Evaluating Robotics Feedback Loops (PID, LQR, Hybrid): How Accurate and Efficient are they?

Rohan Nagale

rohnagale@gmail.com

Lyons Township High School

STEM Research

Michelle Harbin

May 14, 2026

Introduction

Robotic systems are widely used across industries and have many applications. Whether they are autonomous or manned systems, though, they rely heavily on their stability, control, and precision. To meet these three criteria, robot systems commonly use feedback control loops to provide precision and control for advanced multi-degree-of-freedom robots. Through complex vector calculus, physics, and differential equations, computer algorithms can dynamically mitigate error in real time–allowing robot arms/systems to self-correct and move exactly from one position to another. Common feedback loops include PID (Proportional Integral Derivative), PI, PD, LQR (Linear Quadratic Regulator), and more. The most traditional approach is the PID, which constantly evaluates a robot's position and compares it to a target position to enhance the stability and speed of the robot, while also overcoming unforeseen obstacles like air resistance, friction, etc. Despite these systems' widespread use, however, there is much debate as to which control loop offers the best efficiency and accuracy, as well as other metrics.

Besides the standard feedback loops, there has also been research in hybrid and advanced approaches that transform traditional programs. For example, Rahmani and Rahman (2018) developed a novel fractional PID combined with a sliding-mode controller for a 7-degree-of-freedom exoskeleton robot. Their research suggested improved disturbance rejection as well as trajectory tracking, in comparison to other traditional PID approaches. This was determined through trials under uncertainty and injected noise. Their research also concluded that PID controllers are effective, but their structure and tuning significantly impact performance, and are a primary concern.

Similarly, Hamidi et al. (2023) implemented a PID controller on a self-balancing two-wheeled robot that was modeled like an inverted pendulum. Balancing systems, like inverted pendulums, are inherently unstable and nonlinear, which is why the authors emphasized that proper PID tuning is critical for maintaining stability as well as rapid disturbance correction. These findings further reinforce the importance of gain selection and dynamic response in feedback control systems, and prove that PID tuning is a major consideration that must be acknowledged during any experimentation.

Beyond traditional PID control, researchers have also explored hybrid and optimization-based methods to enhance performance. Xu, Wang, and Lu (2025) compared a traditional PID control system to a hybrid Linear Quadratic Regulator (LQR) and Proportional-Derivative (PD) controller for a wheeled bipedal robot. Their simulations and physical experiments suggested that hybrid PD/LQR controllers better accounted for center-of-mass changes (which were frequent and common among bipedal robots) and more complex dynamics than standard PID controllers. This suggested that LQR-based systems may provide significant advantages in handling multi-variable and dynamic robotic systems, and simultaneously introduce the idea of system combinations.

While previous research has demonstrated improvements through fractional PID, composite PID systems, and hybrid LQR approaches, there is a gap for a direct comparative testing of multiple feedback systems through identical physical conditions. Many studies rely heavily on simulations or focus on a single robotic architecture, so testing different controllers on a general 6-dof robotic arm using varying paths and evaluation metrics would provide clear insight into each system's relative strengths and weaknesses.

The purpose of this study is to evaluate and compare PID, PI, PD, LQR, and hybrid PD + LQR controllers using an assembled SO101 6-degree-of-freedom Arduino robotic arm. The study will measure both accuracy and efficiency through programmed, predefined three-dimensional paths/shapes. Accuracy will be evaluated using Mean Average Error (MAE), and efficiency will be assessed using battery consumption and motor output magnitude. Based on prior literature, it is hypothesized that PID-based controllers will produce more accurate trajectory tracking, and LQR-based or hybrid systems will demonstrate improved energy efficiency.

Methodology

Materials

- SO101 6-degree-of-freedom robotic arm (assembled from official kit documentation)
- Arduino microcontroller and motor driver modules
- Python software for implementing PID, PI, PD, LQR, and PD + LQR controllers
- MATLAB software for trajectory modeling and MAE calculations
- 6x servo motors with position feedback sensors for measuring x, y, z displacement
- Stable power supply with measurable battery percentage output

Procedure

1. Assemble the SO101 robotic arm according to the official kit documentation. Verify that all joints, motors, and electrical connections are functioning correctly.
2. Calibrate the robotic arm by setting it to a fixed home position. Zero all sensors and confirm that each joint operates within its intended range of motion.
3. Develop feedback control algorithms (PID, PI, PD, LQR, and PD + LQR hybrid) in Python. Include a protocol to measure the average. motor output during each trial. Test

each controller in MATLAB simulation to ensure proper trajectory tracking before implementation on hardware.

4. Upload each control algorithm to the Arduino microcontroller individually, ensuring that only one controller is active during each trial.
5. Select predefined three-dimensional paths for testing. These include straight-line trajectories and more complex paths such as curves or S-shaped patterns. Program each path as a sequence of target x, y, z coordinates over time.
6. Reset the robotic arm to the calibrated home position before beginning movement.
7. Execute the programmed path using one feedback controller at a time. Complete multiple trials for each controller-path combination to increase reliability and reduce random error.
8. During each trial, record real-time position data from sensors to capture the actual x, y, z coordinates of the robotic arm.
9. Calculate the tracking error at each time step using:
   Error = Target Position − Actual Position
10. Compute the Mean Average Error (MAE) across the entire trajectory to quantify overall accuracy for each controller.
11. Measure efficiency by recording motor output throughout each trial to determine average power usage. Average motor output is the mean of the total movement commands sent to all joints at each control step, averaged across the full duration of a trial.
12. Maintain consistent testing conditions across all trials, including surface stability, environmental factors, and power supply conditions.
13. Compile all data into comparative tables for each controller and path combination. Analyze differences in accuracy (MAE) and efficiency (motor magnitude).

14. Determine which feedback control system achieves the optimal balance between trajectory accuracy and energy efficiency.

Research group organized by table:

Controller/Path	PI	PD	PID	LQR	PD+LQR
Circle	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3
Square	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3
Heart	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3	Trial 1 Trial 2 Trial 3

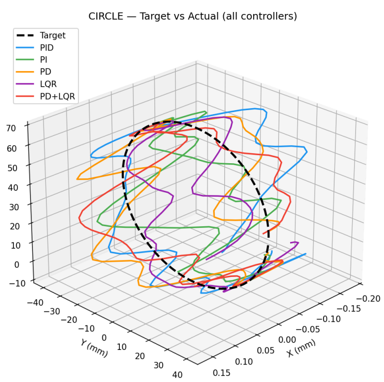
Extraneous Variables:

1. Keeping conditions constant: attempting to control friction throughout experiments is difficult, but ideally, the friction stays consistent throughout all trials.
2. Keeping the initial post consistent at the start of each trial (achievable through code).
3. Keeping the center-of-gravity constant (affix the robot to a weighted box to prevent tipping).

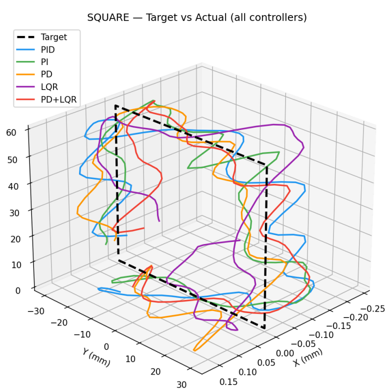
Data:

To get the broadest scope of the trials, the error results were plotted (in real time) in MATLAB to yield a visual representation of the paths/trials, so it is easy to understand what the error actually looks like. This solely serves as a visual aid, and unfortunately, it is difficult to discern the exact compiled error from the graphs below. In each plot, the dashed line represents the “target,” which is the shape (circle, square, heart) that the controller is following. The feedback loop aims to be as close to the target line as possible, but of course, there will be error–in this case, measured by millimeters.

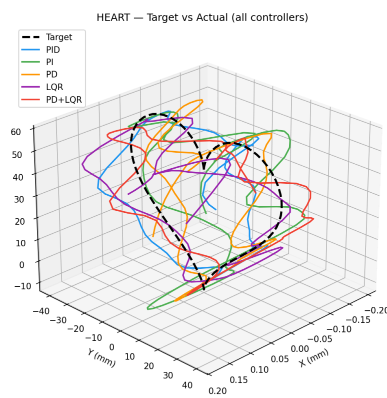
Circle graph:



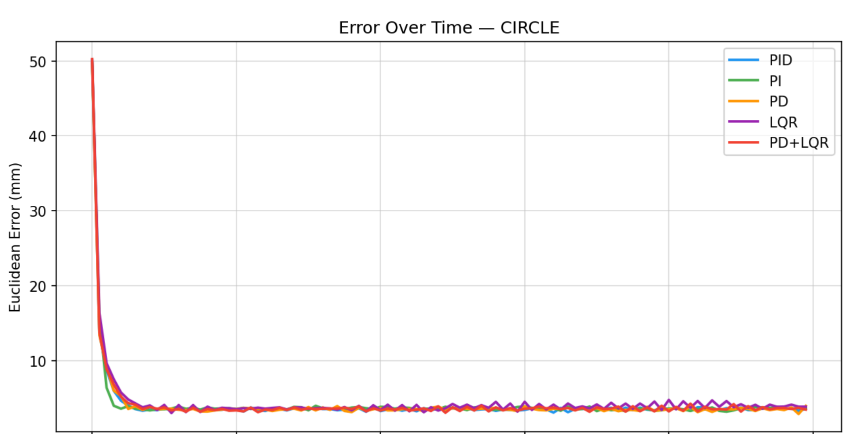
Square graph:

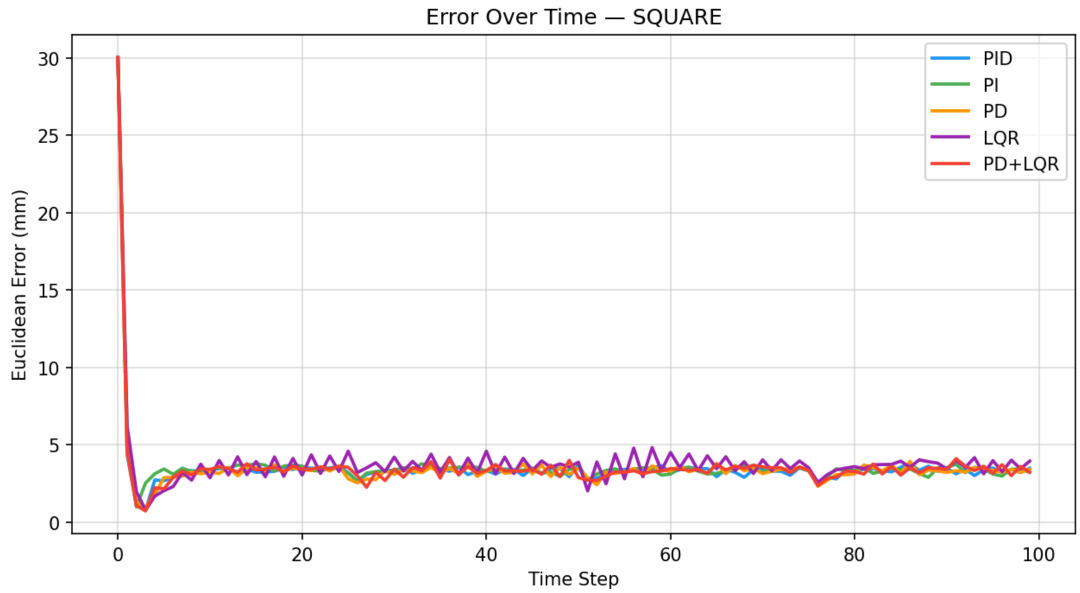


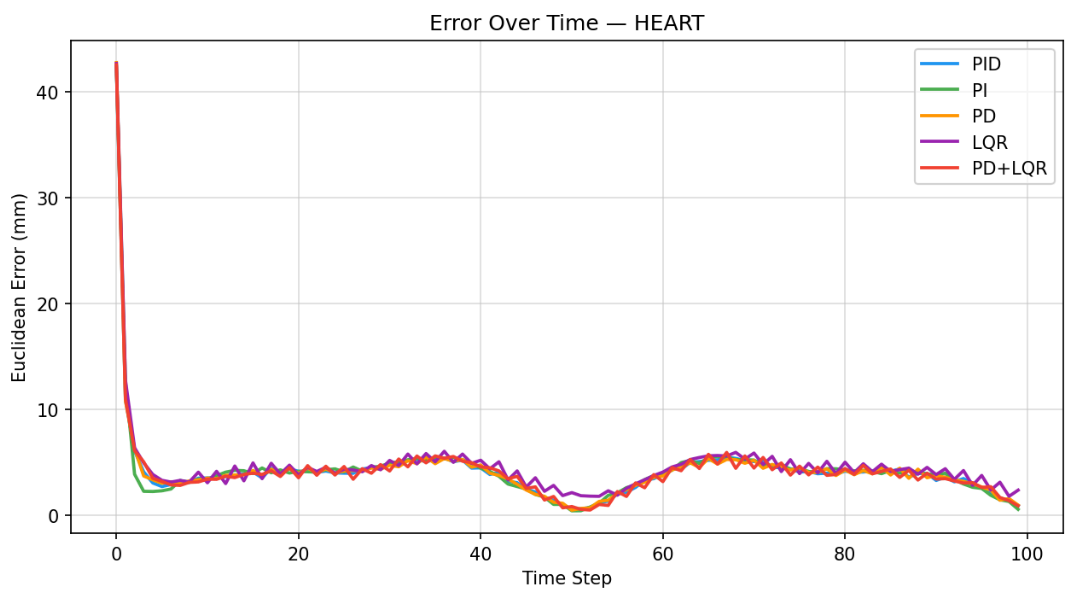
Heart graph:



Here, it is evident that the heart shape yields more error than the square and circle, with the square visibly causing the least error. Although the error looks large, it is actually quite small, relative to the units the graph is measured in (mm). The next graphs plotted were compilations of the ones above, which instead measure the error over time. The Euclidean error is the exact mathematical explanation of the straight-line distance between a predicted point and the true value, calculated as the root of the sum of squared differences. While the change in error was not a specific aspect of the procedure/experiment, it is easier to see the total accumulated error across all 5 controllers, as opposed to just the images of the paths.



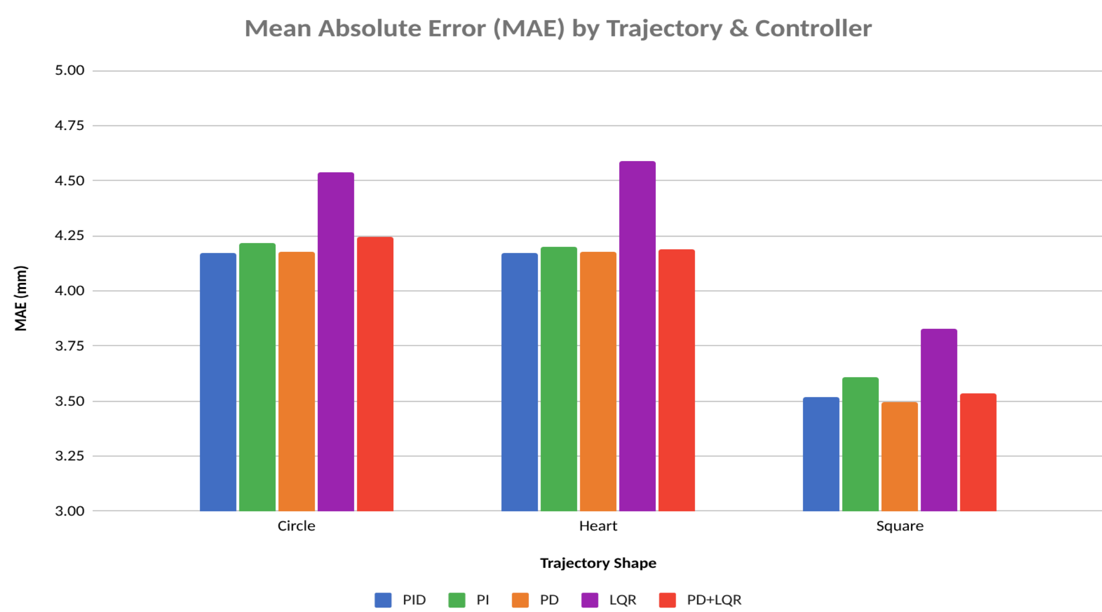




It is clear that the results in these graphs closely align with the prior figures that depict the error across the shape. Again, this is because the ‘Error over time’ graph is a direct accumulation of the real-time error plotted in the ‘target vs actual’ graphs. Note the scale on the y-axis–error under 5mm in the square path reinforces the claim that this is the most accurate path to follow. Variability and high error continue to suggest the difficulty of tracking heart shapes, and occasionally circle shapes as well. Ultimately, these figures were primarily visuals of the experiments, and not extremely important to any final results, because the aim of the project is directed towards efficiency and accuracy for the controller, not the shape. Still, within each figure, the disparity between each controller is evident, but the margin is so small that it is difficult to get a quantitative estimate of this difference that the experiments seek to explore.

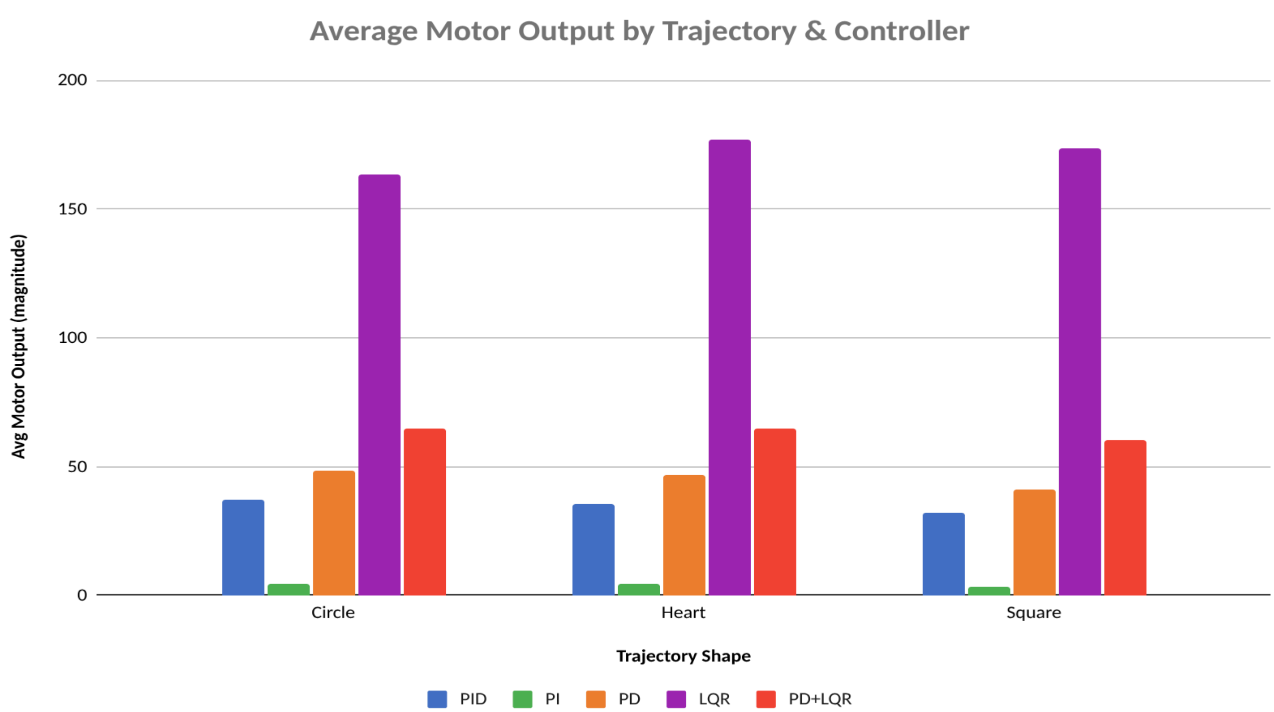
Analysis & Conclusion

For each controller, I organized the mean-absolute-error (accumulation of mean-average-error) and average-motor-output results by shape. This way, I can compare each controller against the other controllers, without confounding variables like the shape of the path. Though not crucial to the experiment, another reason I grouped the shapes was to understand (briefly) how the shape may affect both the error and power usage. The MAE (mean-absolute-error) varies from 3mm to 5mm on the y-axis, and each controller is differentiated by color, as indicated by the key.



Through each shape, it is evident that the LQR has the highest error among all controllers. As suggested by prior graphs, the circle and heart have similar amounts of error, with the LQR resulting in over 4.5mm of error in either shape. In the square, the LQR controller nets 3.75+mm of error, which far exceeds the other controllers. The best performing controller is the PD, which has the least error in the square, and T1 (tied in first place) in the circle and heart. The PID is an extremely close second, with a <0.05mm difference from the PD in every shape. Consistent with hypotheses, the LQR+PD hybrid approach outperforms traditional LQRs (with quite a large margin), though it is unable to exceed the PIDs. The PD is statistically the most accurate, though, with a narrow statistical significance of $p = 0.046 < (\alpha = 0.05)$, performed through a difference in means t-test, $t = -2.16$.

Next, the average motor output was examined; the graph is organized identically to the error graph, with magnitude varying from 50-200 units as indicated by the y-axis.



Again, results stay consistent throughout every shape. However, contrary to the hypothesis, the LQR controller is the least motor efficient, exceeding 150 units in every shape. Conversely, the PI is by far the most motor efficient, with <5 units of magnitude in every shape, which is 43x less than the LQR. The PID is second, but by quite a bit margin–still, it is very close to the output magnitude of PDs, just like accuracy metrics. Again, the hybrid PD+LQR outperforms the LQR in efficiency, but is still underwhelming compared to PIDs, which have significantly lower magnitude. These results are much more statistically significant, with a 2 mean t-test yielding a p value of 0.0001–much less than $\alpha = 0.05$.

Ultimately, the hypothesis, "PID-based controllers will produce more accurate trajectory tracking, and LQR-based or hybrid systems will demonstrate improved energy efficiency," is partially correct. PID-based is broad and refers to the PI, PD, and PID, so therefore the hypothesis is true in predicting the PD to be the most accurate. Yet, the LQR was actually the least energy efficient, and the PID-based systems were actually significantly more efficient than the linear counterpart. Hybrid approaches did demonstrate improved energy efficiency, though, seeing that the LQR+PD was much more efficient than the LQR alone. Feedback loops are found everywhere in robotics, and really, all moving systems. Having a comprehensive understanding of the differences in performance between controllers is vital for discerning the best loop for certain scenarios, and in general. With these findings, researchers can implement PIDs/PDs when prioritizing accuracy and PIs for maximizing efficiency.

Limitations & Considerations

Still, some limitations and confounding factors may have influenced the results. One variable that was acknowledged previously (but not addressed) was the potential incidence of friction and other physical hardware constraints that varied by trial. These variables would likely add noise that was not initially captured in the simulations, and also skew results by a large factor, considering that the results only varied by millimeters. Additionally, a variable that was unforeseen and may have affected the trials was the servo temperature drift that could have affected later trials. As the trials were run sequentially, the motors could have accumulated heat, and this would lead to servo drift that occurs after many trials have run. This could have been neutralized by either random selection of trials, and/or large breaks between trials.

Extentions

Because feedback loops are such a big point of research within robotics, there are many extension points and opportunities for future research. One such point is: what other hybrid systems can be evaluated? PID-based systems and LQRs are only 2 of the many feedback loops that can be evaluated in terms of accuracy and efficiency. Additionally, only the PD+LQR hybrid approach was explored, but there are many combinations of other loops that can be examined, and it would be interesting to see how they compare against the single approaches and the other hybrid approaches. It would also be interesting to test controllers on a different robot platform (e.g., wheeled or bipedal) to see if results generalize.

Acknowledgements

Tillman Dillon - PhD student-mentor: Helped understand all concepts and selected the so101 repository.

Mrs. Harbin - STEM Research Class Teacher: Provided constant support and guidance from proposal to final presentation.

Mrs. Campos & Mrs. Remus: Enabled access to robot servos and all corresponding parts; robot assembly would not be possible without their guidance/support.

IRB Board: Passed the research proposal and offered feedback for the final project.

Bibliography:

Cheong, J., & Lee, S. (2008). Linear PID composite controller and its tuning for flexible link robots. *Journal of Vibration & Control, 14*(3), 291–318. https://doi.org/10.1177/1077546307073130

Feng, H., Yin, C.-B., Weng, W., Ma, W., Zhou, J., Jia, W., & Zhang, Z. (2018). Robotic excavator trajectory control using an improved GA based PID controller. *Mechanical Systems & Signal Processing, 105*, 153–168. https://doi.org/10.1016/j.ymssp.2017.12.014

Hamidi, E. A. Z., Effendi, M. R., Basuni, A., & Wildan, M. (2023). Implementation of a self-balancing robot based on the proportional integral differential (PID) controller parameter. *AIP Conference Proceedings, 2646*(1), 1–9. https://doi.org/10.1063/5.0115865

Priambodo, A. S., Dhewa, O. A., Nasuha, A., Arifin, F., Winursito, A., & Muslikhin. (2025). Marker-based autonomous quadrotor tracking for ground mobile robots. *AIP Conference Proceedings, 3281*(1), 1–11. https://doi.org/10.1063/5.0261170

Rahmani, M., & Rahman, M. H. (2018). Novel robust control of a 7-DOF exoskeleton robot. *PLoS ONE, 13*(9), 1–18. https://doi.org/10.1371/journal.pone.0203440

Urrea, C., & Saa, D. (2023). Design, simulation, implementation, and comparison of advanced control strategies applied to a 6-DoF planar robot. *Symmetry, 15*(5), 1070. https://doi.org/10.3390/sym15051070

Xu, Y., Wang, Z., & Lu, C. (2025). Design and control of a wheeled bipedal robot based on hybrid linear quadratic regulator and proportional-derivative control. *Sensors, 25*(17), 5398. https://doi.org/10.3390/s25175398