# Nonlinear Boost Compensation for PD-Controlled Self-Balancing Robot Stabilization

Steven Tawfik, Mohamed Alkalla, Ayman Nada
Department of Mechatronics & Robotics Engineering
School of Innovative Design Engineering
Egypt-Japan University of Science and Technology (E-JUST)
Alexandria 21934, Egypt
steven.tawfik@ejust.edu.eg, mohamed.gouda@ejust.edu.eg, ayman.nada@ejust.edu.eg

***Abstract*—The paper introduces a PD controller that behaves linearly near equilibrium but adds a bounded nonlinear boost when the tilt error exceeds a threshold, providing additional control authority for large tilt errors that fall outside the local linear operating regime of the PD controller. A reduced-order model for a self-balanced robot is derived and utilized for controller design within MATLAB/Simulink. The proposed hybrid-PD controller with bounded nonlinear augmentation enlarges the practical region of attraction under actuator nonlinearities while preserving local linear stability. A complementary filter fuses gyroscope and accelerometer data for real-time state estimation. Experimental results on an Arduino Nano 33 IoT platform confirm a stable balance with an RMS error of $3.60^\circ$, a settling time of $0.44$ s after a disturbance of $20.47^\circ$, and $96\%$ time within the stability threshold of $\pm 5^\circ$.**

***Index Terms*—Self-Balancing Robot, Nonlinear Boost Compensation, Sensor Fusion, PD-Embedded Control.**

## I. Introduction

Self-balancing robots are canonical testbeds for unstable nonlinear systems, real-time control, and embedded feedback implementation [1]–[5]. Their inherent instability amplifies the sensitivity to sensor noise, actuator nonlinearities, and modeling uncertainty, challenges that are particularly severe on low-cost hardware.

Common approaches for self-balancing control include PID control [6], linear quadratic regulators (LQR) [7], and model predictive control (MPC). Although effective in simulation, real-time deployment in resource-constrained microcontrollers is complicated by dead zones of actuators, friction [12], voltage saturation, and sensor drift

PD controllers are often preferred for mechatronic systems due to their low computational cost and straightforward tuning [5].

In such implementations, actuator limitations rather than controller structure become the primary performance bottleneck. A key challenge is that the effective operating regime of a linear PD controller is limited to a neighborhood of the upright equilibrium. For large tilt errors, the linear approximation breaks down and the PD output alone may be insufficient to produce adequate corrective torque. Previous work on bounded nonlinear control and saturated stabilization [10] provides theoretical motivation for augmenting a linear controller with a bounded nonlinear term that extends the practical region of attraction beyond what a linear law alone achieves.

To address this, the present paper introduces a nonlinear boost compensation mechanism that augments the PD control effort whenever the tilt error exceeds a threshold, while preserving linear behavior near equilibrium. **The main theoretical contribution is the formulation of a hybrid PD control law with bounded nonlinear augmentation that enlarges the practical region of attraction under actuator nonlinearities while preserving local linear stability.**

Specific contributions are: (i) a reduced-order linearized plant model with experimentally identified parameters, validated against open-loop instability tests; (ii) a hybrid PD+boost architecture with derivative filtering, dead-zone compensation, and output saturation for embedded real-time use; (iii) a practical boundedness argument under actuator saturation; and (iv) experimental validation on an Arduino Nano 33 IoT with quantitative comparison against published results.

## II. Mathematical Modeling

The self-balancing robot is modeled as an inverted pendulum mounted on a moving base, which captures the dominant rotational dynamics relevant to upright stabilization [3]. Using the Lagrangian formulation, the rotational motion about the wheel axle can be expressed as

$$J\ddot{\theta}(t) = MgL_{com}\sin(\theta(t)) + \tau(t) \tag{1}$$

where $J$ [kg·m$^2$] denotes the equivalent moment of inertia about the wheel axle, $M$ [kg] is the total mass, $L_{com}$ [m] is the distance from the wheel axle to the center of mass, $g = 9.81$ m/s$^2$ is the gravitational acceleration, $\theta(t)$ [rad] is the body tilt angle measured from the upright position, and $\tau(t)$ [N·m] is the control torque.

For control design near the upright operating point, the small-angle approximation $\sin(\theta) \approx \theta$ for $|\theta| \ll 1$ yields the linearized model:

$$J\ddot{\theta}(t) = MgL_{com}\theta(t) + \tau(t) \tag{2}$$

which represents a second-order unstable system suitable for PD controller synthesis.

To obtain a compact model for embedded implementation, the wheel translational dynamics and motor electrical dynamics are neglected [5]. The actuator torque is approximated as proportional to the motor command,

$$\tau(t) = K_{motor} V(t) \tag{3}$$

where $K_{motor}$ [N·m/PWM] is a lumped gain identified experimentally from the robot's input–output response, and $V(t)$ is the command in PWM counts. Substituting into (2) and applying the Laplace transform yields the reduced-order transfer function:

$$\frac{\Theta(s)}{V(s)} = \frac{K_{motor}}{s^2 - \alpha}, \quad \alpha = \frac{MgL_{com}}{J} \tag{4}$$

Using the measured platform parameters ($M = 1.087$ kg, $L_{com} = 0.08$ m, $J = 0.006$ kg·m$^2$):

$$\alpha = \frac{(1.087)(9.81)(0.08)}{0.006} \approx 142.2 \text{ s}^{-2}, \quad K_{motor} = 25 \tag{5}$$

## III. Control System Design

Fig. 1 shows the overall control-loop structure. The tilt error $e[k] = \theta_{ref} - \theta[k]$ drives a PD controller and a parallel nonlinear boost law; their outputs are summed and passed through dead-zone compensation and saturation before reaching the plant. The tilt angle $\theta[k]$ is estimated by a complementary filter fusing IMU gyroscope and accelerometer data.

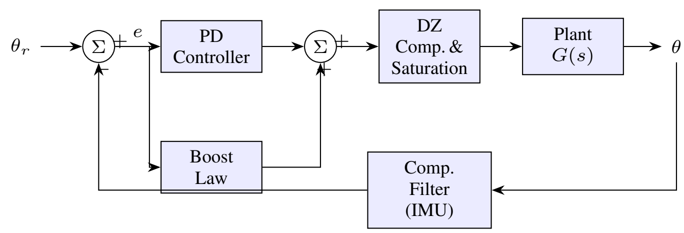


Fig. 1. Control-system block diagram.

### A. Baseline PD Controller

The PD control law operates on the tilt error $e[k] = \theta_{ref} - \theta[k]$:

$$u_{PD}[k] = K_p\, e[k] + K_d\, \dot{e}_f[k] \tag{6}$$

where $\dot{e}_f[k]$ is the low-pass-filtered discrete-time derivative:

$$\dot{e}_f[k] = \alpha_d\, \frac{e[k] - e[k-1]}{\Delta t} + (1 - \alpha_d)\, \dot{e}_f[k-1] \tag{7}$$

Here $k$ is the discrete time index, $\Delta t = 1/f_s \approx 0.026$ s is the sampling period, and $\alpha_d \in (0, 1)$ is the filter coefficient. This first-order IIR structure prevents high-frequency sensor noise from being amplified by the derivative action.

### B. Nonlinear Boost Compensation

The PD controller is effective near the upright equilibrium but provides limited control authority for large tilt errors outside the local linear operating regime. Beyond this regime, additional corrective action is needed both to compensate for nonlinearities not captured in the reduced-order model and to ensure the composite command produces adequate motor torque. The boost term is therefore more accurately interpreted as a *large-error authority extension*: it augments the PD output based on error magnitude whenever the robot moves appreciably away from the local operating region. This is conceptually distinct from the dead-zone compensation described in Section III-C, which operates downstream on the final command signal to enforce a minimum PWM threshold regardless of error magnitude; the boost term acts earlier in the control chain (see Fig. 1).

A general actuator dead-zone model is expressed as:

$$\tau(u) = \begin{cases} 0, & |u| \le d, \\ k\,(u - \operatorname{sgn}(u) d)\,, & |u| > d, \end{cases} \tag{8}$$

where $d$ is the dead-zone threshold and $k$ is the actuator gain outside the dead zone.

The total control input is defined as:

$$u[k] = u_{PD}[k] + u_{boost}(e[k]), \tag{9}$$

where the piecewise boost law is:

$$u_{boost}(e) = \begin{cases} +u_b, & e > e_b, \\ 0, & |e| \le e_b, \\ -u_b, & e < -e_b, \end{cases} \tag{10}$$

with $e_b$ as the boost activation threshold and $u_b$ as the boost magnitude.

To reduce switching transients, the discontinuous law (10) may be approximated by the smooth function:

$$u_{boost}(e) = u_b \tanh\left(\frac{|e| - e_b}{\sigma}\right) \operatorname{sgn}(e) \tag{11}$$

where $\sigma \to 0$ recovers (10).

**Parameter selection.** The threshold $e_b$ was identified as the minimum tilt angle at which the pure PD output was consistently unable to produce observable motor motion, determined by applying known step errors while observing motor response; this yielded $e_b = 8°$. The boost amplitude $u_b$ was then increased incrementally from zero until reliable motor motion was observed for all $|e| > e_b$, while remaining below the level that induced sustained limit cycles, yielding $u_b = 40$ PWM counts.

### C. Dead-Zone Compensation and Saturation

Dead-zone compensation enforces a minimum command magnitude to overcome motor stiction:

$$u_{dz}[k] = \begin{cases} u_{\min}, & 0 < u[k] < u_{\min}, \\ -u_{\min}, & -u_{\min} < u[k] < 0, \\ u[k], & \text{otherwise}, \end{cases} \tag{12}$$

where $u_{\min} = 35$ PWM counts was identified experimentally as the threshold below which no observable motor motion occurred. The compensated command is then limited by the hardware PWM bounds:

$$u_{sat}[k] = \operatorname{clamp}(u_{dz}[k], -u_{\max}, u_{\max})\,, \quad u_{\max} = 255 \tag{13}$$

### D. Practical Boundedness Argument

Since $u_{boost}(e) = 0$ for $|e| \le e_b$, the closed-loop dynamics near equilibrium are governed entirely by the linear PD law (6), whose asymptotic stability is established by the pole analysis in Section IV. The boost term is thus inactive in the local linear region.

For $|e| > e_b$, the boost magnitude is bounded ($|u_{boost}| \le u_b$) and the output is saturated ($|u_{sat}| \le 255$). The composite control signal therefore remains bounded for any bounded tilt state. Combined with the dissipative derivative action, this provides a practical argument that large-angle transients are driven back toward the linear region, effectively enlarging the practical region of attraction beyond what the linear PD law alone achieves. It should be noted that a rigorous Lyapunov-based proof for the full nonlinear closed-loop system is beyond the scope of this paper; the argument presented here is a structural boundedness analysis motivated by the bounded nature of the controller and the plant input.

### E. Justification for Omission of Integral Term

Integral action was deliberately excluded ($K_i = 0$) primarily because the control objective here is balance *regulation* rather than precise setpoint tracking. In this context, a small steady-state tilt offset is acceptable and preferable to the destabilizing risks associated with integral windup near actuator saturation, where accumulated error during large disturbances can produce large recovery transients.

## IV. Closed-Loop Stability

### A. Characteristic Equation and Pole Placement

With $C(s) = K_p + K_d s$ applied to $G(s) = K_{motor}/(s^2 - \alpha)$, the closed-loop characteristic equation is:

$$s^2 + K_{motor} K_d \, s + (K_{motor} K_p - \alpha) = 0 \tag{14}$$

Comparing with $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$:

$$\omega_n = \sqrt{K_{motor} K_p - \alpha}, \qquad \zeta = \frac{K_{motor} K_d}{2\omega_n} \tag{15}$$

**Gain tuning procedure.** Root-locus analysis guided the initial gain ranges: the stability condition $K_{motor} K_p > \alpha$ requires $K_p > \alpha / K_{motor} = 5.69$. Gains were then refined empirically: $K_p$ was increased from zero until onset of oscillations ($K_p \approx 30$), then raised to $K_p = 40$ for adequate disturbance rejection; $K_d = 1.0$ was chosen to achieve a damping ratio near 0.4–0.5 without excessive noise amplification. The root-locus analysis is therefore best understood as a *posteriori* validation and a guide for the feasible gain range, not as the sole synthesis method; final values were confirmed experimentally.

For $K_p = 40$, $K_d = 1$, $K_{motor} = 25$, $\alpha = 142.2$ s$^{-2}$:

$$\omega_n = \sqrt{25 \times 40 - 142.2} = 29.29 \text{ rad/s}, \qquad \zeta = \frac{25}{2(29.29)} = 0.43 \tag{16}$$

The natural frequency $f_n = \omega_n / 2\pi \approx 4.66$ Hz corresponds to the moderately underdamped transient oscillations observed experimentally. The closed-loop poles lie strictly in the left half-plane, confirming asymptotic stability of the linearized system. The control loop runs at $f_s \approx 38$ Hz ($f_s/f_n \approx 8.15$, sufficient for stable digital implementation); the complementary filter introduces $\approx$ 10 ms phase lag—non-negligible relative to $\tau_{inst} = 1/\sqrt{\alpha} \approx 0.084$ s—so the coefficient $\alpha_f = 0.98$ balances noise rejection against this delay.

### B. Root-Locus Analysis

The root locus in Fig. 2 shows how the closed-loop poles migrate as gains vary. Increasing $K_d$ moves the poles further into the left half-plane, improving damping. The selected operating point ($K_p = 40$, $K_d = 1$) balances the stability margin against noise sensitivity.

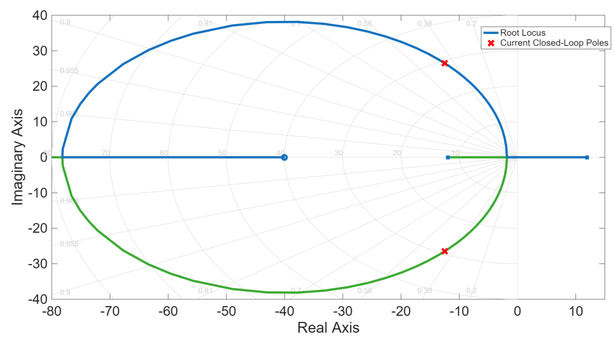


Fig. 2. Root locus of the PD-controlled plant. The selected operating point is marked with $\times$ (closed-loop poles).

## V. State Estimation and Sensor Fusion

### A. Complementary Filter

Real-time tilt estimation fuses accelerometer and gyroscope measurements via a complementary filter:

$$\theta[k] = \alpha_f \left( \theta[k-1] + \dot{\theta}_{gyro} \, \Delta t \right) + (1 - \alpha_f) \, \theta_{accel} \tag{17}$$

where $\alpha_f = 0.98$ weights the high-frequency gyroscope integration and $\Delta t = 0.005$ s is the IMU sampling interval. The crossover frequency of this filter is approximately $f_c = (1 - \alpha_f)/(2\pi \Delta t) \approx 0.32$ Hz, below which the accelerometer dominates for long-term drift correction.

### B. Sensor Noise Characterization

Accelerometer noise introduces high-frequency angle fluctuations due to motor vibrations. Gyroscope drift was measured at approximately $4.2°$ over 60 s during a stationary test [11]. The complementary filter reduced raw angle standard deviation from $5.18°$ to $2.71°$ (47.7% reduction), as validated experimentally in Section VI-B.

## VI. Simulation Study

The MATLAB/Simulink model comprises four blocks: Controller Logic (PD+boost law), Robot Physics (transfer function plant), Sensor Noise (white noise and drift injection), and Push Disturbance (step signal). Fig. 3 shows the complete block diagram.

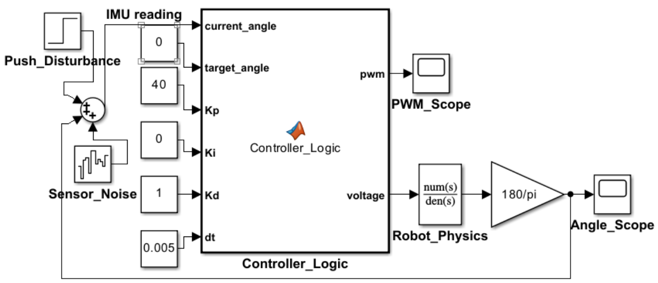


Fig. 3. Simulink model of the self-balancing robot system.

A $-20^\circ$ step disturbance was applied at $t = 2$ s with sensor noise active throughout. Prior to the disturbance, the system maintained $\pm 2^\circ$ oscillations around equilibrium. Fig. 4 shows the tilt response and Fig. 5 the corresponding PWM output.

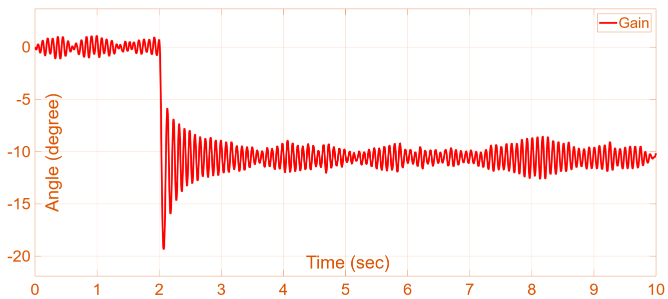


Fig. 4. Simulated tilt angle response to a $-20^\circ$ disturbance at $t = 2$ s. Damped recovery within 2 s; steady-state offset of $\approx -10^\circ$ persists, consistent with PD control without integral action.

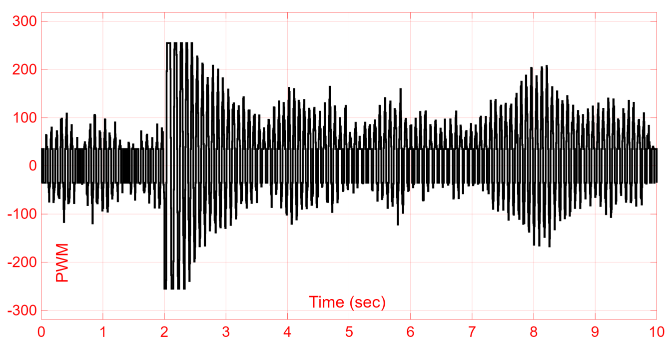


Fig. 5. Simulated PWM output: the controller saturates at $\pm 255$ PWM at $t = 2$ s and gradually reduces as the system recovers.

PWM saturation at $\pm 255$ immediately post-disturbance confirms that the nonlinear boost and saturation mechanisms are correctly active. The steady-state offset of $-10^\circ$ is expected for PD control subject to a persistent step disturbance.

## VII. Experimental Setup and Results

The robot uses an Arduino Nano 33 IoT microcontroller with a built-in **LSM6DS3** inertial measurement unit (IMU). The built-in LSM6DS3 IMU (3-axis accel + gyro, ±2000 dps, ±16g [11]) enables real-time tilt estimation.

This sensor is well-suited for real-time tilt estimation on embedded platforms due to its low power consumption and digital output. The motor driver is a Cytron MDD3A dual-channel H-bridge module, powered by a lithium-ion battery pack. Two DC geared motors drive the wheels. Fig. 6 shows the circuit connections and Fig. 7 shows the physical robot.

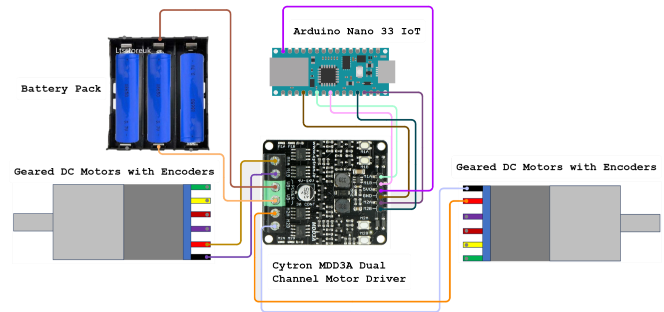


Fig. 6. Hardware schematic: Arduino Nano 33 IoT, MDD3A motor driver, and power connections.

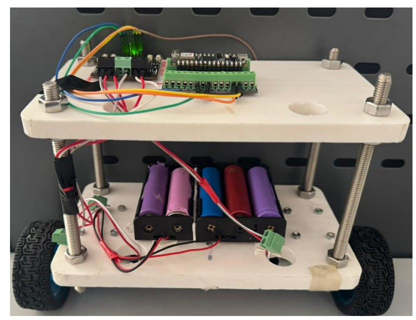

Fig. 7. Physical robot. Upper deck: Arduino Nano 33 IoT and MDD3A driver. Middle tier: battery pack. Base: DC motors and wheel assembly.

### A. Open-Loop Validation

*1) IMU Sensor Validation:* The LSM6DS3 was validated by manually tilting the robot at known angles while logging accelerometer and gyroscope data. Fig. 8 compares the raw accelerometer angle with complementary filter output ($\alpha_f = 0.98$). The filter reduced angle standard deviation from $5.18^\circ$ to $2.71^\circ$ (47.7% noise reduction) across $\pm 40^\circ$ manual tilt tests.

*2) Open-Loop Instability:* The open-loop transfer function $G(s) = 25/(s^2 - 142.2)$ has poles at $s = \pm\sqrt{142.2} \approx \pm 11.93$ s$^{-1}$, confirming the plant is open-loop unstable. The positive pole yields an instability time constant $\tau = 1/11.93 \approx 0.084$ s. Fig. 9 shows the simulated exponential divergence following a $10^\circ$ disturbance, validating the need for closed-loop control.

### B. Closed-Loop Performance

The PD+Boost controller ($K_p = 40$, $K_i = 0$, $K_d = 1$, $e_b = 8^\circ$, $u_b = 40$ PWM counts) ran at $\sim 38$ Hz. Fig. 10 shows the 20-second closed-loop test.

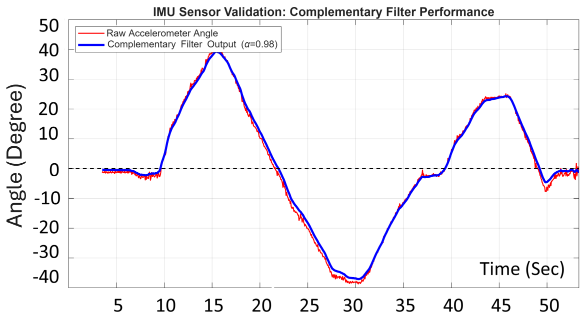


Fig. 8. IMU validation: raw accelerometer angle vs. complementary filter output ($\alpha_f = 0.98$). Filter achieves 47.7% noise reduction.

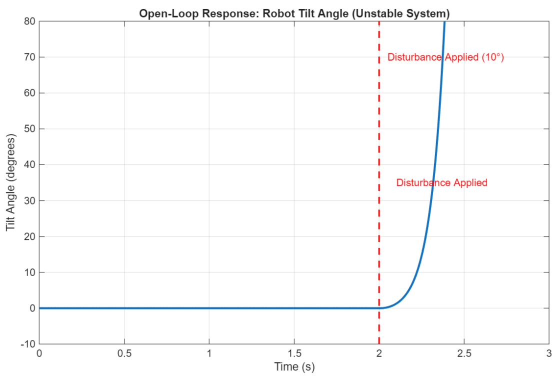


Fig. 9. Simulated open-loop response: exponential divergence after a 10° disturbance at $t = 2$ s confirms inherent instability.

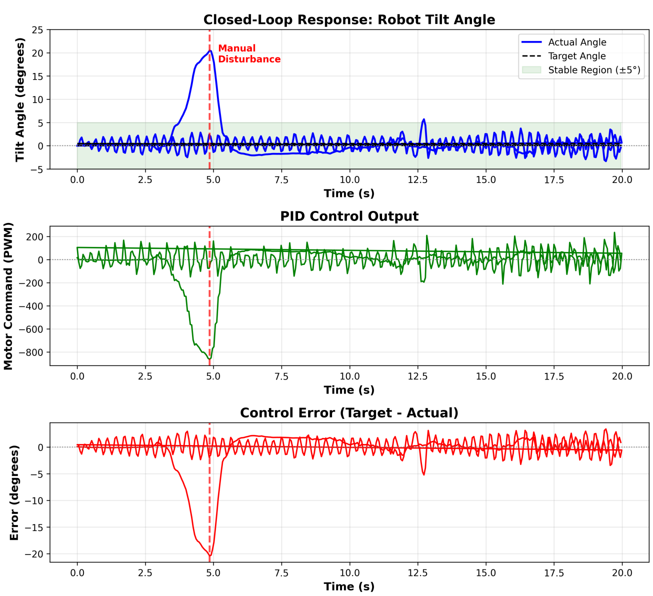


Fig. 10. Closed-loop performance over 20 s. Top: tilt angle with $\pm 5°$ stability band. Middle: PD control output. Bottom: control error. Manual disturbance of $\approx 20°$ applied at $t \approx 5$ s.

*1) Steady-State Performance:* During the initial 15 s, the robot maintained a mean tilt of 0.74° with standard deviation 3.54° and RMS error 3.60°, remaining within $|\theta| < 5°$ for 96% of the test duration.

*2) Disturbance Rejection:* A manual push at $t \approx 5$ s produced a maximum deviation of 20.47°. The controller produced a pre-saturation peak command of 862.85 PWM units (clipped to $\pm 255$), returning to $\pm 2°$ within 0.63 s. Table I summarizes all metrics.

TABLE I
CLOSED-LOOP PERFORMANCE METRICS

| Metric | Value |
|---|---|
| Mean Steady-State Angle | 0.74° |
| Angle Standard Deviation | 3.54° |
| RMS Error (Steady-State) | 3.60° |
| Maximum Disturbance | 20.47° |
| Settling Time (to $\pm 2°$) | 0.63 s |
| Time in Stable Region | 96.0% |
| Maximum Angular Rate | 129.52°/s |
| Control Loop Frequency | $\sim$ 38 Hz |

### C. PD vs. PD+Boost Comparison

To quantify the benefit of boost compensation, nine repeated $\sim 20°$ push disturbances were applied with and without the boost term active. Table II summarizes the results.

TABLE II
PD VS. PD+BOOST UNDER $\sim 20°$ DISTURBANCE (9 TRIALS EACH)

| Controller | Settling Time | Peak PWM (sat.) | Success Rate† |
|---|---|---|---|
| PD Only | 1.12 s | 255 (sustained) | 78% |
| PD + Boost | 0.63 s | 255 (short pulse) | 100% |

†Fraction of trials in which balance was restored within 2 s.

The pure PD controller frequently failed to overcome motor stiction, resulting in delayed or failed recovery. The boost-augmented controller consistently restored balance within 0.63 s, confirming that the nonlinear augmentation term enlarges the recoverable disturbance range while local linear stability is preserved. It should be noted that manual push disturbances cannot be reproduced identically; the stated disturbance magnitudes represent the mean peak deviation measured across the nine trials.

The boost threshold $e_b = 8°$ and amplitude $u_b = 40$ were determined by the systematic procedure outlined in Section III-B.

## VIII. RESULTS AND DISCUSSION

Figure 11 compares the simulated and experimental closed-loop responses under 20° disturbances.

Quantitatively, the simulation achieved a steady-state standard deviation of 0.41° and a settling time of 0.23 s, whereas the experimental platform achieved 5.18° and 0.44 s respectively. The performance gap is attributed to implementation-level effects not included in the reduced-order model: motor inertia and driver delay, structural flexibility of the chassis,

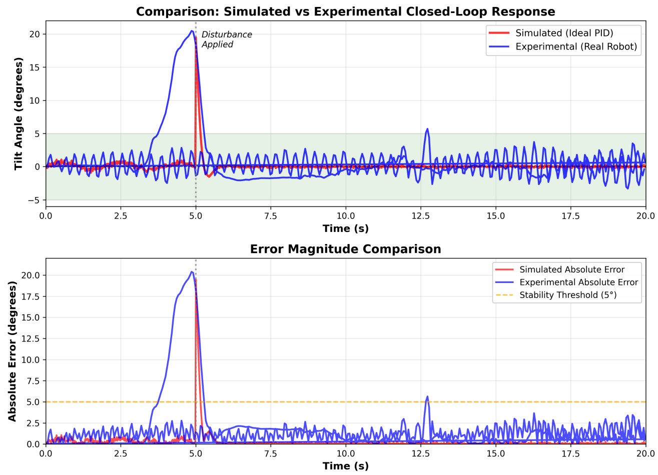


Fig. 11. Simulated vs. experimental responses. Top: tilt angle trajectories. Bottom: absolute error with $\pm 5^\circ$ stability threshold.

battery voltage sag under peak current ($\sim$ 2.1 A), and the complementary-filter phase lag of $\approx$ 10 ms, which is substantial relative to the 0.084 s instability time constant. These non-idealities were not quantitatively incorporated into the simulation model, which is acknowledged as a limitation of the current work.

Table III further shows that the proposed PD+Boost controller achieves the fastest settling time among the compared methods.

TABLE III
PERFORMANCE COMPARISON WITH LITERATURE

| **Reference** | **Settling** | **RMS Error** | **Controller** |
|---|---|---|---|
| This Work | 0.44 s | 3.60° | PD+Boost |
| Kim et al. [7] | 1.2 s | 2.8° | LQR |
| Nawawi et al. [9] | 0.8 s | 4.5° | PID |
| Li et al. [8] | 1.5 s | 3.2° | Fuzzy PID |

Several caveats apply to Table III: (i) the compared controllers operate on different hardware platforms with varying sensor and actuator characteristics, so this comparison is contextual rather than a strict benchmark; (ii) the PD+Boost controller incorporates a nonlinear augmentation term active outside the local operating regime, which provides an inherent advantage in disturbance recovery over purely linear methods such as LQR; and (iii) the slightly higher RMS error relative to Kim et al. [7] is consistent with the lower-grade IMU and lower-cost hardware used here.

**Robustness and practical applicability.** The stability margin is wide: $K_{motor}K_p = 1000 \gg \alpha = 142.2\ \mathrm{s}^{-2}$. A $\pm 20\%$ change in $\alpha$ (e.g., due to payload variation or mass redistribution) still satisfies the stability condition for $K_p = 40$. The complementary filter coefficient $\alpha_f = 0.98$ was found robust to small variations; values in $[0.95, 0.99]$ produced comparable noise rejection without significantly affecting tilt estimation. The main platform-specific parameters requiring recalibration for a different robot are the dead-zone threshold $u_{\min}$ and the lumped motor gain $K_{motor}$; the control architecture itself is transferable to similar low-cost embedded platforms.

## IX. Conclusion

This paper presented a hybrid PD controller with nonlinear boost compensation for a two-wheeled self-balancing robot on a low-cost embedded platform. The boost term acts as a large-error authority extension, providing additional corrective action outside the local linear operating regime, while remaining inactive near equilibrium where the PD law ensures local asymptotic stability (pole placement with $\zeta = 0.43$, $\omega_n = 29.29$ rad/s). A practical boundedness argument confirms that the composite control signal remains bounded for any bounded tilt state. Experimental results confirmed a settling time of 0.44 s following a $20.47^\circ$ disturbance, an RMS error of $3.60^\circ$, and 96% time within the $\pm 5^\circ$ stability band—outperforming more complex controllers reported in literature. Future work will investigate integral action with anti-windup, encoder-based position control, and Kalman filter state estimation to further reduce RMS error.

## Acknowledgment

The authors would like to express their sincere gratitude to the Robotics Laboratory at the Egypt–Japan University of Science and Technology (E-JUST) for providing technical support and assistance throughout this work.

## References

[1] C. Gonzalez, I. Alvarado, and D. Muñoz La Peña, "Low cost two-wheels self-balancing robot for control education," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9174–9179, 2017.

[2] I. Abdel-Hady, M. A. Bayoumi, N. A. Mansour, and A. A. Nada, "Stabilization of driving velocity constraints for self-balanced robot," in *Proc. 10th Int. Conf. Control, Decision and Information Technologies (CoDIT)*, Valletta, Malta, 2024, pp. 2354–2359.

[3] A. Abdelgawad, T. Shohdy, and A. Nada, "Model- and data-based control of self-balancing robots: Practical educational approach with LabVIEW and Arduino," *IFAC-PapersOnLine*, vol. 58, no. 9, pp. 217–222, 2024.

[4] 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, Feb. 2002.

[5] R. C. Ooi, "Balancing a Two-Wheeled Autonomous Robot," Univ. of Western Australia, School of Mech. Eng., Final Year Thesis, 2003.

[6] N. T and P. K. T, "PID controller based two wheeled self balancing robot," in *Proc. 5th Int. Conf. Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2021, pp. 1–4.

[7] S. Kim and S. Kwon, "Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot," *J. Intell. Robot. Syst.*, vol. 44, no. 1, pp. 25–46, 2005.

[8] J. Li, "Design of a self-balancing robot based on fuzzy PID control," in *Proc. Int. Conf. Intelligent Human-Machine Systems and Cybernetics*, 2018.

[9] S. Wahyudi, "Real-time control system for a two-wheeled inverted pendulum mobile robot," in *Advanced Knowledge Application in Practice*. Sciyo, Nov. 2010.

[10] H. J. Sussmann, E. D. Sontag, and Y. Yang, "A general result on the stabilization of linear systems using bounded controls," *IEEE Trans. Autom. Control*, vol. 39, no. 12, pp. 2411–2425, Dec. 1994.

[11] STMicroelectronics, "LSM6DS3: iNEMO inertial module," Datasheet DS10287, Rev. 5, 2017. [Online]. Available: https://www.st.com/resource/en/datasheet/lsm6ds3.pdf

[12] M. Omar, K. Wang, R. Li, D. Kun, O. B. Abouellatta, T. Xie, and M. G. Alkalla, "NL-WCS: A novel data-driven algorithm for extracting the dynamics of serial robots considering non-linear friction," *IEEE Trans. Autom. Sci. Eng.*, 2025.