

Magnetic Levitation System

Jordan Williams | Emerson Hall | Nevan Mukherjee

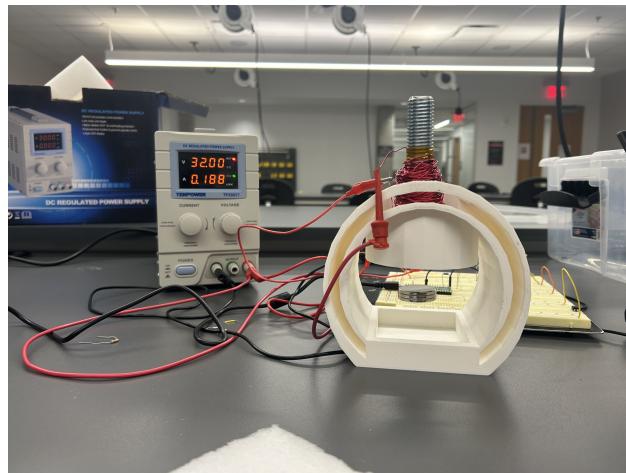


Figure 1: Magnetic Levitator

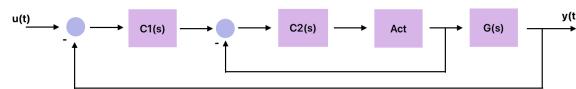


Figure 2: Closed-Loop Block Diagram

Mechanical Dynamics

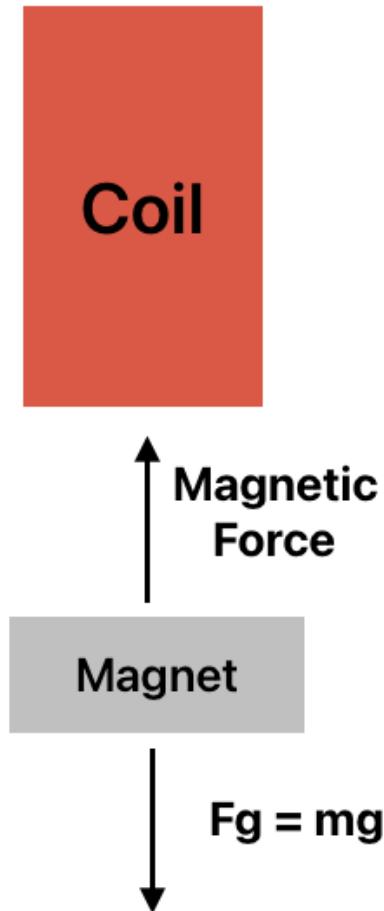


Figure 3: Free Body Diagram of Magnetic Levitation System

This report presents the modeling, control, and implementation of a magnetic levitation system using a cascaded feedback structure. By combining a linearized dynamic model with real-time feedback loops implemented on a Raspberry Pi Pico, the system demonstrates stable levitation under controlled conditions. We analyze the plant dynamics, controller design, sensor integration, and actuator behavior, comparing theoretical models to experimental results.

The magnetic levitation system suspends a small ferromagnetic object (magnet) in mid-air by controlling the current through an electromagnet. The upward magnetic force is regulated in real-time to oppose the downward gravitational force, achieving stable levitation.

Gravitational Force:

- $F_g = mg$
- Mass: $m = 0.029 \times 10^{-3} \text{ kg}$
- Gravity: $g = 9.8 \text{ m/s}^2$

Magnetic Force:

$$F_c = \frac{Ki^2}{h^2}, \quad \text{where } K = \frac{\mu_0 AN^2}{2}$$

Constants:

- $\mu_0 = 4\pi \times 10^{-7} \text{ H/m}$
- $A = 10^2 \times 10^{-6} \text{ m}^2$ (cross-sectional area)
- $N = 429$ (number of turns)
- $K = 0.036$
- Desired levitation height: $h_0 = 3 \times 10^{-3} \text{ m}$

System Dynamics

At equilibrium:

$$F_N = F_c - F_g = 0 \Rightarrow \frac{Ki^2}{h^2} = mg \Rightarrow K \frac{i^2}{h^2} = mg$$

This can be written dynamically as:

$$m\ddot{h} = \frac{Ki^2}{h^2} - mg$$

The magnetic force equation is linearized using a first-order Taylor expansion about the equilibrium point h_0, i_0 :

$$f(h, i) \approx f(h_0, i_0) + \frac{\partial f}{\partial h}(h - h_0) + \frac{\partial f}{\partial i}(i - i_0)$$

Start from:

$$F_c = \frac{Ki^2}{h^2}$$

Partial derivatives:

$$\frac{\partial F_c}{\partial h} = -\frac{2Ki^2}{h^3}, \quad \frac{\partial F_c}{\partial i} = \frac{2Ki}{h^2}$$

Substitute into net force expression:

$$m\ddot{h} = \frac{Ki^2}{h^2} + \frac{2Ki}{h^2}(i - i_0) - \frac{2Ki^2}{h^3}(h - h_0) - mg$$

Canceling equilibrium terms:

$$m\ddot{h} = 2K \left(\frac{i}{h^2}(i - i_0) - \frac{i^2}{h^3}(h - h_0) \right)$$

Let:

$$K_1 = 2K \frac{i_0}{h_0^2}, \quad K_2 = 2K \frac{i_0^2}{h_0^3}$$

Then:

$$m\ddot{h} = K_1(i - i_0) - K_2(h - h_0)$$

Define deviations:

$$\tilde{h} = h - h_0, \quad \tilde{i} = i - i_0$$

Differential Equation:

$$m\ddot{\tilde{h}} = K_1\tilde{i} - K_2\tilde{h}$$

Transfer Function Derivation

Taking the Laplace transform (zero initial conditions):

$$ms^2H(s) + K_2H(s) = K_1I(s) \Rightarrow \frac{H(s)}{I(s)} = \frac{K_1}{ms^2 + K_2}$$

Plug in values:

$$G(s) = \frac{-21280}{0.029 \times 10^3 s^2 - 1886826.67}$$

Open-Loop Characterization

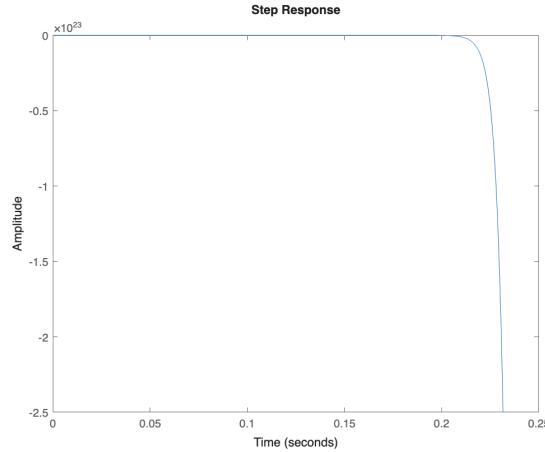


Figure 4: Step Response of Plot

The step response of the open-loop system reveals an extremely large and negative amplitude, on the order of 10^{23} , with a rapid falloff occurring around 0.22 seconds. This behavior indicates a highly unstable system. Such an exaggerated response to a unit step input is consistent with theoretical expectations for the magnetic levitation system, which is known to be open-loop unstable. The magnitude and direction of this response imply that even minimal input disturbances can drive the system into divergence, emphasizing the need for a well-designed stabilizing feedback controller.

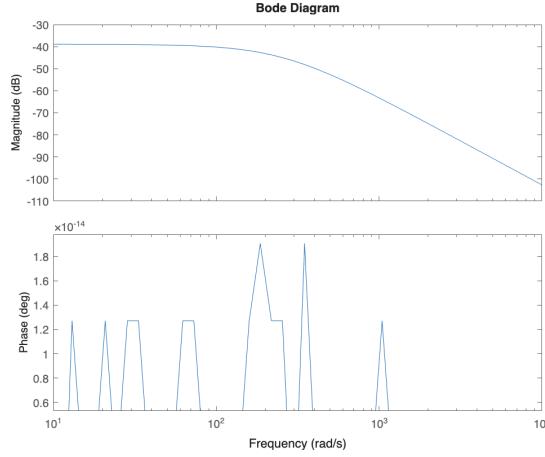


Figure 5: Bode Plot of Plant

The Bode plot of the open-loop plant provides insight into its frequency response characteristics. The magnitude plot starts at approximately -40 dB and continues to drop steadily, indicating strong attenuation at higher frequencies typical of a second-order system. However, the phase plot exhibits nearly zero phase shift across the frequency spectrum, with

fluctuations on the order of 10^{-14} degrees. This unusual result is likely due to numerical artifacts or limitations in the linearized model, especially given the open-loop instability of the system. As such, the phase information in this plot should be interpreted with caution, as it does not meaningfully reflect the actual system dynamics.

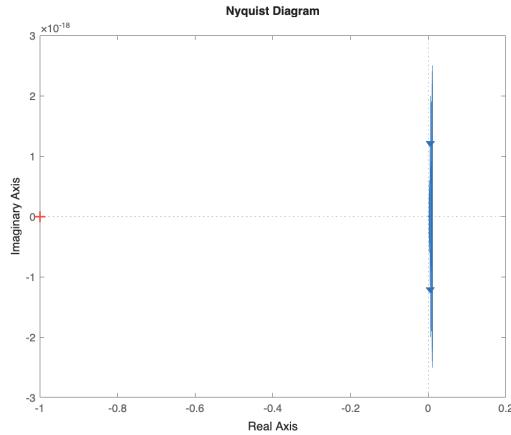


Figure 6: Nyquist Plot of Plant

The Nyquist plot of the open-loop system further confirms the presence of open-loop instability. The trajectory appears nearly vertical, centered close to the imaginary axis, with real parts clustered near zero and imaginary components on the order of 10^{-18} . The plot does not encircle the critical $(-1, 0)$ point, but due to the presence of right-half plane poles in the open-loop transfer function, the Nyquist stability criterion indicates that the system is unstable. This visualization reinforces the conclusion that feedback is essential to achieve stable levitation.

Closed-Loop Improvements

To improve the closed-loop stability and responsiveness of the system, our team iteratively tuned PID controllers based on the system's open-loop dynamics. Initially, a simple proportional-derivative (PD) controller stabilized the levitation for three magnets. After a hardware failure reduced us to two magnets, gains were recalculated to maintain performance.

To further enhance stability, especially as wire resistance dropped with temperature, we added a discrete inner-loop controller (lead compensator) to directly regulate solenoid current. This reduced overshoot and helped stabilize the levitation height even under fluctuating thermal conditions.

2. Sensors

To measure the position of the levitated magnet, we used a Hall effect sensor that detects the magnetic field strength emitted by the magnet. The sensor outputs an analog voltage corresponding to the proximity of the magnet, which is interpreted by the controller as a proxy for vertical position.



Figure 7: Hall Effect Sensor

The sensor was placed directly beneath the solenoid to capture vertical changes in the magnetic field. The output voltage was read by the Raspberry Pi Pico's ADC and scaled accordingly. We chose a target operating voltage of 2.05 V to represent the desired levitation height. To convert raw ADC readings into usable voltage values, we used the formula:

$$V = \frac{\text{ADC}}{65535} \times 3.3$$

Filtering and control decisions were applied in Python to ensure stable tracking of the reference voltage corresponding to the set levitation height.

3. Initial Controller Design

To stabilize the inherently unstable magnetic levitation system, we designed a multi-layered control architecture composed of an outer-loop PID controller and an inner-loop current regulation compensator. These controllers were developed iteratively, in response to changes

in system behavior and hardware limitations.

The original PID controller was tuned to levitate three identical magnets simultaneously. The chosen gains were:

- $K_p = 1.45 \times 10^6$
- $K_d = 1.14 \times 10^5$
- $K_i = 0$

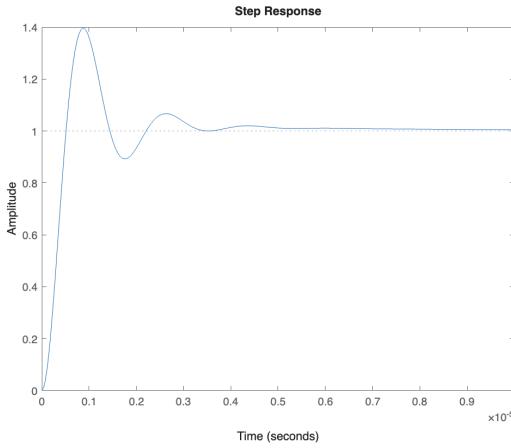


Figure 8: Step Response of System with PD Controller

This configuration represents a PD controller (with $K_i = 0$), which provided sufficient responsiveness to correct for position deviations without introducing integral wind-up. The proportional term scaled the error between the target Hall sensor voltage and the measured value to drive the output, while the derivative term introduced damping to suppress oscillations and rapid transients. This controller worked well in the initial setup, achieving stable levitation with minimal overshoot and good tracking performance.

To address the additional challenge of fluctuating current due to coil resistance changes (caused by thermal effects), we introduced an inner-loop current controller implemented as a discrete-time lead compensator. This regulator directly controlled the current through the solenoid by adjusting the PWM signal driving the MOSFET, based on the difference between the desired and measured current.

The compensator was implemented using the following difference equation:

$$u[n] = b_0 e[n] + b_1 e[n-1] - a_1 u[n-1]$$

where:

- $b_0 = 2.865$
- $b_1 = 1.915$
- $a_1 = 0.752$
- $e[n]$ is the current error at time n
- $u[n]$ is the control output (PWM duty cycle)

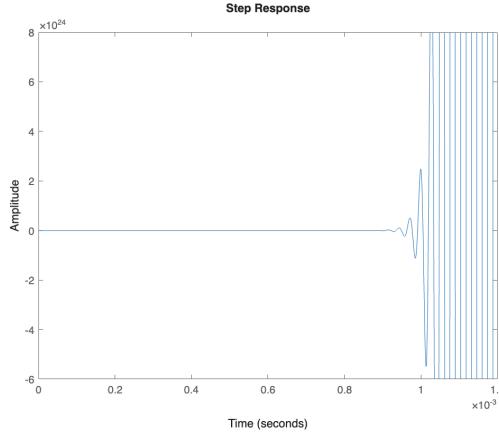


Figure 9: Step Response of System with 2 Loops

This discrete lead compensator introduces phase lead to improve the inner loop's responsiveness and compensate for lag in the current dynamics. By placing this controller in the current path, we were able to significantly reduce overshoot and improve stability when the coil's resistance changed due to heating. The outer PID controller continued to manage position control, but its output was now interpreted as a desired current value, which the inner controller tracked accurately.

Together, the outer-loop PD and the inner-loop lead compensator created a cascaded control structure that allowed us to stabilize and precisely regulate the levitation height in real time under varying thermal and electromagnetic conditions.

4. Actuators

Figure 10: Solenoid with 429 Coils

The actuator in our system is a custom-wound electromagnet consisting of 429 turns of 22 AWG copper wire around a steel bolt. When current flows through the coil, a magnetic field is generated which pulls the nearby ferromagnetic object (the magnet) upward.

The amount of force applied to the magnet is directly proportional to the square of the current through the coil and inversely proportional to the square of the vertical distance between the coil and the magnet. By modulating this current via PWM from the Raspberry Pi Pico and a MOSFET, the magnetic force is finely controlled.

We used a logic-level MOSFET (IRLZ44N) to switch the 9V supply through the coil. The PWM signal was generated by the Pico and adjusted in real-time using the outer-loop PID and inner-loop lead compensator. This ensured accurate and responsive adjustments in magnetic force to maintain stable levitation under different operating conditions.

5. Controller Design

Our controller design was based on augmenting the nonlinear open-loop plant with precise feedback mechanisms, enabling robust levitation. Physical constraints limited the system to a maximum of 0.3 A through the electromagnet to prevent thermal damage, and the Hall effect sensor provided position feedback constrained within the 0–3.3 V ADC range.

The plant was linearized about an operating point (i_0, h_0) , resulting in a second-order, open-loop unstable system. Our goal was to stabilize this plant and reject disturbances by shaping the closed-loop response.

We initially implemented a proportional-derivative (PD) controller. When one magnet broke, the reduced magnetic mass required retuning of the outer-loop controller gains to restore system responsiveness. However, we observed current overshoot and degraded stability, prompting the implementation of an inner-loop discrete-time lead compensator to directly regulate the solenoid current.

While MATLAB simulations of the linearized system confirmed open-loop instability, they also highlighted challenges in phase margin and gain crossover, even under cascaded control. In contrast, our real-world system achieved temporary levitation, though it degraded due to non-idealities like thermal drift and coil resistance changes — factors not fully captured in simulation.

In contrast, our real-life system initially achieved levitation. However, the controller exhibited growing oscillations over time and eventually led to instability, especially as the coil heated and resistance changed. These differences between simulated and real responses suggest that non-idealities, such as coil inductance saturation and supply ripple, affected real-world behavior in ways not captured by the Matlab model.

To stabilize the real system, we began each run by initializing the magnet near its desired hover height ($h_0 = 3.6$ mm, or 2.05 V on the Hall sensor). The controller engaged af-

ter startup transients subsided to avoid the magnet slamming into the coil or falling due to underactuation. This approach provided temporary stability, sufficient for testing and capturing experimental results.

6. Controller Implementation

The controller was deployed on a Raspberry Pi Pico using MicroPython. The outer-loop PID controller regulated the magnet's vertical position via feedback from the Hall effect sensor, while the inner-loop discrete lead compensator stabilized the solenoid current. Both loops operated at a frequency of 1 kHz to support real-time control.

The Hall sensor was connected to ADC GPIO 26, and the solenoid current was measured across a shunt resistor using a differential amplifier routed to ADC GPIO 27. PWM signals generated on GPIO 16 controlled a MOSFET at 10 kHz for solenoid actuation.

During each control loop iteration, ADC values were read and converted to voltages. The outer-loop PID controller computed the position error and generated a target current. The inner loop then compared the measured current with the target and applied a lead compensator to determine the PWM duty cycle, which was applied to the coil.

```

error = target_voltage - voltage_from_hall
p_term = Kp * error
...
# Discrete lead compensator (current loop)
e = target_current - measured_current
u = b0 * e + b1 * e_prev - a1 * u_prev
u_prev = u
...
# Apply to PWM
pwm.duty_u16(clamp(int(u), 0, 65535))

```

In practice, the controller successfully levitated the magnet, but exhibited marginal stability. As the coil heated, the system became increasingly sensitive and would eventually enter an unstable oscillatory state. While initial performance matched theoretical expectations in terms of transient response, the observed degradation confirmed the importance of real-world testing beyond simulation.

Overall, the results demonstrate that the cascaded control design is effective for temporary stabilization of a nonlinear magnetic levitation system, but requires additional robustness improvements to maintain long-term stability in physical hardware. Another issue at hand

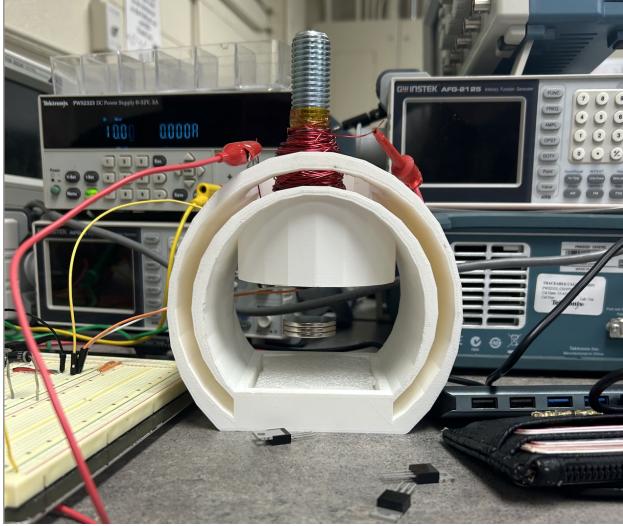


Figure 11: Magnet Levitating

was not all the real-life aspects were applied to the model.

7. Conclusion

This project demonstrated the design, simulation, and implementation of a two-loop feedback controller for a magnetic levitation system using a Raspberry Pi Pico. The system, inherently unstable in open-loop, was stabilized through a cascaded control architecture. The outer loop used a proportional-derivative (PD) controller to regulate position based on Hall effect sensor feedback, while the inner loop employed a discrete-time lead compensator to control current through the solenoid.

Through theoretical modeling and linearization of the nonlinear plant, we identified key challenges associated with phase margin and disturbance rejection. Although simulations in Matlab indicated persistent instability, the real system achieved temporary stable levitation under controlled conditions. Our implementation revealed that while the controller could initially stabilize the magnet, the system became increasingly sensitive to disturbances and thermal drift over time.

Despite these limitations, the project achieved its core objective: demonstrating closed-loop magnetic levitation using accessible hardware and custom control design. The experimental results underscored the importance of combining theory with practical validation, particularly when dealing with nonlinear, thermally-sensitive systems.

Future improvements should include modeling additional real-world factors such as MOSFET resistance, nonlinear magnetic force behavior, and thermal characteristics to enhance long-

term robustness.