6.320 Midterm Project

Magnetic Levitation with One Hall Effect Sensor

Vibha Agarwal and Franklin Zhang
May 2019

1. Overview and Motivation

(Authored by Franklin Zhang)

On a previous lab, we used an analog controller to stabilize a magnetic levitation system. In that system, two Hall effect sensors were required to determine the distance of the magnet from the bottom of the electromagnetic coils. The measured strength of the magnetic field can be used as a proxy for the distance. To gain a better understanding of the system, we wanted to rebuild the system and the controller but use only one Hall effect sensor. However, a challenge arises when one sensor is removed; a single Hall effect sensor will pick up not only the field of the magnet, but also the field of the coils. This makes the process of designing a controller more difficult. In this project we explored the possibility of stabilizing the system using only one Hall effect sensor. Figure 1 illustrates the set-up we used in the previous lab; we used a similar set-up with the top Hall effect sensor removed.

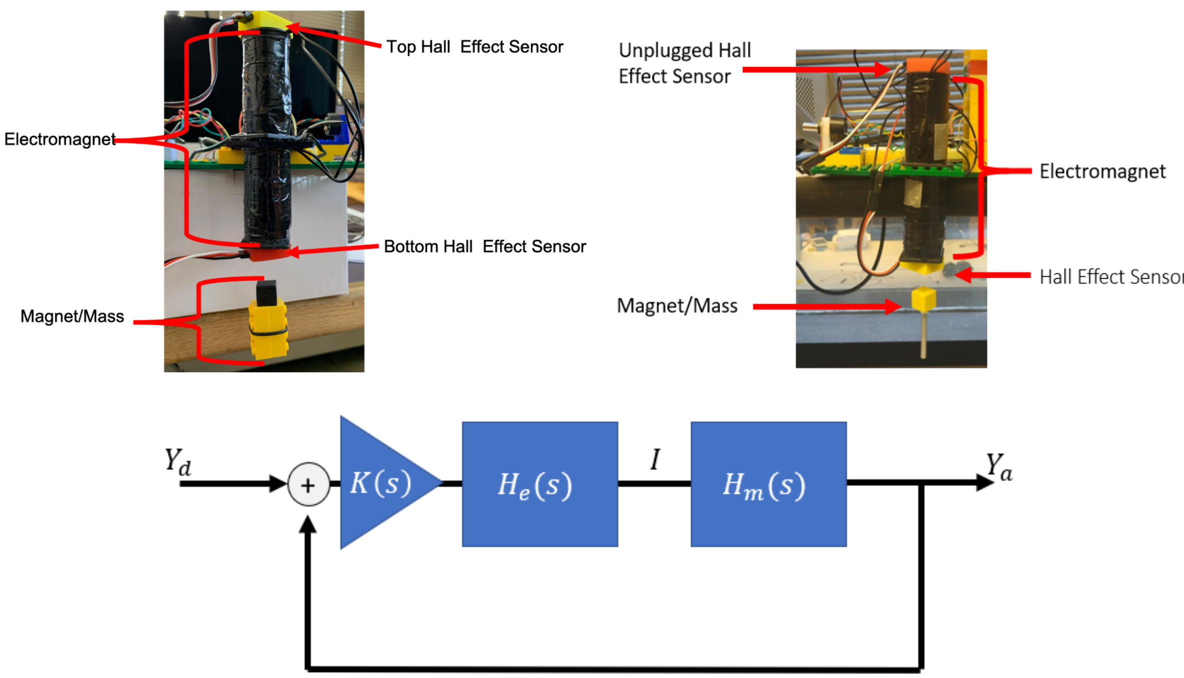

Figure 1: *(Top Left)* The original two Hall effect sensor setup. *(Top Right)* The new one Hall effect sensor setup. (*Bottom)* The closed loop control diagram for the original two Hall effect sensor.

2. Self Field Cancellation

(Authored by Franklin Zhang)

In the original system, two Hall effect sensors were placed at the opposite ends of the solenoid. In that system, the sensor was calibrated to output 2.5 V for no magnetic field, and ranged from 0V to 5V depending on the magnitude and direction of the magnetic field. When current flows through the coils, one sensor's reading increases by a value ΔV and the other sensor's value decreases by a value ΔV. If you connect the output of the two sensors (2.5+ΔV and 2.5-ΔV) to a voltage divider, the output is 2.5 V. Thus having this two Hall effect sensor setup removes the self-field measurements. However, because our system only has one Hall effect sensor, this self-field cancellation doesn't occur, and our sensors output a value that includes the changing magnetic field of the solenoid, in addition to the value we are looking for, the change in voltage due to change in magnet position.

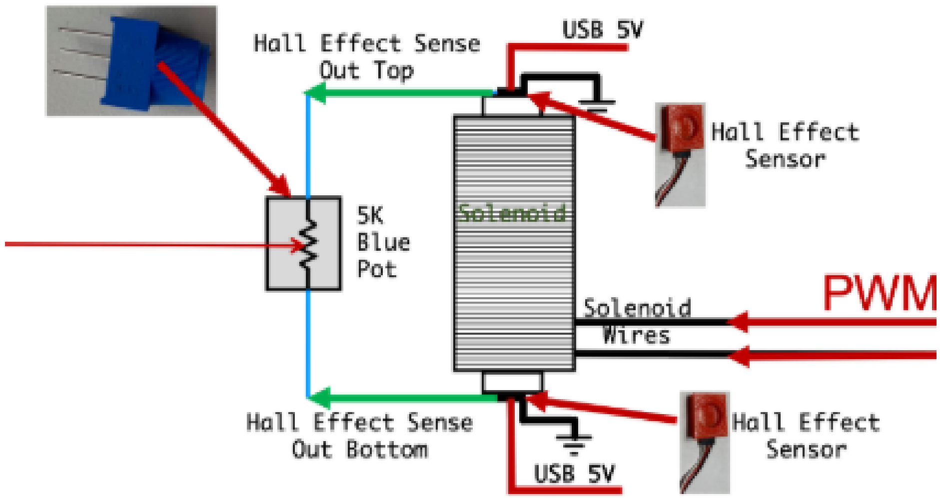

Figure 2: The circuit diagram for the solenoid and Hall effect sensors. The potentiometer was used as a voltage divider that does the self-field cancellation.

However, with only one coil, the output of the coil will include that ΔV value, representing the voltage reading from the self field. That extra value propagates through the control loop and effects the output of the system. Therefore, we must account for this ΔV in our overall control system diagram, as explained in Section 3.

3. System Overview

(Authored by Franklin Zhang)

For a system with only one Hall effect sensor, we needed to create a new model because unlike the previous system, the measured distance is not equal to or proportional to the actual distance due to the introduction of the solenoid field in the sensor readings. Instead, the measured distance is equal to the sum of a value proportional to the distance (same as before) and a term proportional the solenoid current (because of the lack of self-field cancellation). The solenoid field is proportional to the solenoid current.

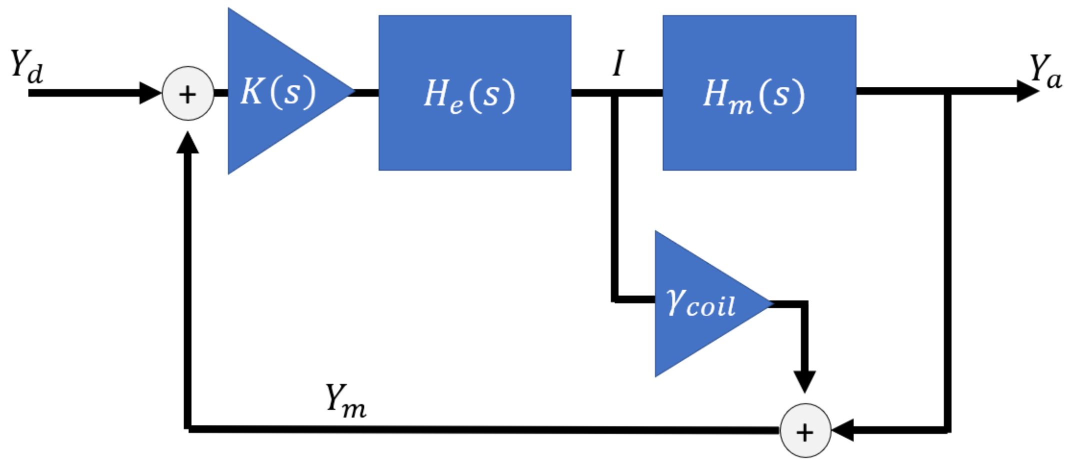

Figure 3: The complete system diagram for a one Hall effect sensor magnetic levitation system. $K(s)$ represents the compensator, $H_e(s)$represents the electronic pole of the plant, $H_m(s)$ represents the 2 mechanical poles of the plant, and the γ_{coil}represents the constant factor between current in the coil and resulting electromagnetic field.

$$\frac{y_a}{y_d} = G(s) = \frac{K(s)H_e(s)H_m(s)}{1 + \gamma_{coil} K(s)\, H_e(s) + K(s)\, H_e(s)\, H_m(s)} \quad (1)$$

4. Angle Sensor Sensitivity

(Authored by Vibha Agarwal)

To start building our compensator, proportional control needed to be implemented to introduce oscillations and increase the stability of our system; however, that required adjusting the angle sensor sensitivity. We figured this out because with the original sensitivity setting, we weren't able to find a suitable Kp value. We determined that this was because the voltage output of the Angle Sensor potentiometer was scaled by $\frac{R_1}{R_1+R+2} = \frac{10k\Omega}{110k\Omega} \approx 0.09$before being fed into V^-, meaning that V^- would only range from 0V to .5V. This was not enough of a range to see the effects of adjusting the angle sensor. Once we replaced the $100k\Omega$resistor with a $10k\Omega$ resistor (see Figure 4), the scaling factor increased to $\frac{R_1}{R_1+R+2} = \frac{10k\Omega}{20k\Omega} \approx 0.5$k , which meant that the range of V^- was now 0V to 2.5V, enough to see the impact of turning the Angle Sensor. After this adjustment, we were better able to find a Kp term.

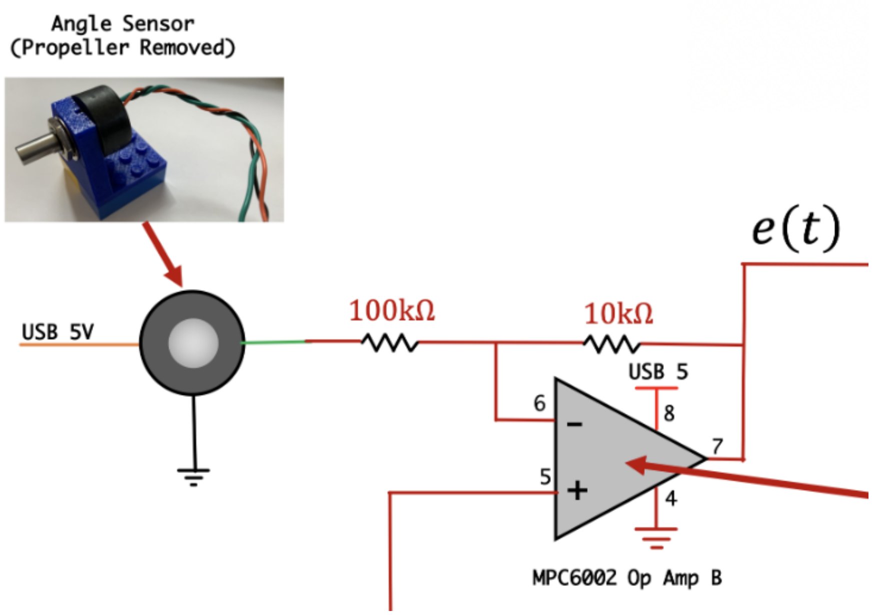

Figure 4: We replaced the $100k\Omega$ resistor with a $10k\Omega$ resistor to increase the effect of the angle sensor reading.

5. Finding The Electronic Pole:

(Authored by Vibha Agarwal)

In order to create an effective compensator, we first need to model our physical system. One property that we are interested in is the electronic pole, which defines how quickly a commanded voltage is converted into a change in magnetic field from the solenoid. To calculate the electronic pole, we looked at the step response of the system and measured how long it took the output to rise to 2/3 of the steady state response.

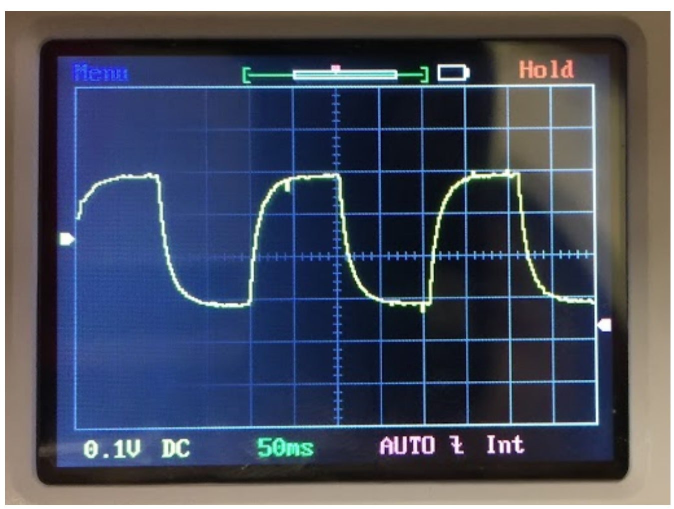

Figure 5: The step response of the Mag-Lev system. The time it takes to get to two-thirds of the steady state response is 15ms, or .015s.

$$p \ = \ -\frac{1}{\tau}\ where\ \tau\ is\ the\ time\ it\ takes\ to\ get\ to\ \frac{2}{3}\ of\ steady\ state$$

(2)

$$p \ = \ -\frac{1}{.015} = \ -66.66$$

From Figure 5, we knew that the time constant is .01s, and calculated the electronic pole to be -66.66, as shown in Equation 2.

6. Finding The Mechanical Poles

(Authored by Vibha Agarwal)

In addition to finding the electronic pole of our system, we calculated a rough approximation for the two mechanical poles. To do this, we used simple mechanics equations relating distance and time and assumed a constant acceleration to calculate the poles. We held the magnet 1 centimeter (.01m) from the coil, and timed how long it took to snap to the coil when we let go; it took .2 seconds. From this, we were able to calculate the value of the constant between change in position and acceleration, or γ_{dy2a}, as seen in Equations 3-5.

$$\frac{d^2}{dt^2}\Delta y \ = \ \alpha * \Delta y \ = \ \gamma_{dy2a}\Delta y \qquad (3)$$

$$\Delta x \ = \ \frac{1}{2}\alpha(\Delta t)^2\ where\ \Delta x \ = \ .01m,\ \Delta t = .2s \qquad (4)$$

$$\alpha \ = \ .5$$

$$\gamma_{dy2a} = \ 50$$

$$\Delta Y \ = \ \frac{\gamma di2a}{s^2 - \gamma dy2a}\Delta I \qquad (5)$$

$$p_m = \ \pm\ 50$$

Once we found the value of γ_{dy2a}, we used Equation 5 to calculate the two mechanical poles, which are the square roots of γ_{dy2a}, or $p_m = \ \pm\ 50$.

7. Making the Controller

(Authored by Franklin Zhang)

We used an analog feedback controller to stabilize the Mag-Lev system with one Hall effect sensor. Using an op-amp, the error term was scaled by a factor $\frac{R_f}{R_{in}}$ and this term was used as K_p in the control loop. Although a proportional controller isn't able to stabilize the system by itself, it was able to make the magnet oscillate. The analog control circuit is shown in Figure 6.

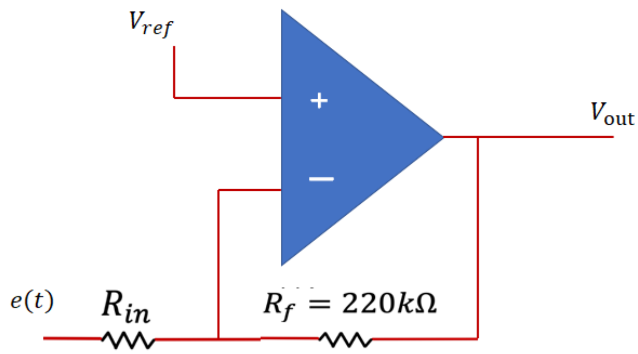

Figure 6: The circuit used to make a continuous time proportional controller. The gain is determined by the ratio of the two resistors.

In order to better understand the system, we plotted a bode plot (Figure 7) of the open loop non-compensated system with Equation 6.

$$H_{noncomp}(s) = \gamma_{coil} K(s) H_e(s) + K(s) H_e(s) H_m(s) \quad (6)$$

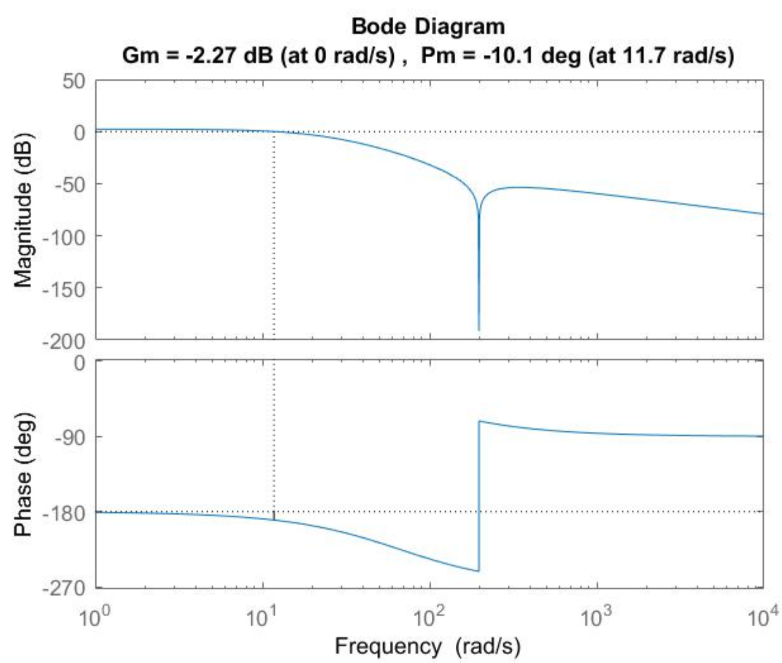

Figure 7: The bode plot of the non-compensated open loop transfer function. Because some of the constants are estimated, this isn't an accurate model of the true system, however we can see that with our estimated parameters, we have a very small phase margin that requires compensation.

Because of the small phase margin predicted by plotting the bode plot and the apparent instability of the physical system, further compensation was required. With lead compensation, a phase bump can be added to increase the phase margin at the unity gain frequency to further stabilize the system. The circuit for the lead compensator is shown in Figure 8. The poles and zeros were chosen so that the last pole of the non-compensated system falls between the zero and the pole of the compensator. The new compensated system was estimated to have a larger phase margin. In the physical world, adding the compensator circuit increased the system's stability.

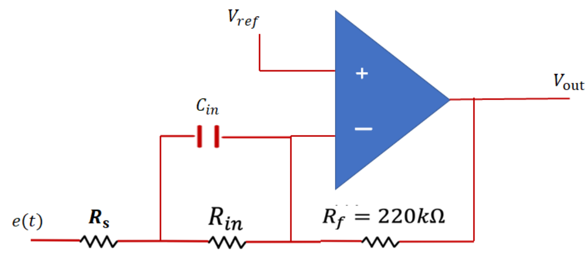

Figure 8: The circuit used to create a lead compensator. The compensator transfer function is shown in Equation 5.

$$H_c(s) = -\frac{R_f}{R_s}\left(\frac{s-z}{s-p}\right) \qquad (7)$$

$$z = \frac{-1}{C_{in}R_{in}}, \quad p = \frac{-1}{C_{in}R_{in}}$$

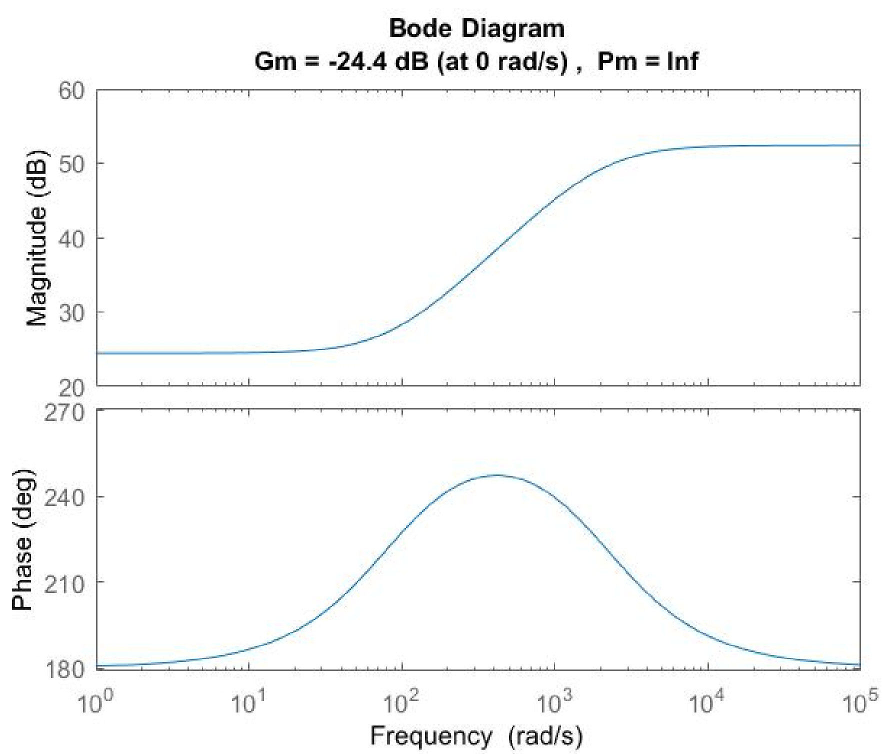

Figure 9: The bode plot of the lead compensator will give about 180 degrees of phase bump at high and low frequencies and 257 degrees of phase bump at around frequencies of 400 rad/s.

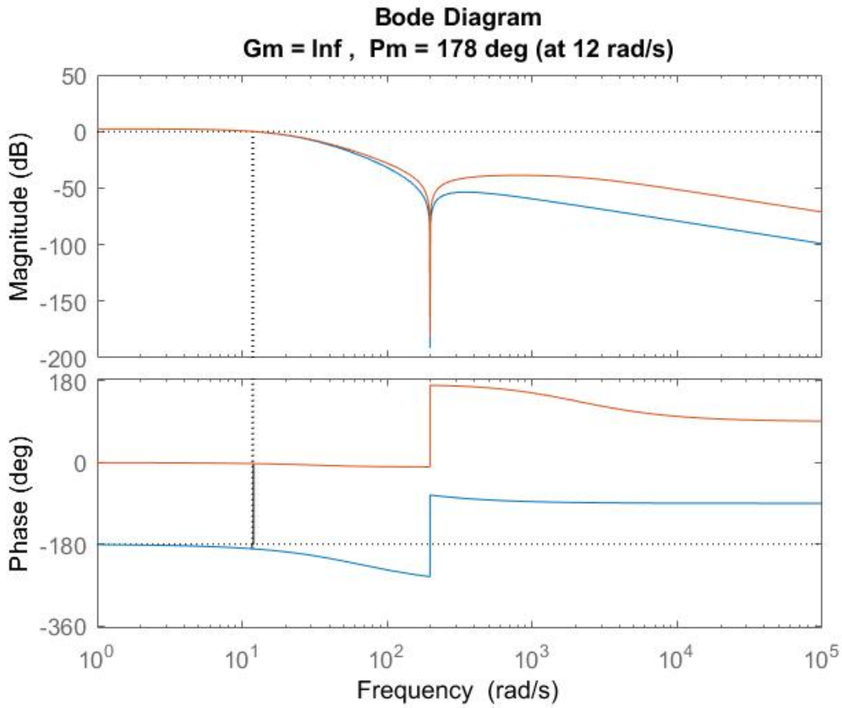

Figure 10: The blue plot is the image of the original open-loop function and the orange plot is the compensated open-loop function. The phase margin went from 10 degrees to 178 degrees. This shows that the system is much more stable.

8. Results

(Authored by Vibha Agarwal)

After numerous iterations, we were able to make the magnet float with only one Hall effect sensor. We could adjust the position of the magnet by turning the angle sensor potentiometer, but only within a range of approximately 1 centimeter. Furthermore, the magnet was unable to float indefinitely, and would fall to the ground after about 30 seconds of floating. We determined this was because our approximations for the mechanical poles were not completely accurate, given that we assumed a constant acceleration, which is clearly untrue. Despite this, we declared the project a success, as our compensator was more or less stabilizing the system as desired. The final magnet levitation with one Hall effect sensor can be seen in Figure 11.

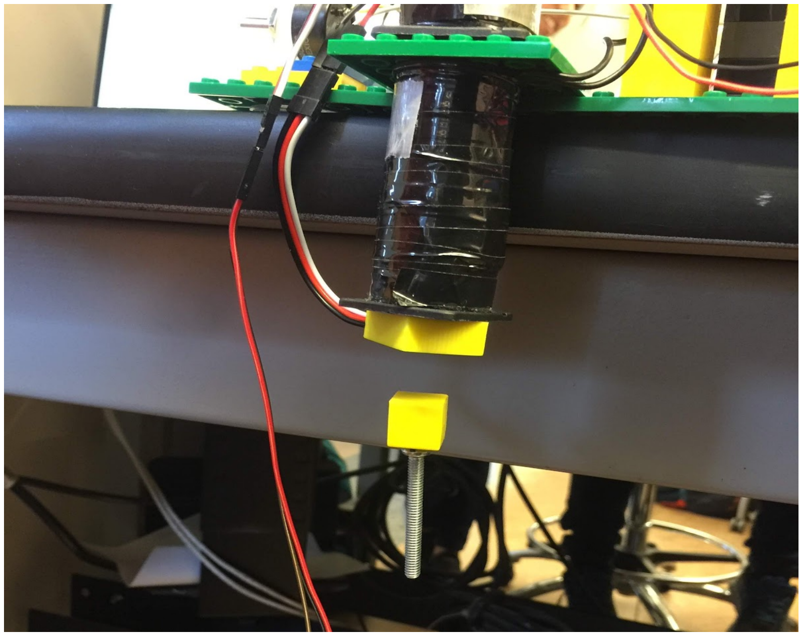

Figure 11: A magnet levitating with our compensated system using only one Hall effect sensor.

9. Lessons Learned

(Authored by Vibha Agarwal)

Many concepts from this class were tied together very well through this project. For example, deriving the transfer function equation (Equation 1) was difficult given that its set-up is of a form that we never encountered in class. Figuring out how to create a system diagram (Figure 3) and manipulate equations to get the right transfer function cleared up what each part of the system contributes to the control loop.

In addition, finding the electronic pole was valuable, because previously it was unclear what parts of the lab were for learning about the model and what parts were for controlling the system, but doing it again through the project made the line between the two tasks more distinct. Finally, determining the best compensator to stabilize our system reinforced our understanding of lead and lag compensation. Our guesses for the resistance and capacitance values were smarter than before, and we were able to clearly see how increasing certain values affected the phase bumps. Because we also (unsuccessfully) tried implementing a lag compensator, we learned more about the differences between the two.

Appendix A: MATLAB Code

```
% CONSTANTS %
Rf = 200000;
Rin = 12000;
Kp = Rf/Rin;
Rs = 1000;
Cin = 10*10^(-6);

% TRANSFER FUNCTION SETUP %
s = tf('s');
gamma_di2a = 20; %gamma_di2a
gamma_dy2a = 50; %gamma_dy2a FOUND
Hm = gamma_di2a/(s^2 - gamma_dy2a); %mechanical poles
He = 66/(s+66); %guess because electronic pole = -66.66
gamma_coil = 0.10; %can be pos or neg, try both

% COMPENSATOR %
z = -1/(Cin*Rin);
p = -1/(Rs*Cin);

%Cin = -1/(Rin*z)
%Rs = -1/(Cin*p)
Hc = (-Rf*(s-z))/((s-p)*Rs);
figure(1); clf; margin(Hc)

% PLOTTING %
nonCompH = Kp*Hm*He + Kp*He*gamma_coil
compH = Hc*Hm*He + Hc*He*gamma_coil;
G = compH/(1 + compH); %numerator doesn't matter
G_fdbk = feedback(1, compH);
margin(nonCompH); hold on;
margin(-compH)
legend("nonCompH", "compH")
hold off;
pole(nonCompH)
pole(compH)
```