Introduction to Feedback Control Systems

LAB EXPERIMENT 1

Harel Ko
BS Electronics Engineering, Samar State University (of Aff.)
Institute of Electronics Engineers of the Philippines (of Aff.)
Catbalogan City, Philippines
Harelko999@gmail.com

Luis Nacional
BS Electronics Engineering, Samar State University (of Aff.)
Institute of Electronics Engineers of the Philippines (of Aff.)
Catbalogan City, Philippines
Obeywes7@gmail.com

***Abstract*—This experiment investigates the behavior of a DC motor system under open-loop and closed-loop control configurations using an Arduino-based setup with a potentiometer for position sensing and an L293D H-bridge motor driver. In the open-loop test, the motor operated at a fixed PWM input, resulting in observable RPM fluctuations and no active correction of system deviations. In contrast, the closed-loop test employed a PID controller to dynamically adjust motor behavior based on feedback from the potentiometer. Results showed significant improvement in system responsiveness and reduction in steady-state error with feedback implementation, despite residual fluctuations and simulation limitations. The experiment highlights the fundamental importance of feedback control in achieving greater stability and accuracy in dynamic systems.**

I. RATIONALE

Feedback control systems are fundamental for regulating dynamic systems by adjusting the input based on the output. Understanding the role of feedback is crucial to designing stable and efficient systems in various engineering fields.

II. OBJECTIVES

- Quantify the reduction in steady-state error when transitioning from an open-loop to a closed-loop system. Materials and Software
- Measure the improvement in system stability by comparing the time-domain responses (e.g., rise time, settling time) of the open-loop and closed-loop systems.
- Determine the PID controller gains that minimize steady-state error while maintaining a response that is within 5% of the desired value.

III. MATERIALS AND SOFTWARE

- Materials: Arduino UNO, Jumper Wires, 9V Battery, Potentiometer (for position sensing), H-bridge motor driver (L293D), DC motor(position sensor).
- Software:TinkerCAD, Google Colab(with control, matplotlib, numpy)

IV. PROCEDURES

1) Set up the system: Connect the DC motor to the motor driver/controller. Use a potentiometer to measure the motor's position and connect it to the input of a PID controller (simulated or hardware-based).
2) Initial Test: Run the motor in an open-loop configuration (without feedback) and observe the response on the oscilloscope. Record the steady-state error.
3) Implement Feedback: Introduce feedback into the system. The position of the motor should now be fed back into the controller to adjust the input continuously.
4) Tune the Controller: Adjust the PID controller parameters to achieve the desired per formance (e.g., minimal steady-state error, fast response time, etc.).
5) Observation: Compare the system behavior with and without feedback. Analyze the impact of feedback on accuracy and stability.
6) Analysis: Use Python or Scilab to simulate the open-loop and closed-loop transfer functions, compare the theoretical and experimental results, and plot the response curves.

V. OBSERVATION AND DATA COLLECTION

OBSERVATION:

Initial Test – Open-Loop (Without Feedback)

1) In the initial open-loop test, a simple code structure was used to apply a fixed PWM value (0–255 range). The PWM was set to 1 to intentionally keep the DC motor speed low and within a manageable range.
2) As expected for an open-loop system, the potentiometer had no influence on the motor's behavior. However, potentiometer readings were still monitored through the Serial Monitor, which confirmed that changes in its value were detected by the Arduino.
3) Upon starting the simulation, the DC motor's speed was observed to fluctuate between approximately 59 and 71 RPM. These fluctuations are likely due to simulation artifacts in TinkerCAD or minor variations in how the motor model processes the constant PWM input.
4) Since there was no designated set-point or feedback mechanism, steady-state error could not be meaningfully defined in this open-loop configuration.

Implement Feedback - Closed loop (PID Control)

1) After implementing feedback using a PID control structure, the potentiometer input was integrated into the system, allowing it to influence the motor behavior dynamically.

2) Upon starting the simulation, it was immediately apparent that the system experienced noticeable lag and occasional freezing, particularly when abrupt or large changes to the potentiometer were made during operation. This suggests that TinkerCAD's simulation environment may have limitations in handling real-time dynamic interactions among multiple active components.
3) However, changes in system behavior were clearly observable: The speed and direction of the DC motor responded to the calculated error between the potentiometer reading and the set point. As the potentiometer approached the setpoint, the motor slowed and fluctuated within a smaller error margin.
4) Tuning the PID gains (proportional, integral and derivative) helped to reduce the amplitude of fluctuations. However, it was not possible to eliminate them completely, which is consistent with real-world behavior, where mechanical, electrical, and sensor imperfections introduce unavoidable disturbances and noise.
5) It was also noted that when the steady-state error became significantly high, the system tended to overcompensate, leading to increased lag or freezing during simulation. This behavior highlights the challenges of achieving both stability and responsiveness in practical control systems.

DATA COLLECTION:

https://drive.google.com/drive/folders/1nlBYb46zLNtSZMrwBAwLMYGsc64G5E30?usp=sharing

Click here to open the Drive

VI. DATA ANALYSIS

In the open-loop test, the DC motor was driven by a fixed PWM signal (PWM = 1), and its speed was observed to fluctuate between 59 RPM and 71 RPM. Since no feedback was present, the potentiometer reading had no influence on motor behavior, although its values were successfully monitored in the Serial Monitor. The fluctuation in motor speed was attributed to inherent system noise and simulation limitations within TinkerCAD, rather than any control action.

In the closed-loop test with PID control, the motor was actively driven based on the difference between the desired setpoint (potentiometer reading) and the motor's current behavior. When the potentiometer was adjusted, the system attempted to correct the error by changing motor speed and direction. Steady-state errors were reduced, but not eliminated, due to both imperfect tuning and simulation artifacts. Adjustments to the PID gains significantly influenced system behavior, affecting response speed, stability, and the magnitude of oscillations.

However, limitations in the TinkerCAD environment, such as processing delays and simulation lag, introduced artificial "freezing" and overshooting effects, especially during abrupt changes. These effects were not purely characteristic of the control system but also of the simulation platform's performance under load.

VII. DISCUSSION AND INTERPRETATIONS

The open-loop configuration highlighted a key limitation of systems without feedback: no self-correction mechanism exists to adjust for disturbances or deviations. Despite being fed a constant PWM signal, the motor's RPM still fluctuated, demonstrating that even under theoretically stable inputs, real (or simulated) physical systems exhibit variability.

Upon implementing PID feedback control, the motor system gained the ability to react to real-time changes in the input (potentiometer), dynamically adjusting its behavior to minimize error. This demonstrated the central advantage of feedback systems: improved accuracy and reduced steady-state error compared to open-loop systems.

PID tuning played a crucial role in system performance:

1) Increasing the proportional gain (Kp) made the system react faster but introduced larger oscillations.
2) Adding integral gain (Ki) helped to eliminate residual steady-state error but made the system more prone to overshoot and instability.
3) Derivative gain (Kd) improved damping and helped reduce oscillations caused by sudden changes.

Despite tuning efforts, perfect stability was not achieved. Some residual fluctuations persisted, which reflects real-world behavior where no physical system is perfectly noise-free or instantaneous. Additionally, the simulation environment introduced artificial lag and freezing when overwhelmed by large or sudden input changes, affecting perceived system performance.

The general system behavior matched theoretical expectations:

1) Open-loop systems could not correct themselves or respond to disturbances.
2) Closed-loop PID systems reduced steady-state errors and allowed dynamic tracking of a desired position (potentiometer value), albeit with some inevitable imperfections.

VIII. CONCLUSION

The experiment successfully demonstrated the fundamental differences between open-loop and closed-loop control systems. In the open-loop configuration, no feedback correction occurred, resulting in observable motor speed fluctuations without any mechanism for correction. Implementing a PID-based feedback system significantly improved the system's ability to dynamically adjust motor speed and direction in response to changes in the potentiometer, effectively reducing steady-state error.

Tuning the PID controller highlighted the trade-offs between responsiveness, stability, and precision. Although fluctuations could not be completely eliminated—due to both physical system limitations and simulation artifacts—the experiment effectively illustrated the critical role of feedback control in minimizing error and improving system stability.

Overall, the results aligned well with theoretical predictions, reinforcing the importance of feedback systems in engineering applications where precision and adaptability are essential.