

Magnetic Levitation Using Arduino: An In-Depth Technical Review

September 16, 2024

1 Introduction

Magnetic levitation technology enables objects to float by balancing magnetic forces and gravitational pull. The project focuses on constructing a magnetic levitation system using an Arduino, emphasizing control through sensors and precise electromagnetic fields. This system successfully levitated a magnet vertically using a solenoid coil, Hall-effect sensors, and a PID (Proportional-Integral-Derivative) controller.

2 System Overview

The primary objective was to develop a closed-loop control system that levitates a magnet by manipulating the electromagnetic force generated by a solenoid coil. The system consists of:

- **Solenoid Coil:** Generates the magnetic field to attract the magnet.
- **Hall-Effect Sensors:** Measure the magnetic field strength to provide feedback on the magnet's position.
- **Arduino Microcontroller:** Processes the sensor data and adjusts the coil's current accordingly.
- **PID Control Algorithm:** Ensures stable levitation by dynamically adjusting the control signal.

3 General System Analysis

The proposed magnetic levitation system is composed of an electromagnet placed above the levitating object. The electromagnet generates a magnetic field that is controlled by the current flowing from the Arduino. The magnetic field produces a vertical magnetic force on the object that acts in the opposite

direction to its weight. The goal is to balance these forces so that the object remains suspended in the air.

A Hall-effect sensor detects the position of the levitating object by measuring the magnetic field generated by both the magnet and the coil. The sensor outputs an electrical signal corresponding to the object's position. This signal is fed into the controller, which then adjusts the magnetic field strength in real-time to ensure the object remains stable at the desired reference position. The feedback control loop continuously adjusts the current to maintain the object's position.

A control system block diagram is used to represent how the components interact with each other:

- The electromagnet generates the magnetic field.
- The Hall-effect sensor detects the position of the object by measuring the magnetic field.
- The sensor's signal is sent to the Arduino (controller).
- The Arduino adjusts the current in the coil based on the position feedback, controlling the magnetic field to maintain stable levitation.

The system dynamically balances the forces acting on the object, which requires precise real-time control. The PID controller plays a crucial role in minimizing the error between the desired and actual positions of the levitating object. The control loop ensures that any disturbance in the object's position is corrected quickly by adjusting the magnetic field generated by the electromagnet.

4 Physics and Electromagnetism

The foundation of this project relies on **electromagnetic theory**, particularly the interaction between a solenoid's magnetic field and a ferromagnetic object (the magnet).

4.1 Magnetic Field of a Solenoid

The solenoid generates a magnetic field when current flows through its windings. The magnetic field (B) inside a solenoid is given by:

$$B = \mu_0 \cdot \frac{N \cdot I}{L} \quad (1)$$

Where:

- B is the magnetic flux density (measured in Tesla),
- μ_0 is the permeability of free space ($4\pi \times 10^{-7} \text{ T} \cdot \text{m/A}$),

- N is the number of turns of the coil,
- I is the current through the coil (in amperes),
- L is the length of the solenoid.

The **force** exerted on the magnet is directly related to this magnetic field and opposes gravity. This balancing act ensures levitation:

$$F_m = B \cdot \mu_m \quad (2)$$

Where μ_m is the magnetic moment of the object.

4.2 Levitation Equilibrium

The system must balance the upward magnetic force (F_m) with the downward gravitational force (F_g):

$$F_m = F_g = m \cdot g \quad (3)$$

Where m is the mass of the levitating object and g is the gravitational acceleration. However, maintaining stable levitation requires continuously adjusting B based on the magnet's position, which is done through feedback control.

5 Sensor Calibration and Placement

The levitation system used two Hall-effect sensors, one placed above the solenoid and the other below, to measure the magnetic field changes as the magnet moved.

5.1 Hall-Effect Sensors

These sensors detect the magnetic field by converting it into a proportional voltage. A Hall-effect sensor's output voltage (V_H) is described by:

$$V_H = K_H \cdot B + V_0 \quad (4)$$

Where:

- K_H is the sensitivity of the sensor (in volts per Tesla),
- B is the magnetic flux density at the sensor's position,
- V_0 is the output when no field is present (typically 2.5V for 5V-supplied sensors).

By placing one sensor **above** and one **below** the levitating magnet, the system can detect the difference between the field created by the solenoid and the field created by the magnet. This configuration allows the system to isolate the magnet's contribution, critical for accurate position sensing.

The output from each sensor:

$$S_1 = B_{\text{magnet}} + B_{\text{solenoid}}, \quad S_2 = B_{\text{solenoid}} \quad (5)$$

Taking the difference:

$$S_1 - S_2 = B_{\text{magnet}} \quad (6)$$

This eliminates the solenoid's field from the measurement, leaving only the magnet's field for position feedback.

5.2 Sensor Output to Distance Conversion

The output voltage of the Hall-effect sensor can be converted into a corresponding distance using a mathematical model derived from the calibration data. The non-linear relationship between voltage and distance was approximated by the following exponential equation:

$$Z = a + b \cdot e^{\frac{-V_H}{c}} + d \cdot e^{\frac{-V_H}{e}} \quad (7)$$

Where:

- Z is the distance from the magnet to the sensor (in cm),
- V_H is the voltage output from the Hall-effect sensor (in Volts),
- a, b, c, d, e are constants determined through the calibration process.

6 Circuit Design

The electrical system consisted of several key components to drive the solenoid and process sensor signals.

6.1 Current Amplification

The Arduino alone cannot supply enough current to the solenoid to produce the necessary magnetic field. A **TIP112 Darlington transistor** was used to amplify the control signal from the Arduino. The transistor operates in two states: cutoff (no current through the solenoid) and saturation (maximum current through the solenoid).

The transistor's base current controls the output current through the solenoid, where:

$$I_c = \beta \cdot I_b \quad (8)$$

With a high current gain ($\beta \approx 1000$), the transistor ensures that small adjustments from the Arduino result in significant changes in the solenoid's current.

6.2 Difference Between Sensors and Amplifier

The difference between the two Hall-effect sensor outputs was calculated using an electronic circuit. Specifically, an operational amplifier (op-amp) in a differential configuration was employed to subtract the output of the bottom sensor (S_2) from the top sensor (S_1). This method electronically isolates the magnetic field (B_{solenoid}) generated by the magnet from the field of the solenoid.

The output voltage was fed into an operational amplifier circuit. This amplified the small voltage differences between the sensors, making the system more sensitive to minor positional changes. The amplified output was then fed to the Arduino's analog input for processing.

7 Control Theory: The PID Algorithm

The core of the system's stability lies in the **PID controller**, which dynamically adjusts the solenoid's current based on the magnet's position error.

7.1 PID Control Equation

The PID controller adjusts the output by summing three terms:

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (9)$$

Where:

- $u(t)$ is the control signal (PWM value),
- $e(t)$ is the error (difference between desired and actual position),
- K_p is the proportional gain,
- K_i is the integral gain,
- K_d is the derivative gain.

7.2 Code Implementation

The Arduino code implemented the PID control by reading the sensors, computing the error, and adjusting the PWM signal accordingly. Here's a simplified version of the key code:

```
double Kp = 2.0, Ki = 0.5, Kd = 1.0;
double setPoint = 0.0; // Desired magnet position
double input, output;
double lastInput, I, error;

void loop() {
```

```

    input = readSensor(); // Get sensor data
    error = setPoint - input;

    // Proportional term
    double P = Kp * error;

    // Integral term
    I += (Ki * error);

    // Derivative term
    double D = Kd * (input - lastInput);

    // Compute the PID output
    output = P + I - D;

    // Apply the output to the solenoid (PWM)
    analogWrite(SOLENOID_PIN, output);

    // Update last input for next iteration
    lastInput = input;
}

```

This code continuously monitors the magnet's position, calculates the PID values, and adjusts the solenoid's current via PWM.

8 System Calibration

To optimize the system's performance, the sensors were calibrated, and the PID constants were fine-tuned experimentally. Calibration involved measuring the sensor outputs at known positions and adjusting the gains (K_p , K_i , K_d) for stable levitation.

The calibration also involved characterizing the system's dynamic response, ensuring the magnet would not oscillate excessively. The proportional gain was adjusted to react quickly, while the derivative gain was tuned to minimize overshooting.

9 Conclusion

The magnetic levitation system successfully demonstrated stable levitation using electromagnetic control. The integration of Hall-effect sensors, feedback control, and a well-tuned PID algorithm allowed for precise vertical position regulation. Despite the system's non-linear nature, the use of two sensors and a differential measurement strategy provided accurate feedback. The project's outcome met the initial goals of levitating a magnet with a small, controllable range of motion.