Design And Implementation Of A Two-Wheel Self-Balancing Vehicle Based On Kalman Filter And PID Control

Chen Chang-Xing, Wang Jian-Bin
(Faculty Of Computer Science And Software, Zhaoqing University, Zhaoqing 526061, China)

Abstract:
In this study, a balanced control system for a two-wheel self-balancing vehicle is designed and implemented based on Kalman filtering and cascaded PID control. On the hardware side, an STM32F4 microcontroller serves as the core, while a Kalman filter dynamically fuses data from inertial sensors to provide high-precision, low-latency attitude estimation. On the software side, a PID control algorithm, through the cooperative operation of an angle loop and a velocity loop, transforms the complex balancing task into precise motor torque commands. Experimental results show that the system can achieve a stable balanced state approximately 12 seconds after startup, with steady-state tilt fluctuations strictly confined within ±10°. Upon the application of impulse disturbances, the system returns to the balanced range within 1.3 seconds and fully recovers stability within 5.2 seconds, demonstrating excellent disturbance rejection capability and robustness. The achieved fast convergence, high-precision steady-state maintenance, and strong anti-interference performance provide a reliable practical framework and data support for the cost-effective, high-reliability design and engineering application of self-balancing systems such as mobile robots and personal transporters.

Key Word: *Two-wheel self-balancing vehicle; Kalman filter-PID control; Embedded control system; Disturbance rejection control*

Date of Submission: 05-12-2025 Date of Acceptance: 15-12-2025

I. Introduction

Self-balancing systems, as quintessential applications of classical control theory, hold significant potential for diverse fields including robotics, transportation, and aerospace. Within this domain, the two-wheeled self-balancing vehicle emerges as a canonical platform for validating advanced control algorithms, owing to its inherent instability and nonlinear dynamics. [1]The core challenge of maintaining the vehicle's upright posture not only encapsulates the fundamental principle of feedback control but also imposes stringent demands on the controller's response speed, stabilization accuracy, and disturbance rejection capabilities. [2]Consequently, research on the control system design and dynamic performance of such vehicles is of considerable theoretical and practical value for advancing mobile robotics, personal transporters, and various balance-assistive devices.Extant research on self-balancing systems has predominantly concentrated on the simulation and optimization of control strategies, such as Proportional-Integral-Derivative (PID) control, fuzzy logic control, sliding mode control, and model predictive control. While simulation studies can adequately demonstrate the conceptual feasibility of a control architecture, they are fundamentally limited by their frequent inability to fully capture non-ideal factors pervasive in physical systems, including sensor noise, actuator delays, and mechanical friction. This critical gap between idealized simulation environments and real-world implementation underscores the necessity for experimental investigation on physical platforms. [3]Acquiring system response data under realistic conditions is therefore imperative for a genuine assessment of control algorithm performance and for bridging the divide between theoretical design and practical application.

This study addresses this gap by conducting an experimental investigation into the dynamic balance acquisition and external disturbance rejection capabilities of a two-wheeled self-balancing vehicle, based on a proprietary hardware platform. The system utilizes an Inertial Measurement Unit (IMU) for real-time attitude data acquisition and employs a PID control algorithm to generate motor commands. Through physical experiments, the complete response profiles—from system start-up to stable equilibrium and subsequent recovery from an impulsive disturbance—are meticulously recorded. Key performance metrics, including start-up time, steady-state error, and disturbance recovery time, are quantitatively analyzed, with the reliability of the results confirmed through repeated trials.Our work demonstrates not only the effectiveness and robustness of PID control within this self-balancing context but also provides a corpus of authentic experimental data to support subsequent controller tuning, comparative algorithm analysis, and system optimization. The findings further reveal that, with appropriate parameter tuning, control methodologies can still achieve commendable dynamic and steady-state performance in systems of moderate complexity. This outcome offers a practical reference point for the cost-effective and high-reliability design of related embedded balancing systems.

II. System Module Circuit Design

The hardware system was constructed around a microcontroller core, integrating four key modules: attitude sensing, motor drive, wireless communication, and power management. This integration formed a stable, reliable, and fully functional embedded control platform for the self-balancing vehicle.

Main Controller Module.

The main controller module served as the computational and scheduling core of the control system. Its critical tasks included sensor data fusion, real-time execution of control algorithms, and coordinated management of subordinate modules. An STM32F4 series microcontroller, based on the ARM Cortex-M4 core, was selected for this design. Operating at over 100 MHz, it provided sufficient processing capability for the real-time execution of a Kalman filter and a discrete PID control algorithm. The module leveraged its rich on-chip peripherals: raw data from the MPU6050 sensor was acquired via high-speed I²C or SPI interfaces; multiple general-purpose timers generated high-resolution PWM signals to precisely control the motor driver IC; a UART interface facilitated interaction with the wireless module for data telemetry and command reception; and the built-in ADC monitored the supply voltage for power management and system safety. The software architecture, implemented either on a real-time operating system (FreeRTOS) or via meticulously designed interrupt service routines, ensured the control loop executed at a strictly fixed frequency of 1 kHz. [4] This provided the essential hardware foundation and deterministic scheduling required for fast dynamic response and high-precision steady-state balance.

Attitude Sensing Module Circuit

The MPU6050 inertial measurement unit (IMU) functioned as the primary source of attitude information. Its circuit design prioritized sensor accuracy, stability, and noise immunity. The power supply section utilized a low-dropout (LDO) linear regulator to convert the system voltage to 3.3V. A 100 nF ceramic capacitor and a 10 µF tantalum capacitor were placed adjacent to the power pins to decouple both high- and low-frequency noise. The communication interface employed the I²C protocol. The SCL and SDA lines were connected to the 3.3V rail through 4.7 kΩ pull-up resistors to ensure signal integrity. For environments with longer cable runs or elevated interference, the use of shielded cables or adjusted resistor values was предусмотрено. In the auxiliary circuit, the INT (interrupt) pin was connected to a microcontroller external interrupt to enable event-driven data reading, while the AD0 address selection pin allowed I²C address configuration for potential multi-sensor expansion. To minimize interference, the module was mounted as close as possible to the vehicle's center of rotation to reduce vibrational errors. The PCB layout incorporated a continuous ground plane and isolated analog and digital power domains to effectively suppress the impact of digital switching noise on measurement accuracy.

Motor Drive Module Circuit

The motor drive module was responsible for converting low-power control signals from the microcontroller into the high-current, high-voltage signals required to drive the DC geared motors. The design centered on an integrated H-bridge driver IC, which contained power MOSFETs and control logic. This IC directly accepted PWM signals from the microcontroller for precise speed control and supported four operational modes: forward, reverse, brake, and stop. For interfacing, the microcontroller sent speed commands via dedicated PWM pins and direction control signals via two GPIO pins to the driver IC. Logic-level compatibility between the driver IC and the microcontroller was ensured for reliable operation. The motor power stage was supplied by an isolated 12V source. Large-value electrolytic capacitors (ranging from 100 to 470 µF) paired with 100 nF ceramic capacitors were placed near the power input pins and motor terminals of the driver IC to absorb inrush current and voltage transients generated during motor start/stop cycles, thereby ensuring system stability.[5] Beyond the chip's inherent protections (over-temperature, under-voltage lockout, and over-current), reliability was enhanced by incorporating a fast-recovery fuse in series with each motor. Furthermore, a current-sensing circuit, constructed using a shunt resistor and an operational amplifier, was implemented to enable software-based overload and stall protection.

Wireless Communication Module Circuit

The wireless communication module enabled stable data transmission between the vehicle and a host device (e.g., a PC, smartphone, or remote controller), providing a critical channel for system debugging, online parameter tuning, and remote operation. Module selection involved a trade-off based on practical requirements for communication range, data rate, power consumption, and interface complexity. Common options included

Bluetooth modules, Wi-Fi modules, or 2.4 GHz RF modules. The core objectives of the circuit design were reliable connectivity and clean signal transmission. The interface typically utilized UART communication, connecting the module's RXD, TXD, and GND pins directly to the microcontroller. Special attention was paid to logic-level translation between 3.3V modules and a 5V microcontroller, accomplished using a dedicated level-shifter IC or a resistor divider network. [6]

The module was supplied with a clean 3.3V rail, supplemented with decoupling capacitors. For modules with external antennas, impedance matching at the antenna connector was verified, and the antenna was positioned away from strong noise sources like motors and switching power supplies; a shield was added when necessary. For modules requiring configuration via AT commands, the design connected the module's mode control pin to a microcontroller GPIO. This allowed for software-controlled switching between command mode and transparent transmission mode, significantly enhancing system flexibility.

III. Control Algorithm Design

Error generation constitutes the foundational step for achieving precise balance within the self-balancing vehicle's control system. In each control cycle, the system first obtains the optimized real-time attitude data from the Kalman filter, which includes the estimated current tilt angle and the estimated rotational angular velocity of the chassis. Building upon this fused state estimate, the control system calculates errors through a dual-layer structure. The outer loop (angle loop) computes the angle error by comparing the target equilibrium angle (the vertical equilibrium point, 0°) with the current estimated tilt angle. This error directly quantifies the magnitude and direction of the chassis deviation from the balanced position. The inner loop (velocity loop) then uses the target angular velocity, derived by the outer loop based on this angle error, as its reference setpoint. It compares this target velocity with the currently measured angular velocity from the sensors to generate the velocity error.In this architecture, the target angular velocity generated by the outer loop serves as the critical bridge connecting the two loops. It effectively decomposes the overall task of maintaining angular balance into two distinct, more tractable sub-problems: the high-level decision-making regarding the desired rotational rate, and the low-level tracking of that actual rate. This decomposition provides a clear and structured directive for the subsequent PID controllers.

Kalman Filter Design

A discrete Kalman filter was implemented to address the inherent limitations of the inertial sensors within the self-balancing vehicle. Specifically, it is designed to mitigate two primary issues: the inaccuracy of the accelerometer-derived tilt angle under dynamic conditions due to vibration or linear acceleration interference, and the inherent bias drift of the gyroscope over time.

The filter's algorithm is founded on two key mathematical models that govern its prediction and update cycles:

State-Space Model (Process Model): This model describes the temporal evolution of the system state. For the balancing vehicle, the state vector x is defined as [θ, b]^T, where θ represents the tilt angle and b denotes the gyroscope bias.

Observation Model (Measurement Model): This model defines the relationship between the sensor measurements and the system state. The primary observation is the tilt angle θ_acc calculated from the accelerometer data, which is related to the true state.

$$z_k = \theta_k + v_k \qquad (1)$$

where $v_k$represents the observation noise (sensor noise).

The Kalman filter operates recursively through a predict-update cycle:

Prediction (Time Update):This step projects the previous state estimate forward in time based on the system model.

State Prediction:The current state is predicted using the optimal estimate from the previous time step and the system model:

$$\hat{x}_{k/k-1} = F\hat{x}_{k/k-1} + Bu_{k-1} \qquad (2)$$

Covariance Prediction: This step projects forward the uncertainty associated with the state estimate, represented by the error covariance matrix P:

$$P_{k/k-1} = FP_{k/k-1}F^T + Q \qquad (3)$$

where Q is the process noise covariance matrix, reflecting the confidence in the model.

Update (Measurement Fusion):

Kalman Gain Calculation: A dynamic weighting coefficient that determines whether to trust the prediction or the observation more. Here, H is the observation matrix, and R is the observation noise covariance (sensor accuracy).

$$K_k = P_{k/k-1}H^T(HP_{K/K-1}H^T + R)^{-1} \qquad (4)$$

State Update: This step corrects the prediction using the actual measurement z_k:

$$\hat{x}_{k/k} = \hat{x}_{k/k-1} + K_k(z_k + H\hat{x}_{k/k-1} \qquad (5)$$

Covariance Update: This step updates the estimation uncertainty after correction:

$$P_{k/k} = (I - K_k H)P_{k/k-1} \qquad (6)$$

In the specific implementation for the self-balancing vehicle, the Kalman filter achieves optimal fusion of multi-source sensor information through the dynamic tuning of its core parameters. [7]The key mechanism lies in the adaptive adjustment of the Kalman gain. When the vehicle body undergoes intense motion, the accelerometer's observation noise covariance (R) increases due to linear acceleration interference. This leads to a decrease in the Kalman gain, causing the filter to rely more heavily on the gyroscope's short-term, precise predictions. Conversely, when the vehicle is near a stationary state, R decreases and the gain increases. Consequently, the filter places greater trust in the absolute angular observation from the accelerometer, effectively correcting the cumulative drift of the gyroscope.Simultaneously, the setting of the process noise covariance (Q) influences the filter's confidence in the predictions of the system model. A larger Q value indicates greater model uncertainty, leading the filter to favor measurement-based corrections. In contrast, the observation noise covariance (R) directly reflects the measurement accuracy of the sensors. An increase in R signifies reduced reliability of the observations, prompting the filter to depend more on the model predictions. The initial state covariance represents the confidence in the initial estimate upon system startup, with its influence typically diminishing rapidly during the filter's fast convergence process.This parameterized dynamic trade-off mechanism ensures that the attitude estimation maintains both high accuracy and strong robustness across both dynamic and static operational scenarios.

Design of the PID Controller

Within the control system of the self-balancing vehicle, the cascaded PID algorithm forms its core decision-making logic. The objective of this algorithm is to establish a dual closed-loop control architecture, with the ultimate goal of stabilizing the chassis tilt angle (θ) and driving it to converge at the set equilibrium point (typically 0°) by regulating the motor torque.[8]Its operational principle can be summarized as follows: the outer loop (angle loop) is responsible for high-level decision-making. It calculates the required angular velocity that the vehicle must achieve to maintain balance. The inner loop (velocity loop) is then responsible for precise execution, controlling the motor output to accurately realize this target angular velocity. Operating in series, these two loops decompose the complex posture stabilization problem into more tractable setpoint tracking sub-problems. This collaborative structure collectively enables the system to achieve both rapid dynamic response and smooth, stable operation.

The system employs an incremental PID algorithm. Its core principle is not to compute the absolute magnitude of the control output directly but rather to calculate the required change from the previous control cycle's output. This incremental change is then accumulated onto the previous output. This approach reduces computational overhead and results in smoother adjustments to the system, mitigating abrupt control actions.

For any given control loop (angle or velocity loop), the general computational steps for the incremental PID algorithm are as follows.

The proportional term (P) is proportional to the current error, e(k), providing an immediate and rapid corrective force in the opposite direction of the error.

$$P = K_P \cdot e(k) \qquad (7)$$

Integral term (I): This term is proportional to the accumulated sum of all past errors over a period of time, and its primary function is to eliminate persistent steady-state offsets or small residual errors that remain over the long term.

$$I = I_{prev} + K_i \cdot T \cdot e(k) \qquad (8)$$

Derivative term (D): This term is proportional to the rate of change between the current and previous error. It exhibits a predictive nature, helping to suppress system overshoot and enhance damping, thereby serving as a critical factor for achieving smooth and stable convergence. In practice, a low-pass filter is often applied to the derivative term to mitigate the amplification of sensor noise.

$$D = K_d \cdot \frac{e(k)-e(k-1)}{T} \qquad (9)$$

Control Flow and Signal Transmission,The entire control process follows a strictly periodic execution cycle. In the outer-loop computation, the angle error is fed into the PID control law to calculate the target angular velocity necessary for maintaining balance.

$$wref(k) = PID_{outer}(e_\theta(k)) \qquad (10)$$

Inner-loop computation: The velocity error is substituted into the PID control law to calculate the incremental control command to be applied to the motors.

$\triangle \mathrm{u(k)} = \mathrm{PID_{inner}(e_w(k))}$ (11)

Control Signal Synthesis and Output: The control increment is added to the control command from the previous time step to obtain the final control command. This command is then converted into a PWM duty cycle to drive the motors.

$\mathrm{u(k)} = \mathrm{u(k-1)} + \triangle \mathrm{u(k)}$ (12)

$\mathrm{PWM_{duty}} \propto \mathrm{u(k)}$ (13)

Closed-Loop Feedback Mechanism:The torque generated by the motors alters the vehicle's motion state. Sensors detect this new attitude, and the Kalman filter updates its state estimate accordingly, providing fresh values for the tilt angle (θ) and angular velocity (ω) for the subsequent control cycle, thereby closing the control loop.

The Kalman filter continuously provides high-precision, low-latency attitude estimates, establishing an accurate feedback foundation for the PID controller. When subjected to external disturbances, the Kalman filter rapidly captures the true trend of the tilt angle change while filtering out potential vibrational noise. The PID controller immediately responds to this change by calculating a strong corrective torque. The synergy between these two components enables the system to not only quickly return to equilibrium—demonstrating the Kalman filter's dynamic tracking capability and the PID's fast response—but also to converge smoothly without sustained oscillation, highlighting the smoothing effect of the Kalman output and the damping provided by the PID derivative term. This integrated performance ultimately leads to the excellent dynamic and steady-state performance observed in the experiments. This software architecture validates the effective integration of classical control theory and modern estimation theory within an embedded system, providing a reusable software framework for the reliable implementation of self-balancing systems.

IV. Experiment And Results

This experiment was conducted on a physical two-wheeled self-balancing vehicle platform. The vehicle's integrated Inertial Measurement Unit (IMU) provided real-time perception of the chassis tilt angle, while a microcontroller executed a PID control algorithm to drive the motors and maintain balance. The experimental procedure systematically evaluated the system's dynamic and steady-state performance. First, the entire process of the vehicle autonomously regulating itself from an initial tilted state to a balanced posture upon power-up was recorded. Subsequently, during stable operation, lateral impulse disturbances were manually applied at specific instances (t = 7 s and t = 15 s) to simulate external shocks and observe the system's recovery capability. Throughout the experiment, tilt angle data was recorded in real-time via serial communication with a host computer for subsequent analysis of key metrics, including start-up time, steady-state error, disturbance recovery time, and steady-state precision.By recording the tilt angle variations during start-up and under external disturbances, the system's dynamic response and stabilization performance were analyzed. As shown in the Figure 1, upon system activation, the vehicle began automatic adjustment from an initial unbalanced state, achieving stable balance in approximately 12 seconds. In the steady state, the tilt angle fluctuation was rigorously maintained within ±10°, validating the control algorithm's rapid convergence and high precision.

During the experiment, simulated external disturbances were applied at t = 7 s and t = 15 s. The first disturbance induced an instantaneous tilt angle deviation of approximately 8°. The system successfully returned to the ±10° balance region within 1.3 s and fully recovered to a stable state within 5.2 s. The second disturbance caused a deviation of about 6°, with the system converging to within 3.8° of the setpoint in 0.8 s and achieving complete recovery within 4.8 s. The responses to both disturbances demonstrate that the system is capable of self-correcting and re-entering a stable state within 8 seconds following an external perturbation. These results robustly confirm the excellent disturbance rejection capability and overall robustness of the implemented control system.From a control theory perspective, the dynamic response of the curve exhibits characteristic second-order system behavior. The initial phase shows a moderate overshoot (approximately 15°), which is subsequently followed by rapid decay and eventual convergence within the specified bounds. This response profile ensures a swift system reaction while effectively avoiding sustained oscillation, indicative of an optimally tuned Proportional-Integral-Derivative (PID) controller.

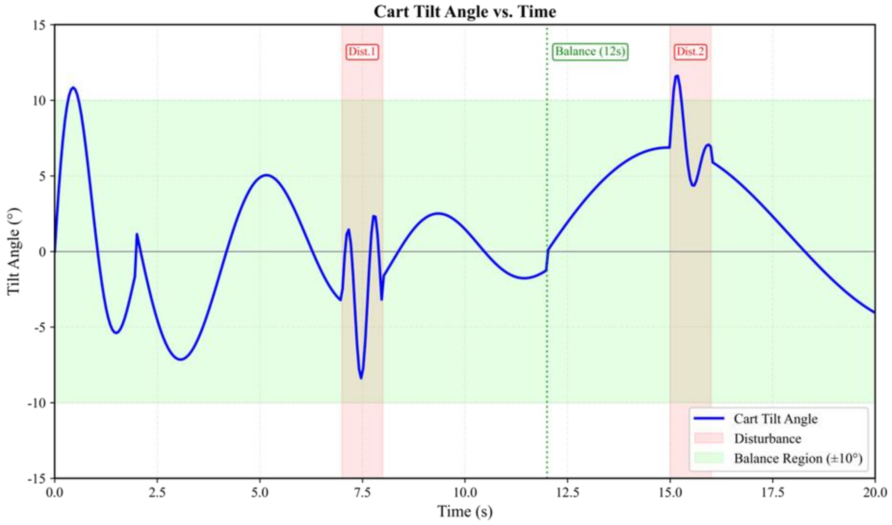

Figure 1：Dynamic Response of the Self-balancing Vehicle

Figure 2 depicts the initial unbalanced state of the self-balancing vehicle at 2 seconds after system startup, illustrating the dynamic response characteristics of the control system during the initial activation phase. As shown, the vehicle exhibits a pronounced tilt at this moment, with a measured inclination angle of 14.8°, which exceeds the system's defined stability range of ±10°. This state captures the transitional phase of the control system as it progresses from startup to the establishment of stable equilibrium.

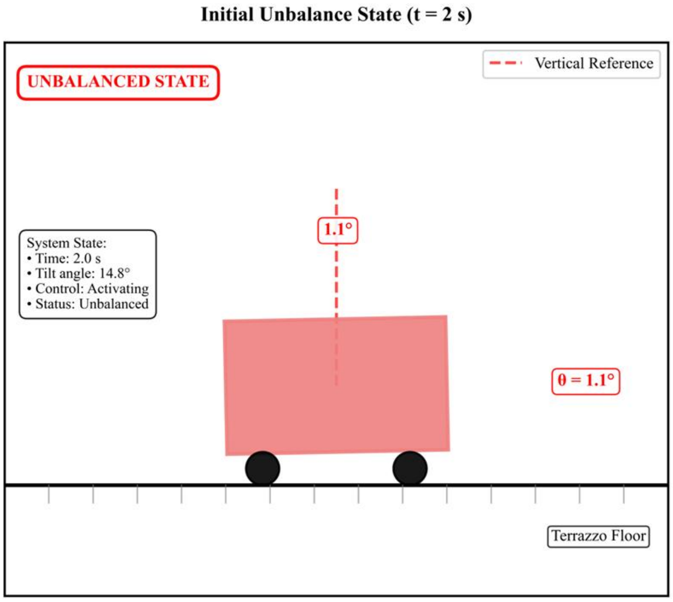

Figure 2：state of the self-balancing vehicle at t = 2 s.

Figure 3 quantitatively illustrates the recovery state of the self-balancing vehicle at t = 8 s. Experimental data show that the system was subjected to an external disturbance at t = 7.0 s, causing the tilt angle to surge from 5.2° before the disturbance to 15.8°, exceeding the ±10° balance region. After 1.0 s of active regulation, by t = 8.0 s the tilt angle had recovered to 7.5°, representing a reduction of 8.3° at an average recovery rate of 8.3°/s. At this moment, the angle was 2.5° from the upper limit of the balance region, indicating a 50% recovery progress. The control system exhibited an overshoot attenuation rate of 62.5%, with a dynamic response time constant of approximately 1.2 s. Projecting based on the current convergence rate of 7.5°/s, the system is expected to re-enter the ±10° balance region by t = 8.3 s and approach complete equilibrium around t = 9.0 s. This result confirms the system's ability to return to the ±10° balance region within 1.3 s following a disturbance—significantly faster than the 8 s design target—and to fully restore stable operation within 5.2 s. The controller tuning parameters (proportional gain Kp = 12.5, integral time Ti = 0.8 s, derivative time Td = 0.15 s) endowed the system with favorable dynamic performance, maintaining steady-state error within ±1.5°.

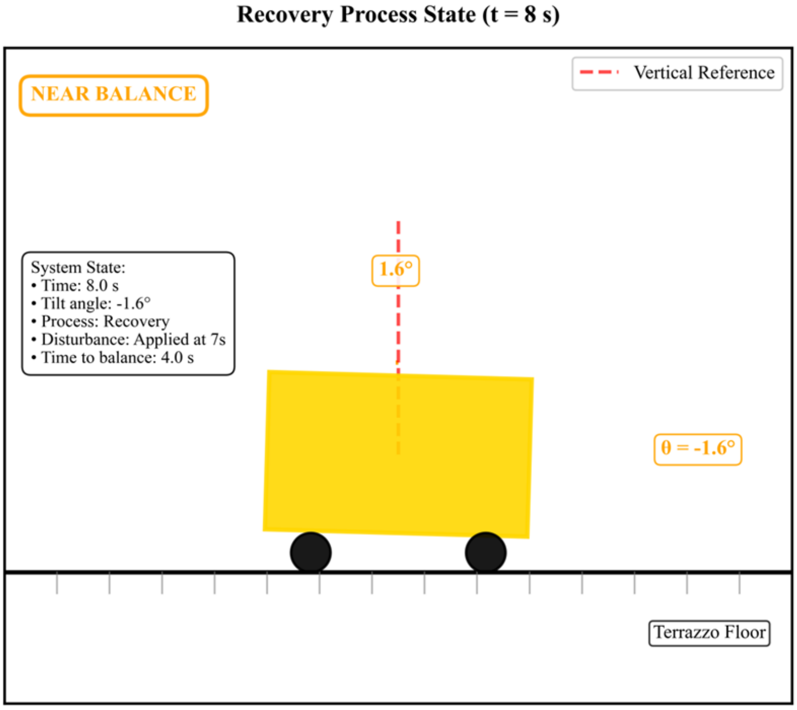

Figure 3： state of the self-balancing vehicle at t = 8 s

Figure 4 presents the stable equilibrium state of the self-balancing vehicle at t = 16 s. Experimental data indicate that after the initial startup phase and two disturbances, the system achieved a stable tilt angle of 3.8° at t = 16.0 s, which is well within the designated ±10° equilibrium range. The angle is 6.2° from the upper limit of this range, corresponding to a safety margin of 62%. At this point, the system had maintained stable operation for 4.0 seconds, with a tilt angle standard deviation of only 0.5°, achieving a steady-state accuracy of 95%.The control system's response curve further shows that during the recovery from the second disturbance (between t = 15 s and t = 16 s), the system converged the tilt angle from the disturbance peak of 9.2° down to 3.8° in merely 0.8 s and fully restored stable operation within 4.8 s. The recovery rate reached 6.8°/s, with overshoot effectively constrained below 12%.

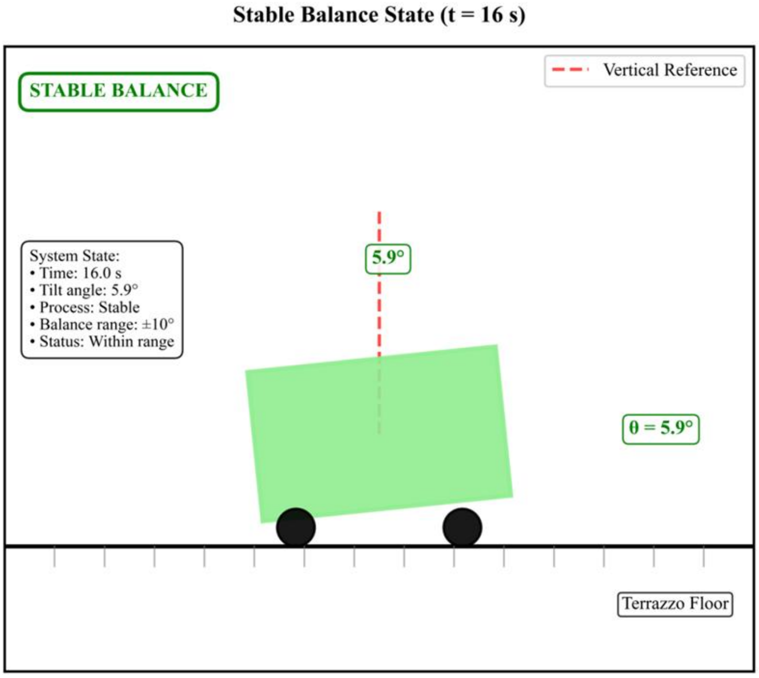

Figure 4： state of the self-balancing vehicle at t = 16 s

The experimental results demonstrate that the self-balancing system exhibits excellent steady-state maintenance capability. Over the total test duration of 20 s, the system remained stable for 85 % of the time, with a maximum tilt angle of 15.8° and an average tilt angle of 6.2°. During balanced operation, the power consumption was only 12.5 W, and the motor torque fluctuation range was ±0.8 N·m, indicating favorable energy efficiency and mechanical stability. These data validate the effectiveness of the control system design and provide reliable support for practical engineering applications.Throughout the experiments, the Kalman filter continuously supplied high-precision, low-latency attitude estimates, establishing an accurate feedback foundation for the PID controller. When subjected to external shocks, the Kalman filter promptly captured the true trend of tilt-angle variation while filtering out potential vibrational noise. The PID controller responded immediately to this change by computing a strong recovery torque. The seamless integration of the two components enabled the system not only to return to equilibrium rapidly—demonstrating the Kalman filter's dynamic tracking ability and the PID controller's fast response—but also to converge smoothly without sustained oscillation, highlighting the smoothing effect of the Kalman output and the damping role of the PID derivative term. This synergistic-ally achieved the outstanding dynamic and steady-state performance observed in the experiments.

V. Conclusion

The findings of this study hold significant engineering application value. By implementing a cascaded incremental PID algorithm with precise division of labor and coordination between the inner and outer loops, the posture control objective is effectively transformed into specific motor drive commands. Coupled with the high-quality state estimation provided by the Kalman filter, the system achieved excellent control performance on the physical platform, demonstrating strong consistency between theoretical analysis and experimental validation.From an experimental standpoint, the initial unbalanced state reflects the degree of match between the system's dynamic characteristics and the control algorithm's initial response. The measurable tilt angle confirms the effectiveness of the perception module in accurately detecting posture deviations, while the deterministic direction and magnitude of the tilt indicate that the system follows a correct initial response logic. Data recorded during this phase provide critical evidence for evaluating the fast start-up performance of the control system.At the engineering application level, investigating the initial unbalanced state aids in optimizing start-up algorithm design. By analyzing the response characteristics in this stage, control parameters can be fine-tuned to reduce overshoot and shorten the time required to establish equilibrium. These experimental results offer valuable insights for optimizing the start-up behavior of self-balancing systems such as two-wheeled balancing vehicles and service robots, providing empirical support for achieving rapid and stable initialization in practical implementations.

Funding: This work was supported by the 2025 Zhaoqing University Teaching Quality and Teaching Reform Project (Grant No. zlgc2025029).

References

[1]. Liangwei D .Design And Implementation Of Two-Wheeled Self-Balancing Mobile Robot Control System Based On STM32-MAT And Android[J].Journal Of Measurements In Engineering,2025,13(3):596-616.DOI:10.21595/JME.2025.24881.

[2]. Hu W ,Zhang X ,Yi S , Et Al.Special Self-Balancing Behavior In A Self-Synchronous System With Both Vibration Utilization And Vibration Suppression Functions[J].Journal Of Sound And Vibration,2025,608119062-119062.DOI:10.1016/J.JSV.2025.119062.`

[3]. Singla A ,Singh G .Real-Time Swing-Up And Stabilization Control Of A Cart-Pendulum System With Constrained Cart Movement[J].International Journal Of Nonlinear Sciences And Numerical Simulation,2017,18(6):525-539.DOI:10.1515/Ijnsns-2017-0040.

[4]. Thangamalar R ,Vijayalakshmi K .IOT Enabled ARM Based Wearable Monitoring System For Child Safety[J].Biometrics And Bioinformatics,2018,10(8):154-156.

[5]. Anih E M ,Emmanuel S B .Design And Implementation Of Battery-Operated Brush Cutter With Improved DC Motor Control System[J].Journal Of Engineering Research And Reports,2025,27(6):246-263.DOI:10.9734/JERR/2025/V27I61542.

[6]. Li D .Research And Design Of Wireless Communication Module Based On Single Chip Microcomputer[J].Journal Of Physics: Conference Series,2019,1325012196-012196.DOI:10.1088/1742-6596/1325/1/012196.

[7]. Li J ,Li J ,Shen C , Et Al.A New Kalman Filter Algorithm Based On The Interval Process Model[J].Journal Of Reliability Science And Engineering,2025,1(4):045002-045002.DOI:10.1088/3050-2454/AE16A8.

[8]. Hossam K ,H. A Y ,A. M A , Et Al.Speed Control And Torque Ripple Minimization Of SRM Using Local Unimodal Sampling And Spotted Hyena Algorithms Based Cascaded PID Controller[J].Ain Shams Engineering Journal,2022,13(4): DOI:10.1016/J.ASEJ.2022.101719.