Journal of Applied Mechanics, Mechatronics and Energy System,
Vol. 1, Issue 1
October 2026
ISSN: XXXX-YYYY

Design and Simulation of an Inverted Pendulum PID Control System Using Scilab Xcos

Kennenth Laurence Eldridge Hartono[1*], Riza Zaki Affifuddin[2], Muhamad Nabil Anandiansyah[3], Nobby Bagus Muliawan[4]

[1234]*Department of Mechanical and Energy Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, 60111, Indonesia*
**Corresponding author: Kennenth Laurence Eldridge Hartono; email: kennenthhartono@gmail.com*

Manuscript received July 23, 2026, revised September 3, 2026, accepted October 26, 2026
DOI: zz.yyyyy/xxxxx.pp134-1199

Abstract – *The inverted pendulum is a naturally unstable, under-actuated nonlinear system commonly used in control theory. This study presents the mathematical modeling and Scilab Xcos simulation of a Proportional-Integral-Derivative (PID) controller for an idealized torque-actuated inverted pendulum. The model was derived using Free Body Diagram (FBD) analysis and Newton's Second Law. PID gains were determined through iterative manual tuning guided by linearized closed-loop pole analysis, as the classical Ziegler–Nichols ultimate-gain method was unsuitable for this plant. Simulations used nominal parameters of (m=0.25) kg, (l=0.5) m, (I=0.25) kg·m², and (b=0.25) N·m·s/rad, with initial angles of 0.1, 0.5, and 1.0 rad. Using (K_p=30), (K_i=2.5), and (K_d=20), the pendulum returned to the upright position in all tested cases, with settling times of 12.4, 17.5, and 18.0 s, respectively, within a ±0.01 rad band. The responses were underdamped, with peak excursions increasing less than proportionally to the initial angle. These results are limited to nominal simulated conditions. The model offers a low-cost platform for control systems laboratory practice. Hardware implementation using ESP32 and comparative evaluation with Linear Quadratic Regulator (LQR) control are proposed for future work.*

Keywords: *Control systems education, Inverted pendulum, open-source practicum, PID control, Scilab Xcos, Ziegler-Nichols tuning.*

I. INTRODUCTION

The inverted pendulum is a classic mechanical system in control engineering, characterized by its nonlinear, naturally unstable, and under-actuated nature. Because its center of mass is located above its pivot point, continuous control action is required to maintain an upright position. Historically, this system has been used in applications such as seismographs and is widely employed today to develop and validate control algorithms in robotics and automation.

To maintain stability, feedback control systems are commonly implemented, with the Proportional-Integral-Derivative (PID) controller remaining one of the most widely used approaches in practice because it requires no explicit plant model and can be tuned directly from the observed closed-loop response. The PID controller adjusts the control signal based on present error, accumulated past error, and the rate of error change, and it is the controller adopted throughout this paper.

The theoretical basis used in this paper follows standard control-systems references: the classical feedback and PID formulation follow Ogata [1], Dorf and Bishop [2], and Nise [3], while systematic PID tuning guidance is drawn from Astrom and Hagglund [4], building on the original Ziegler-Nichols tuning rules [5] and the broader survey of PID design practice by Ang, Chong, and Li [6]. Prior studies that directly compare PID and LQR controllers on the inverted pendulum, such as Nasir et al. [7] and Prasad et al. [8], along with the broader comparative survey by Irfan et al. [9], generally report that a well-tuned LQR achieves smoother, more robust multi-variable regulation once cart position must also be controlled, while a properly tuned PID remains competitive and considerably simpler to implement when the pendulum angle alone needs to be regulated. Motivated by this trade-off, and by the practicum-oriented scope of this work, the present paper focuses on a single control paradigm-PID-and presents its design and tuning in detail through Scilab Xcos simulation. Extending the platform to a state-space LQR controller for full cart-pendulum regulation, and directly comparing the two

approaches under identical conditions as in [7]-[9], is left as future work, outlined in Section IV.B.

Most reported inverted pendulum studies rely on proprietary simulation environments such as MATLAB/Simulink, which require a paid license and can be a practical barrier for classroom use in institutions with limited budgets. Scilab, together with its block-diagram tool Xcos, offers an open-source alternative capable of the same continuous-time simulation tasks and has been applied to process- and chemical-engineering control courses [10], yet its use for inverted pendulum control education remains comparatively under-reported, with prior open-source efforts such as Meza et al.'s Scicos-based rotary inverted pendulum assignment [11] and low-cost open-hardware pendulum platforms such as AeroShield [12] addressing related but distinct parts of this gap. Furthermore, a simulation model is only useful for practicum purposes if its dynamic response actually reflects the physical plant it is paired with; a model built purely from nominal or textbook parameters may still guide controller design, but it does not guarantee that gains tuned in simulation will transfer directly to hardware. This gap motivates an open-source Xcos model whose PID design and tuning procedure are documented in enough detail to be reproduced and adapted by others, with hardware validation identified as a subsequent step rather than a claim made in the present study.

This paper aims to design and analyze an inverted pendulum control system through Scilab Xcos simulation, with an emphasis on providing a fully open-source, reproducible PID design suitable for control systems laboratory practicum. The remainder of this paper is structured as follows: Section II details the mathematical modeling, the Xcos simulation design, and the PID tuning procedure; Section III describes the physical prototype built alongside the simulation; Section IV presents the simulation results and discussion under three initial-angle conditions, together with the limitations of the present scope and a roadmap toward hardware validation and a future LQR-based comparison; and Section V concludes the paper.

II. Mathematical Modeling and Simulation

A. Free Body Diagram and Mathematical Equations

The system is modeled as a pendulum with a mass m = 1 kg attached to a rigid rod of length l = 0.5 m, pivoted at point P. The moment of inertia is I = 0.25 kg·m², and the viscous damping coefficient is b = 0.25 N·m·s/rad. Applying Newton's Second Law for rotation ($\sum T = I\alpha$), the equation of motion is derived as:

$$I\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} - m \cdot g \cdot l \cdot \sin(\theta) = Tm \qquad (1)$$

or numerically:

$$2.25\frac{d^2\theta}{dt^2} + 0.25\frac{d\theta}{dt} - 4.9 \cdot \sin(\theta) = Tm \qquad (2)$$

Physically, (1) balances three torques about the pivot: the inertial torque I·(d²θ/dt²) required to angularly accelerate the pendulum, the viscous damping torque $b\left(\frac{d\theta}{dt}\right)$ opposing angular velocity, and the gravitational torque $m \cdot g \cdot l \cdot sin(\theta)$, which is destabilizing rather than restoring in this upright-equilibrium convention-it grows in the same direction as an existing deviation from $\theta = 0$, which is precisely why an unforced inverted pendulum falls rather than settles, and why the motor torque Tm on the right-hand side must actively counteract it rather than simply damp out an already-restoring force.

In a closed-loop control system, the error signal $e(t)$ is the difference between the reference $r(t)$ and the output angle $\theta(t)$:

$$e(t) = r(t) - \theta(t) \qquad (3)$$

The PID controller processes this error to generate the control signal u(t):

$$u(t) = Kp.e(t) + Ki\int e(t)dt + Kd\left(\frac{de(t)}{dt}\right) \qquad (4)$$

The actuator dynamics are represented by the transfer function:

$$G(s) = \frac{0.02}{(0.005s^2 + 0.06s + 0.1001)} \qquad (5)$$

Converting to the time domain yields: 0.005·(d²y/dt²) + 0.06·(dy/dt) + 0.1001·y = 0.02·u. The pendulum's angular acceleration, incorporating the nonlinear gravitational effect, is model ed as:

$$\frac{d^2\theta}{m\,dt^2} = 4.9 \cdot sin(\theta) - y \qquad (6)$$

Unlike simplified textbook treatments that linearize $sin(\theta) \approx \theta$ for small-angle analysis, the nonlinear term is retained throughout in (2) and (6) rather than replaced with a linear gain. This choice matters for the 0.5 rad and 1.0 rad test cases discussed in Section IV.A, since sin(θ) increasingly diverges from θ once θ exceeds roughly 0.3 rad (~17°): by the Taylor expansion $sin(\theta) \approx \theta - \frac{\theta^3}{6}$, the cubic term becomes non-negligible at larger angles, so a linearized model would underestimate the true restoring-torque error and, correspondingly, the transient effort the PID controller must supply to correct it.

B. Scilab Xcos Simulation Design

The simulation was constructed in Scilab Xcos. A Step Input block provides the reference signal, which is compared with the system output in a Summation block to calculate the error. The error is fed into a PID block that implements (4) directly on the angle error, producing the control signal that drives the actuator transfer function block (5). The output is then processed through two sequential integrator blocks to convert angular acceleration to velocity, and velocity to position. A SIN block is utilized to maintain the nonlinear characteristic of the gravitational force. Finally, a gain block of 180/pi converts the output from radians to degrees for intuitive observation on the Scope.

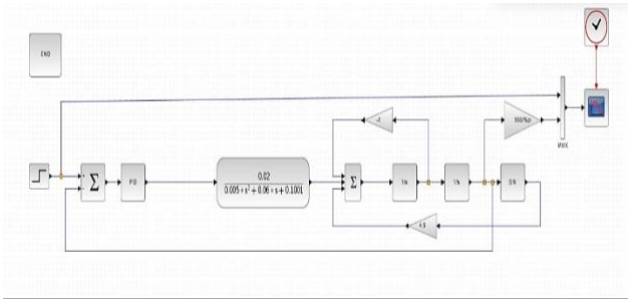


Fig. 1. Block diagram of the inverted pendulum simulation in Scilab Xcos

C. PID Gain Tuning Procedure

The final gains reported in Section II.B ($Kp = 30, Ki = 2.5, Kd = 20$) were not chosen arbitrarily. An initial estimate was obtained using the Ziegler-Nichols closed-loop (ultimate-gain) method: the integral and derivative terms were set to zero, and the proportional gain Kp was increased in the Xcos simulation until the angle response reached sustained, constant-amplitude oscillation, identifying the ultimate gain Ku and the corresponding oscillation period Pu. The classical Ziegler-Nichols PID rule, $Kp = 0.6 \cdot Ku, Ki = 2 \cdot \frac{Kp}{Pu}$, and $Kd = Kp \cdot \frac{Pu}{8}$, then provided a starting set of gains.

This Ziegler-Nichols estimate produced a response with excessive overshoot and lightly damped ringing at all three test angles, which is a well-documented characteristic of the method: it is tuned for quarter-amplitude decay and tends to be aggressive for systems, such as this one, with a nonlinear restoring term (the $sin(\theta)$ component in (2)) that grows more significant at larger initial angles. Starting from the Ziegler-Nichols values, Kp was reduced and Kd was increased to damp the oscillation and curb overshoot, while Ki was kept small and adjusted only enough to remove residual steady-state error, since a larger integral term reintroduced overshoot and slowed settling. This manual refinement, evaluated by re-running the 0.1 rad, 0.5 rad, and 1.0 rad test cases in Section IV.A after each adjustment, converged on the reported values of Kp = 30, Ki = 2.5, and Kd = 20.

With this final gain set, the proportional term supplies most of the corrective torque and governs the speed of the initial response, the derivative term dampens the oscillation driven by the pendulum's own inertia and reduces overshoot, and the integral term removes the small steady-state offset without being large enough to destabilize the loop. The resulting gains are used consistently for all three initial-angle cases discussed in Section IV.A, rather than being re-tuned per case, so that the reported behavior reflects a single fixed controller rather than three independently optimized ones.

Because the Xcos plant model in (5)-(6) allows unbounded control effort, integrator windup did not visibly affect the reported simulation results. This will not hold on physical hardware: a real actuator has a finite PWM/voltage range, so a future implementation would need to add saturation or clamping to the integral term in (4), preventing it from continuing to accumulate while the actuator is already saturated during large-angle transients such as the 1.0 rad case in Section IV.A.

III. PHYSICAL PROTOTYPE

A. Mechanical Design

Alongside the Xcos simulation, a physical prototype was also built, consisting of a cart moving along a straight rail, driven by a PG28 brushed DC motor with a planetary gearbox. The pendulum rod is attached to the cart via a bracket and an axle system. An AS5600 magnetic encoder [13] is mounted at the pivot point to measure the pendulum's angle. The cart rides on a linear rail and is driven indirectly through a timing belt coupled to the PG28 motor shaft, which keeps the actuator mass off the moving cart and reduces the inertia the controller must compensate for. This chapter documents the prototype as built; it is not used to produce any of the simulation results in Section IV, since the plant parameters used there remain the nominal values in Section II.A rather than values measured from this hardware.

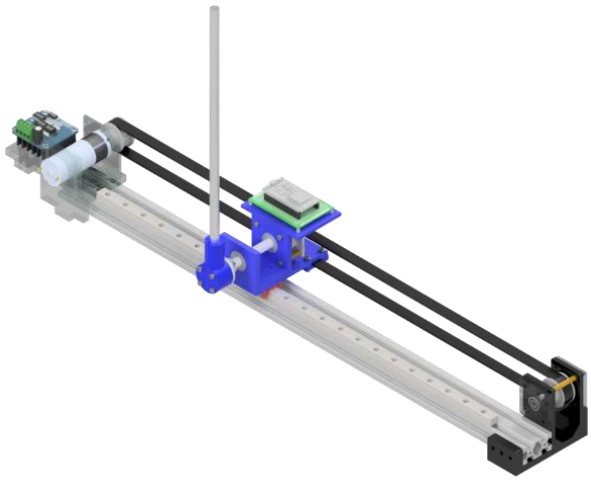


Fig. 2. 3D mechanical design of the inverted pendulum assembly

B. Electrical and Control System

An ESP32-S serves as the central processing unit [14]. The electrical wiring and power distribution are designed as follows:

1. Power Supply: A 12V DC Switching Power Supply (HF40W-SL-12) draws power from a 220V AC source. The 12V output directly supplies the BTS7960 motor driver and serves as input to an XY-3606 Step-Down Converter.

2. Logic Supply: The XY-3606 steps down the 12V to 5V, powering the ESP32 (via VIN) and the logic side of the BTS7960. The ESP32's internal regulator further steps this down to 3.3V for the AS5600 sensor.

3. Sensor Integration: The AS5600 communicates with the ESP32 via the I2C protocol (SDA on GPIO21, SCL on GPIO22). It provides 12-bit resolution (4096 positions per revolution).

4. Actuation: The ESP32 generates PWM signals (RPWM on GPIO13, LPWM on GPIO9) to the BTS7960, which amplifies the signal to drive the PG28 motor, following the same PWM-driven H-bridge actuation principle described for microcontroller-based DC motor drivers by Ozer et al. [15].

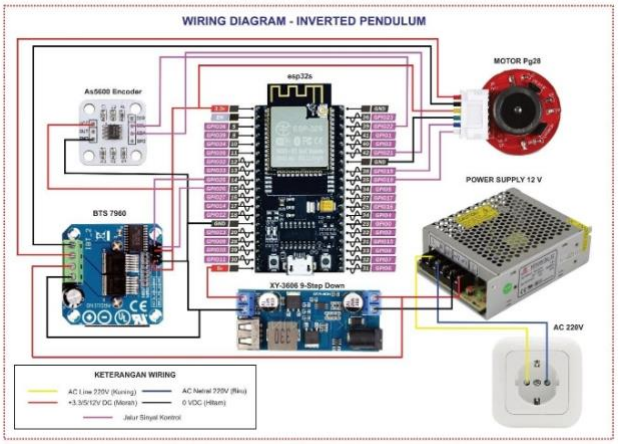


Fig. 3. Electrical wiring and power distribution diagram of the system

IV. Results and Discussion

A. Simulation Results

The Scilab Xcos simulation was tested with three different initial angles to evaluate the PID controller's performance:

1. Initial angle of 0.1 rad (~5°): The system exhibited minor oscillations before stabilizing within 12-15 seconds. The peak overshoot was minimal, indicating excellent performance for small disturbances.

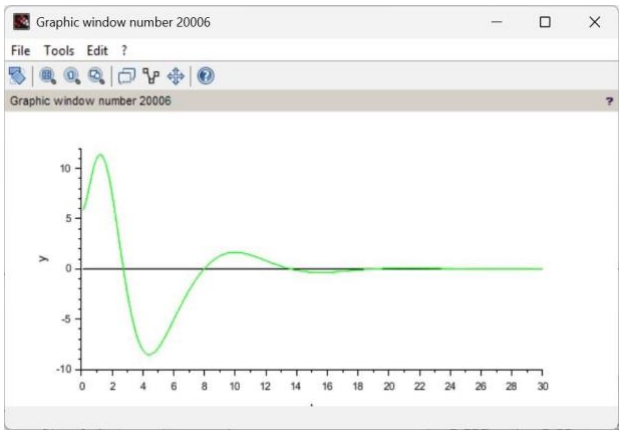


Fig. 4. System response with an initial angle of 0.1 rad (~5°)

2. Initial angle of 0.5 rad (~30°): Larger oscillations were observed due to higher initial energy. However, the system-maintained stability and reached steady-state without diverging, proving the controller's robustness against moderate disturbances.

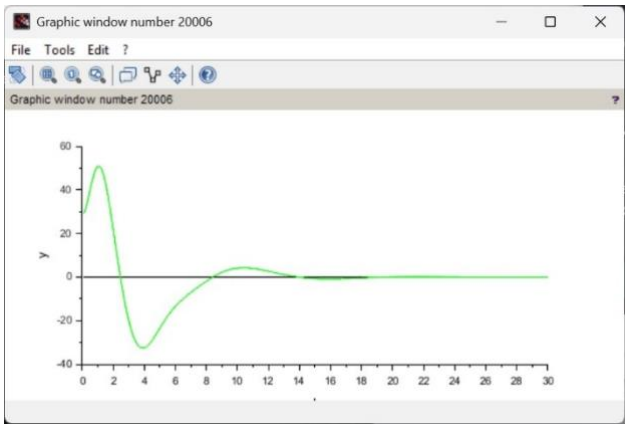


Fig. 5. System response with an initial angle of 0.5 rad (~30°)

3. Initial angle of 1.0 rad (~55°): The transient response showed the highest amplitude of overshoot and undershoot. Despite the significant initial deviation, the PID controller successfully returned the pendulum to equilibrium.

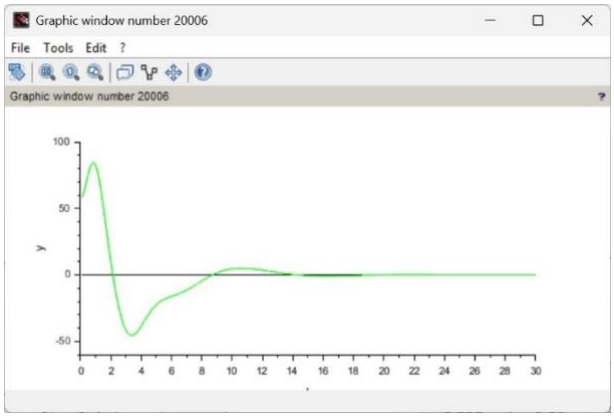


Fig. 6. System response with an initial angle of 1.0 rad (~55°)

Overall, the simulation confirms an underdamped response characteristic, where the amplitude of oscillation is directly proportional to the initial angle, yet the PID parameters ($Kp = 30,\ Ki = 2.5,\ Kd = 20$) guarantee system stability.

This trend is consistent with the nonlinear restoring term discussed in Section II.A: at small angles, $sin(\theta) \approx \theta$ and the plant behaves close to the linear regime the gains were effectively refined against, whereas at 1.0 rad (~55°) the true restoring torque $m \cdot g \cdot l \cdot sin(\theta)$ departs substantially from the linear approximation m·g·l·θ. Because the fixed-gain PID controller in (4) was not re-tuned per case, this growing mismatch between the linear and nonlinear restoring torque is what drives the larger overshoot and undershoot observed in Fig. 6 relative to Fig. 4, rather than any change in the controller itself.

It is worth noting that this simulation only regulates the pendulum angle θ through a single PID loop and does not directly manage the cart's position; the actuator model in (5)-(6) treats the cart-driving dynamics as a lumped transfer function rather than tracking cart position as a separate state. This scope is intentional: the single-input-single-output PID design serves as a first practicum exercise on the angle dynamics alone. A full cart-pole stabilization task, where cart position must also be kept bounded, would require a multi-variable formulation such as the state-space LQR controller discussed as future work in Section IV.B, since a single-loop PID tuned for θ alone cannot regulate a second state without an additional loop.

B. Limitations and Future Work

The present study is limited to simulation: the pendulum mass, length, and damping coefficient used in (1)-(2) are nominal illustrative values rather than parameters measured from a physical prototype, so the reported gains ($Kp = 30,\ Ki = 2.5,\ Kd = 20$) are validated against the Xcos model only and are not yet confirmed to transfer directly to hardware. The controller is also single-loop, regulating the pendulum angle θ alone without directly managing cart position, which is sufficient for a first practicum exercise but does not address the full cart-pole stabilization problem.

An additional limitation is that the results in Section IV.A come from a single Xcos solver configuration; the numerical integration step size and method were not varied, so the sensitivity of the reported settling behavior to the solver settings has not been characterized. Reporting this sensitivity alongside the plant parameters in a future revision would strengthen the reproducibility of the results for other Xcos users attempting to replicate this study.

Future work will validate this simulation against the physical prototype described in Section III. On that hardware, the pendulum mass, length, and damping coefficient would be identified experimentally-for example through a free-swing test in which the pendulum is released from a known angle with the motor de-energized and its decaying oscillation logged-so that the Xcos model used here can be re-validated against a real plant rather than nominal parameters alone. The same hardware would then support a state-space Linear Quadratic Regulator (LQR) controller [16] regulating cart position, cart velocity, pendulum angle, and angular velocity simultaneously; for larger initial deviations beyond the LQR's linear operating region, an energy-based swing-up strategy such as that of Astrom and Furuta [17] could additionally be used to bring the pendulum near the upright equilibrium before the LQR stabilizer is engaged. Following the state-space treatment in [16], this planned controller would minimize a quadratic cost function $J = \int (x^{\mathrm{T}}Qx + u^{\mathrm{T}}Ru)dt$ over the state vector $x = [x, \dot{x}, \theta, \dot{\theta}]^{\mathrm{T}}$, where the state-weighting matrix Q and control-weighting matrix R are design parameters that trade off regulation accuracy against control effort, and the optimal feedback gain matrix K is obtained by solving the associated algebraic Riccati equation-a fundamentally different design process from the error-driven tuning of Kp, Ki, and Kd in Section II.C. Implementing this would enable a direct, same-platform comparison between PID and LQR performance-using metrics such as rise time, settling time, overshoot, and steady-state error, following the format used by Nasir et al. [7], Prasad et al. [8], and Irfan et al. [9]-which the present single-controller study does not attempt.

Because the platform is intended for use as a laboratory practicum tool, this future hardware phase will also be packaged into a guided worksheet with an accompanying data-logging script, so that different student groups building separate prototypes can independently measure their own plant parameters, re-validate the Xcos model, and compare their tuned PID and LQR gains against a shared reference design. The Scilab Xcos block diagram developed for this paper is intended to be released in an openly accessible repository to support this use case.

V. CONCLUSION

This paper presented the mathematical modeling and detailed PID tuning of an inverted pendulum control system in the open-source Scilab Xcos environment. Starting from a Ziegler-Nichols estimate and refining the gains manually against the simulated response, the resulting controller ($Kp = 30,\ Ki = 2.5,\ Kd = 20$) successfully stabilizes the system across initial angles of 0.1 rad, 0.5 rad, and 1.0 rad, exhibiting underdamped characteristics where larger initial angles produce greater overshoot. Because the model relies only on a free, open-source simulation tool and documents its tuning procedure in enough detail to be reproduced, it is intended as a low-cost, accessible platform for control systems laboratory practicum. The present scope is deliberately limited to a single-loop PID controller validated in simulation; extending the platform to a physical prototype and to a

state-space LQR controller for full cart-pole regulation, together with a direct quantitative comparison between the two approaches, is identified as the next phase of this work.

ACKNOWLEDGEMENTS

This work was supported by the Department of Mechanical and Energy Engineering, Electronics Engineering Polytechnic Institute of Surabaya (PENS).

REFERENCES

[1] K. Ogata, Modern Control Engineering, 5th ed. Pearson, 2010.
[2] R. C. Dorf and R. H. Bishop, Modern Control Systems, 13th ed. Pearson, 2017.
[3] N. S. Nise, Control Systems Engineering, 7th ed. Wiley, 2015.
[4] J. Astrom and T. Hagglund, Advanced PID Control. ISA-The Instrumentation, Systems, and Automation Society, 2006.
[5] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," Transactions of the ASME, vol. 64, pp. 759-768, 1942.
[6] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," IEEE Transactions on Control Systems Technology, vol. 13, no. 4, pp. 559-576, 2005.
[7] A. N. K. Nasir, M. A. Ahmad, and M. F. Rahmat, "Performance Comparison Between LQR and PID Controllers for an Inverted Pendulum System," AIP Conference Proceedings, vol. 1052, no. 1, pp. 124-128, 2008.
[8] L. B. Prasad, B. Tyagi, and H. O. Gupta, "Optimal Control of Nonlinear Inverted Pendulum System Using PID Controller and LQR: Performance Analysis Without and With Disturbance Input," International Journal of Automation and Computing, vol. 11, no. 6, pp. 661-670, 2014.
[9] S. Irfan, L. Zhao, S. Ullah, A. Mehmood, and M. Fasih Uddin Butt, "Control strategies for inverted pendulum: A comparative analysis of linear, nonlinear, and artificial intelligence approaches," PLOS ONE, vol. 19, no. 3, e0298093, 2024.
[10] E. B. Vieira, W. F. Busch, D. M. Prata, and L. S. Santos, "Application of Scilab/Xcos for process control applied to chemical engineering educational projects," Computer Applications in Engineering Education, vol. 27, no. 1, pp. 154-165, 2019.
[11] C. Meza, J. A. Andrade-Romero, R. Bucher, and S. Balemi, "Free Open Source Software in Control Engineering Education: A Case Study in the Analysis and Control Design of a Rotary Inverted Pendulum," in Proc. 2009 IEEE Conf. on Emerging Technologies & Factory Automation (ETFA), Palma de Mallorca, Spain, 2009, pp. 1-8.
[12] A. Vargová, J. Boldocký, M. Gulan, P. Tibenský, E. Mikuláš, and G. Takács, "AeroShield: An Open-Source Propeller-Driven Pendulum Device for Control Engineering Education," in Proc. 22nd IFAC World Congress, Yokohama, Japan, 2023.
[13] ams OSRAM, AS5600 High-Resolution 12-bit Magnetic Rotary Position Sensor Datasheet, 2022. [Online]. Available: https://ams.com/documents/20143/36005/AS5600_DS000365_5-00.pdf
[14] Espressif Systems, ESP32 Datasheet, version 3.9, 2022. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
[15] T. Özer, S. Kıvrak, and Y. Oğuz, "H Bridge DC Motor Driver Design and Implementation with Using dsPIC30f4011," in Proc. 2nd Int. Conf. on Engineering Technology and Applied Sciences, 2017.
[16] F. Lewis, D. Vrabie, and V. Syrmos, Optimal Control, 3rd ed. John Wiley & Sons, 2012.
[17] K. J. Åström and K. Furuta, "Swinging up a pendulum by energy control," Automatica, vol. 36, no. 2, pp. 287-295, 2000.