple but requires proper tuning of K_p, K_i, and K_d. If the tuning is incorrect, the robot may oscillate continuously or fall due to overshoot and delay.

Mechanical Imbalance

If the weight distribution is non-uniform, the robot may tilt toward one side. Motor friction and differences in motor speed can also create imbalances. This affects smooth balancing and straight-line movement.

FPGA Implementation Complexity

Although FPGAs provide high speed, designing modules for sensor communication (I2C), filtering, and controllers require more effort than microcontrollers. Implementing floating-point operations is difficult; therefore, fixed-point design is generally preferred in FPGA-based systems.

FUTURE SCOPE

The reviewed works suggest that a Spartan-3E FPGA-based complementary filter and PID controller can achieve stable balancing. However, the system can be further improved by adding more advanced features.

Encoder Feedback for Speed Control

Encoders can be added to both wheels to measure the motor speed. This feedback can improve the straight-line motion and allow for better disturbance rejection.

Advanced Control Techniques

Instead of only PID control, advanced control methods such as LQR, adaptive control, and neuro-fuzzy controllers can be implemented to improve stability and performance under varying load conditions [15].

Kalman Filter Implementation on FPGA

The Kalman filter provides better accuracy for tilt estimation and drift compensation. Future work can include an FPGA-optimized Kalman filter design using fixed-point arithmetic for improved precision.

Wireless Monitoring and Parameter Tuning

Bluetooth or Wi-Fi modules can be used to tune the PID parameters during runtime and remotely monitor the sensor data. This improves the testing and reduces the manual tuning time.

Navigation and Obstacle Avoidance

The self-balancing robot can be extended as an autonomous platform by adding ultrasonic sensors or using camera-based navigation. ROS-based simulation and virtual mapping approaches can also be explored for autonomous movements.

RESEARCH GAP AND PROPOSED REVIEW OUTCOME

From the reviewed literature, it is observed that complementary filtering is widely preferred in self-balancing robot designs because of its simplicity, low computation requirement, and fast response. Kalman filtering improves estimation accuracy and supports drift compensation; however, it increases computational complexity and tuning effort. FPGA-based platforms, such as Spartan-3E, are suitable for real-time balancing because they allow the parallel execution of sensor fusion, control computation, and PWM generation with reduced delay.

Hence, based on this review, the proposed project focuses on:

1. Implementing a complementary filter for stable and fast tilt estimation,
2. Designing a PID controller for real-time balancing control,
3. Generating PWM signals on the Spartan-3E FPGA to drive motors efficiently with improved response.

CONCLUSION

This review presents the working and control requirements of two-wheeled self-balancing robots based on the inverted pendulum principle. Dynamic modeling, sensor fusion techniques, balancing controllers, FPGA-based implementation, and practical challenges were reviewed. Complementary and Kalman filter methods were compared for tilt estimation and drift compensation. Spartan-3E-based complementary filtering with PID control provides a practical and low-cost approach for stable balancing and can be extended further with navigation and advanced control methods.

Declaration of Interest

The authors declare that there are no conflicts of interest regarding the publication of this review paper.

Ethics Approval

This review did not involve human participants or animal experiments. Hence, ethical approval was not required.

Acknowledgements

The authors would like to thank the Department of Electronics and Telecommunication Engineering, RMD Sinhgad School of Engineering, Warje, Pune, for providing guidance and support for completing this review.

REFERENCES

1. Chen J, He N, Xu Z, Dou M, He L. Design and implementation of a novel two-wheeled composite self-balancing robot for stationary operations in unknown terrain. IEEE Access. 2025;13:86032–86045. doi:10.1109/ACCESS.2025.3569325.
2. Cardozo GSS, Vera LMS. Prototype for a self-balanced personal transporter. 2012 Workshop on Engineering Applications, Bogota, Colombia. 2012. p.1–6. doi:10.1109/WEA.2012.6220101.
3. Prongphimai P, Poolperm L, Chaiwas S, Kaewmorakot U. Development of a prototype of self balancing robot for education. 2021 6th International STEM Education Conference (iSTEM-Ed), Pattaya, Thailand. 2021. p.1–4. doi:10.1109/iSTEM-Ed52129.2021.9625106.
4. Lee HJ, Jung S. Gyro sensor drift compensation by Kalman filter to control a mobile inverted pendulum robot system. 2009 IEEE International Conference on Industrial Technology, Churchill, VIC, Australia. 2009. p.1–6. doi:10.1109/ICIT.2009.4939502.
5. Madhira K, Gandhi A, Gujral A. Self balancing robot using complementary filter: Implementation and analysis of complementary filter on SBR. 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India. 2016. p.2950–2954. doi:10.1109/ICEEOT.2016.7755240
6. Bilal M, Razia A, Hussain A, Khan M, Munaim A, Alim M. PWM waveform generation and control using rotary encoder on Spartan-3E starter board. 2023 7th International Multi-Topic ICT Conference (IMTIC), Jamshoro, Pakistan. 2023. p.1–6. doi:10.1109/IMTIC58887.2023.10178475.
7. Thakral S, Joshi AM, Mehta U. PWM waveform generation using rotary encoder on Spartan-3E starter kit. 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India. 2017. p.1–4. doi:10.1109/CIACT.2017.7977372.
8. Zhuang Y, Hu Z, Yao Y. Two-wheeled self-balancing robot dynamic model and controller design. Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, 2014, pp. 1935-1939, doi: 10.1109/WCICA.2014.7053016.
9. Pajaziti A, Gara L. Navigation of self-balancing mobile robot through sensors. IFAC-PapersOnLine. 2019;52(25):429–434. doi:10.1016/j.ifacol.2019.12.576.
10. Hajdu S, Brassai ST, Szekely I. Complementary filter based sensor fusion on FPGA platforms. 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), Brasov, Romania. 2017. p.851–856. doi:10.1109/OPTIM.2017.7975076.
11. Islam T, Islam MS, Mahmud SU, Haider HE. Comparison of complementary and Kalman filter based data fusion for attitude heading reference system. AIP Conf Proc. 2017;1919(1):020002. doi:10.1063/1.5018520.
12. Ali FH, Hussein MM, Ismael SMB. LabVIEW FPGA implementation of a PID controller for DC motor speed control. In: Yasseen S, Ali AA, editors. Proceedings of the 2010 1st International Conference on Energy, Power, and Control (EPC-IQ); 2010 Nov 30–Dec 2; Basrah, Iraq. Basrah: College of Engineering, University of Basrah; 2010. p.139–144.
13. Agarwal C, Gupta A. Modeling, simulation based DC motor speed control by implementing PID controller on FPGA. In: Confluence 2013: The Next Generation Information Technology Summit (4th International Conference). IET; 2013. p.467–471. doi:10.1049/cp.2013.2358.

14. Ramadan EAEHM, El-Bardini M, El-Rabaie NM, Fkirin MA. Embedded system based on a real time fuzzy motor speed controller. Ain Shams Eng J. 2014;5(2):399–409. doi:10.1016/j.asej.2013.10.001.
15. Santoso HP, Saputro JS, Maghfiroh H, Dinata MM. Balancing robot navigation with virtual map and virtual sensor. 2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), Tangerang, Indonesia. 2020. p.218–223. doi:10.1109/ICRAMET51080.2020.9298584.
16. Shafiekhani A, Mahjoob MJ, Akraminia M. Design and implementation of an adaptive critic-based neuro-fuzzy controller on an unmanned bicycle. Mechatronics. 2015;28:115–123. doi:10.1016/j.mechatronics.2015.04.010.

