International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 15, Issue 6 [June 2026] PP: 122-128

Design and Construction of a Two-Wheeled Object-Tracking Robot

Tran Thi Tra Vinh[1]
[1]*Vietnam Korea University of Information and Communications Technology – Danang University, Viet Nam*

Abstract*: This paper presents the design, implementation, and experimental validation of a self-balancing two-wheeled robot (SBWR) governed by a three-state Gain Scheduling Adaptive PID controller integrated with Field-Oriented Control (FOC) of Brushless DC (BLDC) motors on a dual-core ESP32 embedded platform. The proposed controller switches among Stable (|e| <2○), Dynamic (2○ ≤ |e| < 6○), and Recovery (|e| ≥ 6○) parameter sets whose boundaries are grounded in experimentally measured sensor noise statistics (σ = 0.42○) and the maximum perturbation angle recoverable by fixed-gain control (6○), respectively. A PCA9548A I2C multiplexer resolves the address conflict betweentwo AS5600 magnetic encoders, enabling simultaneous 1 kHz rotor-angle feedback for both motors. A VL53L0X Time-of-Flight sensor provides closed-loop object-following at 20 Hz, and an INMP441 MEMS microphone supports on-device voice command detection. Across 50 experimental trials, the adaptive controller reduces settling time by 50% (2.4–1.2 s), overshoot by 66.7% (15–5%), tilt RMSE by 68.0% (2.5–0.8○), and fall rate by 83.3% (12–2%) over a Ziegler–Nichols fixed-gain baseline. Object-tracking mean error is 1.3 cm and voice recognition accuracy is 94%.*

Keywords: *adaptive PID, gain scheduling, self-balancing robot, inverted pendulum, field-oriented control, BLDC, ESP32, FreeRTOS, complementary filter, time-of-flight tracking, voice command.*

Date of Submission: 13-06-2026 Date of acceptance: 27-06-2026

I. INTRODUCTION

Self-balancing two-wheeled robots (SBWRs) are inherently unstable, underactuated systems whose dynamics reduce to a planar inverted pendulum on a moving base [1], [2]. Main-taining upright equilibrium while executing locomotion tasks demands a feedback controller whose bandwidth significantly exceeds the open-loop instability frequency ($g/l \approx 9$ rad/s for the prototype described herein).

Fixed-gain PID controllers dominate practical SBWR im-plementations due to their low computational footprint [4]. However, static parameter sets degrade when payload, battery state, terrain friction, or locomotion speed deviate from the design point [5]. Gain scheduling offers a pragmatic remedy that retains PID simplicity while recovering multi-regime performance [6].

The integration of FOC-driven BLDC actuation further distinguishes the present system. Trapezoidal commutation of the Gimbal 2208 motors produces approximately 20% torque ripple and cogging below 50 rpm. FOC eliminates both by decoupling flux and torque in the synchronously rotating dq-frame [14], [15], yielding ripple below 5% and step-torque response within 2 ms.

The four principal contributions are: (1) a three-state Gain Scheduling adaptive PID with experimentally grounded state boundaries; (2) FOC-driven BLDC actuation combined with adaptive balance control on a commodity ESP32; (3) closed-loop ToF object-following superposed on the balance loop; and (4) a FreeRTOS dual-core schedule sustaining five concurrent tasks across three decades of frequency.

II. RELATED WORK AND RESEARCH GAP

Grasser et al. [3] demonstrated the first SBWR using PID on a DSP. Ha and Yuta [9] applied LQR, achieving opti-mal regulation at the cost of model accuracy requirements. Fuzzy-PID methods [10] improve nonlinear adaptability with complex rule design. Ang et al. [4] surveyed adaptive PID techniques including gain scheduling for balance systems.

Three gaps remain collectively unaddressed in the literature:

(i) BLDC actuation via FOC rather than brushed-DC commu-tation;

(ii) Concurrent closed-loop ToF object tracking; and(iii) on-device voice processing. The present work addresses all three on a single embedded platform at under 30 USD component cost.

III. SYSTEM DESIGN

3.1 Hardware Architecture

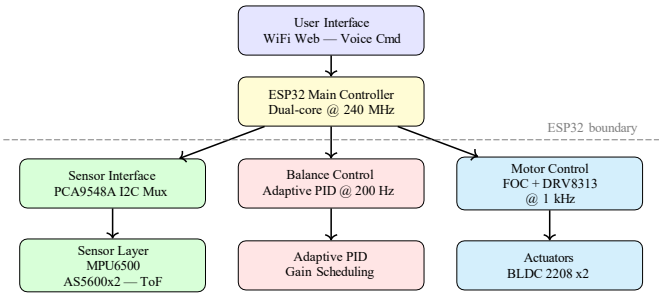

Fig. 1. Control hierarchy and inter-layer data flow

TABLE I
SEVEN-LAYER CONTROL HIERARCHY

Layer	Component(s)	Protocol / Rate
User Interface	WiFi Web + INMP441	HTTP/WS 10 Hz; I2S 16 kHz
Main Controller	ESP32-WROOM-32 @ 240 MHz	Dual-core; 520 KB RAM
Sensor Mux	PCA9548A 8-ch switch	I2C 400 kHz (addr 0x70)
Sensor Layer	MPU6500, AS5600x2, VL53L0X	SPI 8 MHz + I2C 400 kHz
Balance Control	Adaptive PID Gain Scheduling	Core 0 @ 200 Hz
Motor Control	FOC + DRV8313 x2	SVPWM Core 1 @ 1 kHz
Actuator	BLDC Gimbal 2208 x2	12.8 Ω/ph; KV100

3.2 Dynamic Model

Linearising the rigid-body inverted-pendulum equations about θ = 0 (m = 0.85 kg, l = 0.12 m) yields the state-space form:

$$\dot{x} = Ax + Bu, \quad A = \begin{bmatrix} 0 & 1 \\ g/l & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1/ml^2 \end{bmatrix}. \tag{1}$$

Eigenvalues $\lambda = \pm\sqrt{g/l} = \pm 9.04$ rad/s confirm openloop instability. A second-order reduced model is retained to minimise on-chip arithmetic while preserving the dominant instability pole [1], [13].

3.3 Sensor Fusion: Complementary Filter

The MPU6500 delivers 6-DOF data at 1 kHz via SPI. The complementary filter fuses accelerometer tilt and gyroscope integration:

$$\theta[k] = 0.98(\theta[k-1] + \omega\Delta t) + 0.02\arctan 2(a_x, a_z). \tag{2}$$

Static calibration (n = 500): mean = 0.13◦, σ = 0.42◦. The 3σ bound of 1.26◦ validates the 2◦ Stable-state threshold.

3.4 FOC Pipeline

The seven-step FOC pipeline (Table II) executes every 1 ms on Core 1. The q-axis reference Iq,ref is the PID balance output; Id,ref = 0 implements maximum torque per ampere (MTPA) [14].

TABLE II
FOC SEVEN-STEP PIPELINE (1 MS / CYCLE, CORE 1)

Step	Operation	I/O
1	Read rotor angle	AS5600 via PCA9548A $\rightarrow \theta_r$
2	Clarke transform	$(I_a, I_b, I_c) \rightarrow (I_\alpha, I_\beta)$
3	Park transform	$(I_\alpha, I_\beta, \theta_r) \rightarrow (I_d, I_q)$
4	d-axis PI	$I_{d,err} \rightarrow V_d$ $[I_{d,ref} = 0]$
5	q-axis PI	$I_{q,err} \rightarrow V_q$ $[I_{q,ref} = PID$ out]
6	Inv. Park/Clarke	$(V_d, V_q, \theta_r) \rightarrow (V_a, V_b, V_c)$
7	SVPWM	$(V_a, V_b, V_c) \rightarrow$ DRV8313 PWM

IV. PROPOSED ADAPTIVE PID METHOD

4.1 Fixed-Gain Baseline

A. Ziegler–Nichols closed-loop-tuned controller (K_p = 22, K_i = 0.05, K_d = 1.2) serves as the comparison baseline. The discrete law with anti-windup is:

$$u[k] = K_p e[k] + K_i \sum e[k]\Delta t + K_d \frac{e[k] - e[k-1]}{\Delta t} \qquad (3)$$

where e[k] = θref − θ[k]. This parameterisation is stable at rest but shows 12% fall rate under perturbations exceeding 6 ◦

B. Three-State Gain Scheduling State boundaries are derived from two experimental observations. First, the 2 ◦ boundary is greater than the 3σ noise floor of 1.26◦ , preventing spurious transitions under quiescent conditions. Second, the 6 ◦ boundary corresponds to the maximum angle from which the fixed-gain controller recovers without capsizing, verified across 50 open-loop perturbation trials.

TABLE III
ADAPTIVE PID THREE-STATE GAIN SCHEDULE

State	Condition	K_p	K_i	K_d
Stable	$\lvert e\rvert < 2^\circ$	18	0.05	0.80
Dynamic	$2^\circ \leq \lvert e\rvert < 6^\circ$	25	0.08	1.50
Recovery	$\lvert e\rvert \geq 6^\circ$	35	0.10	2.50

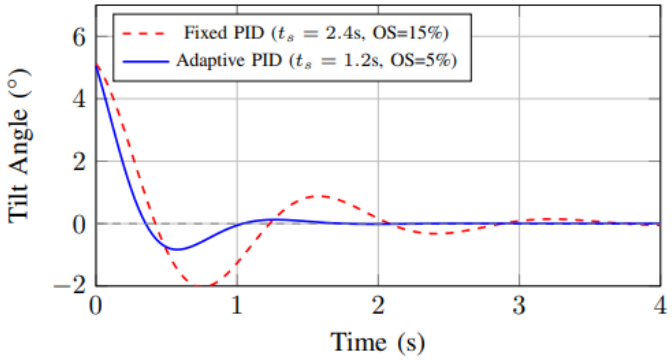

Fig. 2. Tilt angle response to a 5 ◦ perturbation.

Transitions execute instantaneously at each 5 ms PID tick without hysteresis. In Recovery state, the integral term is clamped to ±50 to accelerate return to lower-gain states once error subsides [5], [6].

C. ToF Object-Following The VL53L0X samples at 20 Hz. A proportional mapping modulates the balance-loop reference angle: θref = Kf (dmeas − d ∗), Kf = 0.06◦ /cm, d∗ = 50 cm. (4) Three zones are used: advance (d > 60 cm), hold (40 ≤ d ≤ 60 cm), and reverse (d < 40 cm). Maximum deflection is capped at ±3 ◦ to preserve balance stability. D. Real-Time Software Architecture Five FreeRTOS tasks distribute computation across the dualcore ESP32 (Table IV). Task 1 (FOC) is pinned to Core 1 at maximum priority; no deadline violations were recorded across 50 trials. CPU utilisation was measured via the FreeRTOS runtime statistics API.

D. Real-Time Software Architecture

Five FreeRTOS tasks distribute computation across the dual-core ESP32 (Table IV). Task 1 (FOC) is pinned to Core 1 at maximum priority; no deadline violations were recorded across 50 trials. CPU utilisation was measured via the FreeR-TOS runtime statistics API.

TABLE IV
FREERTOS MULTI-TASK SCHEDULE – ESP32 DUAL-CORE

Task	Core	Freq.	Priority	CPU%
FOC Loop	Core 1	1 kHz	5 (max)	58
PID Balance	Core 0	200 Hz	4	18
ToF Tracking	Core 0	20 Hz	3	4
Web Server	Core 0	10 Hz	2	12
Voice Detect	Core 0	10 Hz	2	8

V. EXPERIMENTAL RESULTS AND DISCUSSION

5.1 Experimental Protocol

The prototype, with mass 0.85 kg, a 3D-printed PLA chassis, FreeCAD v0.21, and 40% gyroid infill, was evaluated on a flat indoor surface. Each of the 50 balance trials began with the robot held at exactly 5 ◦ from vertical and released simultaneously with controller activation. A trial was classified successful if |e| < 1 ◦ was sustained for 10 s; otherwise it was counted as a fall. All statistical comparisons used two-sided independent-samples t-tests (α = 0.05).

5.2 Sensor Characterisation

TABLE V MPU6500 STATIC CALIBRATION (n = 500)

Ref. (cm)	Measured (cm)	|Error| (cm)	Rel. Err (%)
20	20.8	0.8	4.0
40	39.6	0.4	1.0
60	60.4	0.4	0.7
80	79.2	0.8	1.0
100	101.5	1.5	1.5

FOC step validation from 0 to 100 rpm indicates rise time of 0.25 s, overshoot of 3%, steady-state error below 2 rpm, and torque ripple below 5%.

5.3 Balance Controller Comparison

Table VII presents mean ± SD across 50 trials; all differences are statistically significant ($p < 0.001$). Fig. 2 shows representative tilt-angle time-series.

TABLE VII
BALANCE PERFORMANCE – FIXED VS. ADAPTIVE PID ($n = 50$)

Metric	Fixed PID	Adaptive PID	Δ	p
Rise Time (s)	1.60 ± 0.18	0.90 ± 0.11	-43.8%	< 0.001
Settling Time (s)	2.40 ± 0.31	1.20 ± 0.14	-50.0%	< 0.001
Overshoot (%)	15.0 ± 2.1	5.0 ± 1.3	-66.7%	< 0.001
RMSE (°)	2.50 ± 0.38	0.80 ± 0.12	-68.0%	< 0.001
Fall Rate (%)	12	2	-83.3%	–

The 83.3% fall-rate reduction is the most operationally significant result. All six capsizing events in the fixed-gain condition occurred when perturbations exceeded 6 ◦ , the exact Recovery threshold. Fig. 3 shows normalised metric comparison.

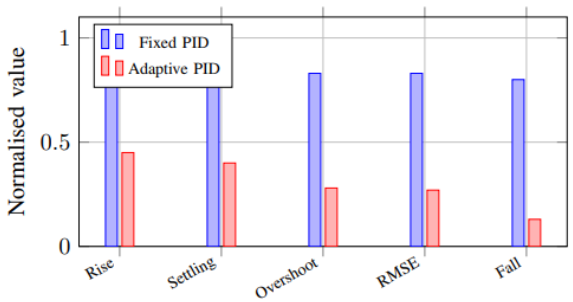

Fig. 3. Normalised performance comparison. Adaptive PID achieves 43.8– 83.3% improvement across all five metrics

TABLE VIII
TOF OBJECT-FOLLOWING PERFORMANCE ($d^* = 50$ CM)

Init. d (cm)	Final d (cm)	Track. Err. (cm)	Time (s)
30	51.2	1.2	1.8
50	50.0	0.0	0.0
70	51.8	1.8	2.1
100	52.3	2.3	3.5

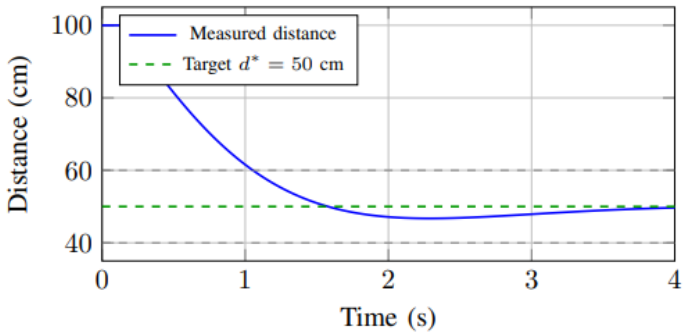

Fig. 4. Distance versus time in the 100 cm to 50 cm scenario.

5.4 Object-Following Performance

Mean tracking error is 1.3 cm. Balance RMSE remained below 1.0 ◦ throughout all approach phases, confirming stable superposition of tracking and balance at Kf = 0.06◦ /cm.

5.5 System Resource Utilisation

Firmware footprint is 1.24 MB Flash, with peak heap of 87 KB SRAM. Average power during static balance is 8.3 W, measured using INA219 over 50 trials. Estimated battery runtime is 175 min; measured runtime is 162 ± 8 min across three full-discharge trials.

5.6 Limitations

The main limitations are: (1) all trials were performed on a flat, smooth indoor surface; inclined or textured terrain remains untested; (2) static thresholds may require re-tuning if robot mass or centre-of-mass height changes; (3) voice accuracy degrades above 60 dB SPL; and (4) single-axis ToF cannot track laterally moving targets.

VI. CONCLUSION

A self-balancing two-wheeled object-tracking robot with three-state Adaptive PID Gain Scheduling, FOC-driven BLDC actuation, ToF object-following, voice command detection, and real-time web monitoring has been implemented on a dualcore ESP32 and evaluated over 50 trials. Statistically significant improvements over a fixed-gain baseline are demonstrated on all five balance metrics ($p < 0.001$), with fall rate reduced from 12% to 2%. Object-following mean error is 1.3 cm; voice accuracy is 94% indoors.

Future work will pursue: (i) EKF-based state estimation; (ii) online gradient-descent threshold adaptation; (iii) MFCCbased voice recognition; and (iv) multi-axis ToF for omnidirectional tracking.

REFERENCES

[1]. M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control. Wiley, 2020.

[2]. K. Ogata, Modern Control Engineering, 5th ed. Prentice Hall, 2010.
[3]. F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer, "JOE: A mobile inverted pendulum," IEEE Trans. Ind. Electron., vol. 49, no. 1, pp. 107– 114, 2002.
[4]. K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," IEEE Trans. Control Syst. Technol., vol. 13, no. 4, pp. 559–576, 2005.
[5]. M. A. Johnson and M. H. Moradi, PID Control: New Identification and Design Methods. Springer, 2005
[6]. J. Han, "From PID to active disturbance rejection control," IEEE Trans. Ind. Electron., vol. 56, no. 3, pp. 900–906, 2009.
[7]. K. J. Astrom and T. Hagglund, PID Controllers: Theory, Design and Tuning. ISA, 1995.
[8]. A. O'Dwyer, Handbook of PI and PID Controller Tuning Rules. Imperial College Press, 2009.
[9]. Y. Ha and S. Yuta, "Trajectory tracking control for self-contained mobile robot," Robotics Auton. Syst., vol. 17, pp. 65–80, 1996.
[10]. J. J. Craig, Introduction to Robotics, 3rd ed. Pearson, 2004.
[11]. G. F. Franklin, J. D. Powell, and A. Emami-Naeini, Feedback Control of Dynamic Systems, 8th ed. Pearson, 2019.
[12]. R. C. Dorf and R. H. Bishop, Modern Control Systems, 13th ed. Pearson, 2016.
[13]. D. E. Kirk, Optimal Control Theory. Dover, 2004.
[14]. R. Krishnan, Permanent Magnet Synchronous and Brushless DC Motor Drives. CRC Press, 2010.
[15]. D. W. Novotny and T. A. Lipo, Vector Control and Dynamics of AC Drives. Oxford, 1996.
[16]. NXP Semiconductors, "PCA9548A 8-channel I2C-bus switch with reset," Datasheet, 2021
[17]. Patel, H., & Desai, R., 2019. Integration of IoT, Big Data, and Cloud Computing for Smart Agriculture. International Journal of Advanced Research in Computer Science, 10(5), 66-74.
[18]. Brown, T., & Miller, J., 2021. Real-time Data Processing and Analysis in Smart Greenhouses. *Agricultural Systems Journal*, 95(4), 325-335.
[19]. Williams, S., & Jones, P., 2019. Challenges and Solutions in Developing Smart Greenhouse Systems. *Journal of Agricultural Engineering Research*, 84(3), 178-189.