

## Lab Assignment 2: MagLev

### TEAM MEMBERS:

Shiva Ghose, @gshiva

John Peterson, @jrpeters

Peter Turpel, @pturpel

Chan-Rong Lin, @pmelin

—

### Teamwork Participation Pledge :: Team 1

I attest that I have made a fair and equitable contribution to this lab and submitted assignment.

My signature also indicates that I have followed the University of Michigan Honor Code, while working on this lab and assignment.

I accept my responsibility to look after all of the equipment assigned to me and my team, and that I have read and understood the X50 Lab Rules.

Name	Email	Signature
Shiva Ghose	gshiva@umich.edu	
John Peterson	jrpeters@umich.edu	
Peter Turpel	pturpel@umich.edu	
Chan-Rong Lin	pmelin@umich.edu	

## Part B

### Question 1

a.

#### Break Beam Sensor

##### Assumptions

- Neglect the effects of diffraction.
- Light rays are parallel.
- Neglect reflection of IR light by the polished ball stand.

As a consequence of these assumptions the IR beam from the LED to the photo-transistor can be modeled as a cylinder with a radius approximately equal to the radius of the phototransistor. The shadow cast by the sphere is then simply a circle with radius equal to the radius of the sphere blocking part of the cross section of the beam. These assumptions also mean that the distance between the sphere and the IR LED does not matter. The only values that matter are the radius of the IR beam, the radius of the sphere, and the perpendicular distance between the axis of the beam and the center of the sphere. The amount of light received by the photo-transistor can then simply be modeled as the intersection of two circles in 2-dimensions as shown in figure 1.

$$Radiance\ Fraction = \begin{cases} \frac{A_{beam} - A_{sphere}}{A_{beam}} & \text{if } r_{sphere} + |d| < r_{beam} \\ 0 & \text{if } r_{beam} + |d| < r_{sphere} \\ 1 & \text{if } |d| < r_{beam} + r_{sphere} \\ \frac{A_{beam} - A_{Overlap}}{A_{beam}} & \text{if } |d| < r_{sphere} + r_{beam} \end{cases}$$

Where:

$$\begin{aligned} A_{beam} &= \pi r_{beam}^2 \\ A_{sphere} &= \pi r_{sphere}^2 \\ y &= \frac{d^2 + r_{beam}^2 - r_{sphere}^2}{2d} \\ d_{beam} &= y \\ d_{sphere} &= d - y = \frac{d^2 - r_{beam}^2 + r_{sphere}^2}{2d} \\ A_{Obeam} &= r_{beam}^2 \cos^{-1}\left(\frac{d_{beam}}{r_{beam}}\right) - d_{beam} \sqrt{r_{beam}^2 - d_{beam}^2} \\ A_{Osphere} &= r_{sphere}^2 \cos^{-1}\left(\frac{d_{sphere}}{r_{sphere}}\right) - d_{sphere} \sqrt{r_{sphere}^2 - d_{sphere}^2} \\ A_{Overlap} &= A_{Obeam} + A_{Osphere} \end{aligned}$$

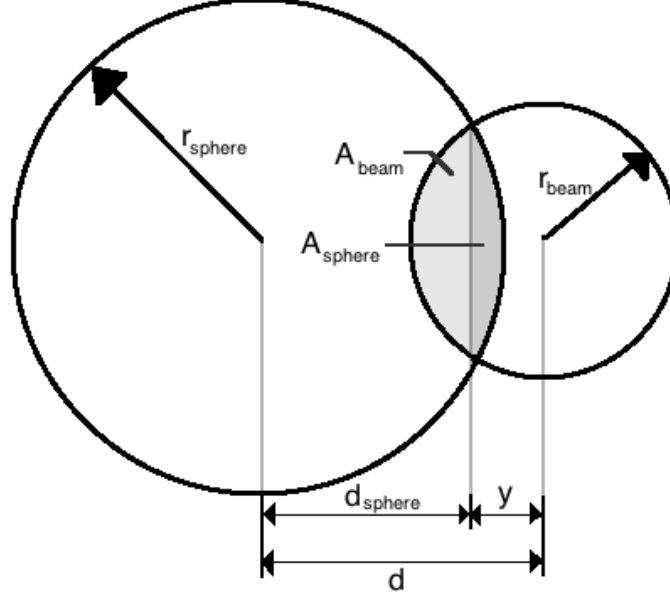


Figure 1: *Planar approximation of IR beam occluded by steel sphere*

## Driver Circuit

### Assumptions

- Op-Amp Golden Rules: voltage at inverting and non-inverting terminals is equal and op-amp input impedance is infinite

The electromagnet driver is a simple op-amp circuit, shown in figure 2, which is designed to pass a specific current through the electromagnet for a given input voltage. The voltage at the non-inverting input can be simply determined by applying the voltage divider rule. The voltage at the inverting input is now known allowing us to apply Ohm's Law to determine the current through  $R_3$  and therefore through the electromagnet. Notice that the resistance of the electromagnet does not affect the current provided by the driver circuit.

$$V^+ = \frac{R_2}{R_1 + R_2} V_{in} \quad I_{EM} = \frac{V^+}{R_3} = \frac{1}{R_3} \left( \frac{R_2}{R_1 + R_2} \right) V_{in} = D V_{in}$$

## Electromagnet

### Assumptions

- $\mu_{Al} \approx \mu_{Cu} \approx \mu_{air}$ , where  $\mu_x$  is the absolute permeability of material  $x$
- $\mu_{steel} \gg \mu_{air}$
- We can model the steel sphere as a cylinder with its axis collinear with the electromagnet

$$A_{cd} \approx A_s = \pi * r_{sphere}^2$$

- We can neglect the effects of fringing on the cross-sectional area of paths through the air and assume particular dimensions that are consistent with the solid objects in the system

$$A_{EM} = \pi * r_{EM}^2 \approx A_{ab} \quad A_s \approx A_{bc} \approx A_{de} \quad A_{EM} \approx A_{be} + A_{bc}$$

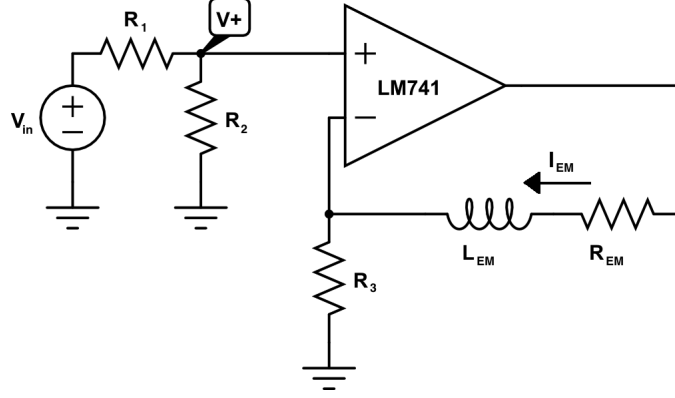


Figure 2: *Electromagnet Driver Circuit*

- Path length from b to c (see figure 2) does not vary with  $x$ :  $\frac{\partial}{\partial x}(L_{bc}) = 0$
- The sphere remains in line with the axis of the electro magnet and in front of the magnet ( $x > 0$ )

Drawing out approximate flux paths yields the diagram in figure 3, which can be converted to an equivalent magnetic circuit, shown in figure 4, by applying Ampere's Law and Maxwell's 2<sup>nd</sup> law resulting in the following three equations.

$$NI = \phi (\mathbb{R}_{EM} + \mathbb{R}_{ab}) + \phi_{main} (\mathbb{R}_{bc} + \mathbb{R}_s + \mathbb{R}_{dc})$$

$$NI = \phi (\mathbb{R}_{EM} + \mathbb{R}_{ab}) + \phi_{leak} (\mathbb{R}_{be})$$

$$\phi = \phi_{main} + \phi_{leak}$$

$$\mathbb{R}_{EM} = \frac{L_{EM}}{A_{EM}\mu_{air}} \quad \mathbb{R}_{ab} = \frac{L_{ab}}{A_{EM}\mu_{air}} \quad \mathbb{R}_{bc} = \frac{L_{bc}}{(A_s)\mu_{air}} \quad \mathbb{R}_{de} = \frac{x}{A_{A_s}\mu_{air}}$$

$$\mathbb{R}_{be} = \frac{L_{be}}{(A_{EM} - A_s)\mu_{air}} \quad \mathbb{R}_s = \frac{2 * r_{sphere}}{A_s\mu_{steel}} \approx 0$$

Solving the 2<sup>nd</sup> equation for  $\phi_{leak}$  gives:

$$\phi_{leak} = C_0 (NI - C_1\phi_{main})$$

$$C_0 = \frac{\mu_{air}A_{EM}(A_{EM} - A_s)}{L_{be}A_{EM} + (L_{EM} + L_{ab})(A_{EM} - A_s)} \quad C_1 = \frac{L_{EM} + L_{ab}}{\mu_{air}A_{EM}}$$

Substituting back into the 1<sup>st</sup> equation lets us solve for  $\phi_{main}$  yielding:

$$\phi_{main} = \frac{NI(1 - C_0C_1)}{C_1 - C_0C_1^2 + \frac{L_{bc}}{\mu_{air}A_s} + \frac{x}{\mu_{air}A_s}}$$

With equations for our fluxes, we can now compute the self inductance and from that compute the force exerted on the steel sphere as a function of position and current.

$$\lambda = N\phi = L(x)I$$

$$W(i, x) = (1/2)L(x)I^2 = (1/2)N\phi(x)I = (1/2)NI(\phi_{main}(x) + \phi_{leak}(x))$$

$$F(i, x) = \frac{d}{dx}(W(i, x)) = (1/2)NI\left(\frac{d}{dx}(\phi_{main}) + \frac{d}{dx}(\phi_{leak})\right)$$

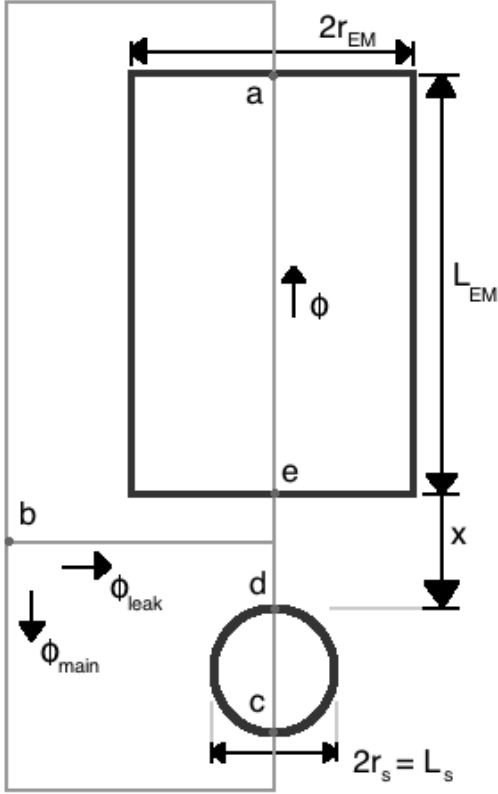


Figure 3: *Approximate Flux paths taken through our system*

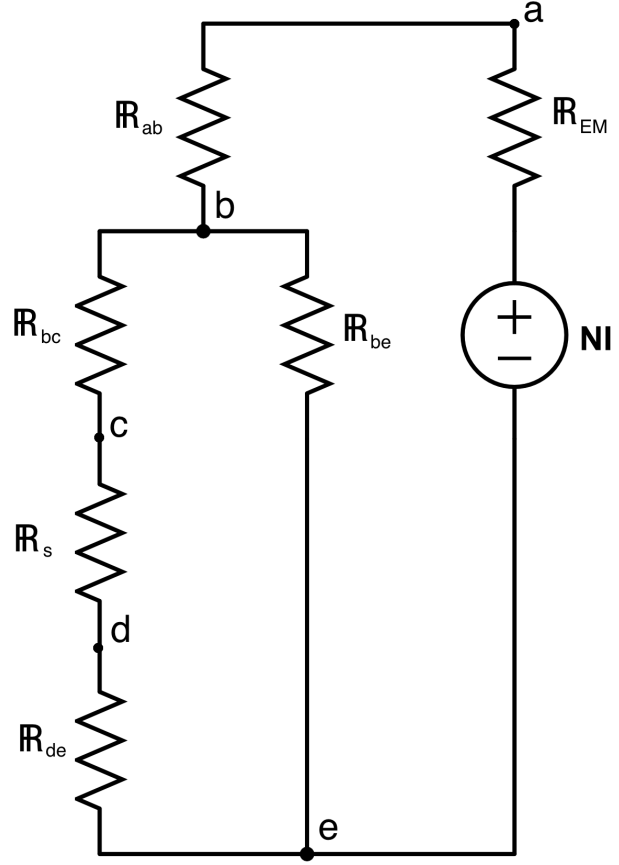


Figure 4: *Equivalent Magnetic Circuit*

$$\frac{d}{dx}(\phi_{main}) = -\frac{NI(1 - C_0C_1\mu_{air}A_s)}{(\mu_{air}A_s(C_1 - C_0C_1^2 + C_2) + x)^2}dx$$

$$\frac{d}{dx}(\phi_{leak}) = -C_0C_1\frac{d}{dx}(\phi_{main})$$

Combining terms yields our final expression for the force.

$$F_{EM}(x) = -\frac{1}{2}(1 - C_0C_1)^2 \frac{\mu_{air}A_sN^2}{(\mu_{air}A_s(C_1 - C_0C_1^2 + C_2) + x)^2}I^2$$

We can then condense these terms to arrive at a simple equation to model the force, shown below. Where values for A and B can be obtained experimentally.

$$F_{EM}(x) = -\frac{AI^2}{(B + x)^2}$$

## Steel Sphere & Magnet System

### Assumptions

- Air resistance is negligible.
- The steel sphere does not collide with the magnet:  $x > 0$

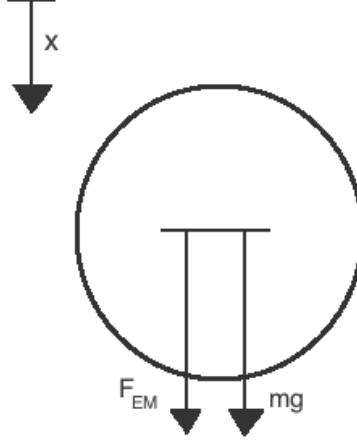


Figure 5: Free body diagram of steel sphere. Note: The  $x$  axis points downwards and our force was computed according to this convention, which is why there is a negative sign in the expression of the force generated by the electromagnet.

According to Newton's second law the sum of the forces on an object is proportional to the acceleration of the object. After constructing a free body diagram of our system shown in figure 5 we can very easily derive a model of the motion of our system.

$$\sum F_x = mg + F_{EM} = mg - \frac{AI^2}{(B+x)^2} = m\ddot{x}$$

b.

#### Break Beam Sensor

The position sensor is very nearly linear over the range of positions that we are concerned with but does saturate for small and large values for  $x$ . To linearize the sensor we must perform a 1st order Taylor approximation which requires the derivative of the function about the nominal value. The numerical derivative is easily obtained, but the symbolic computation of the derivative, shown below, is far more involved.

$$\begin{aligned} d &= x + 4.12 & \partial x &= \partial d \\ y &= \frac{d^2 + r_{beam}^2 - r_{sphere}^2}{2d} = \frac{d}{2} + \frac{r_{beam}^2}{2d} - \frac{r_{sphere}^2}{2d} & \frac{\partial}{\partial d} y &= \left( \frac{1}{2} - \frac{r_{beam}^2}{2d^2} + \frac{r_{sphere}^2}{2d^2} \right) \partial d \\ d_{beam} &= y & \frac{\partial}{\partial d} d_{beam} &= \frac{\partial}{\partial d} y = \left( \frac{1}{2} - \frac{r_{beam}^2}{2d^2} + \frac{r_{sphere}^2}{2d^2} \right) \partial d \\ d_{sphere} &= d - y & \frac{\partial}{\partial d} d_{sphere} &= 1 \partial d - \frac{\partial}{\partial d} y = \left( 1 - \frac{1}{2} + \frac{r_{beam}^2}{2d^2} - \frac{r_{sphere}^2}{2d^2} \right) \partial d = \left( \frac{1}{2} + \frac{r_{beam}^2}{2d^2} - \frac{r_{sphere}^2}{2d^2} \right) \partial d \\ A_{Obeam} &= r_{beam}^2 \cos^{-1} \left( \frac{d_{beam}}{r_{beam}} \right) - d_{beam} \sqrt{r_{beam}^2 - d_{beam}^2} \\ \frac{\partial}{\partial d} A_{Obeam} &= \left( -\frac{r_{beam}}{\sqrt{1 - \left( \frac{d_{beam}}{r_{beam}} \right)^2}} - (r_{beam}^2 - d_{beam}^2)^{1/2} + d_{beam} (r_{beam}^2 - d_{beam}^2)^{-1/2} \right) \frac{\partial d_{beam}}{\partial d} \end{aligned}$$

$$A_{Osphere} = r_{sphere}^2 \cos^{-1}\left(\frac{d_{sphere}}{r_{sphere}}\right) - d_{sphere} \sqrt{r_{sphere}^2 - d_{sphere}^2}$$

$$\frac{\partial}{\partial d} A_{Osphere} = \left( -\frac{r_{sphere}}{\sqrt{1 - \left(\frac{d_{sphere}}{r_{sphere}}\right)^2}} - (r_{sphere}^2 - d_{sphere}^2)^{1/2} + d_{sphere}^2 (r_{sphere}^2 - d_{sphere}^2)^{-1/2} \right) \frac{\partial d_{sphere}}{\partial d}$$

$$A_{Overlap} = A_{Obeam} + A_{Osphere} \quad \frac{\partial}{\partial d} A_{Overlap} = \frac{\partial}{\partial d} A_{Obeam} + \frac{\partial}{\partial d} A_{Osphere}$$

The set point we are linearizing about is in the area where the function above is well behaved:

$$Radiance\ Fraction = \frac{A_{beam} - A_{Overlap}}{A_{beam}} = 1 - \frac{A_{Overlap}}{A_{beam}}$$

$$\frac{\partial}{\partial d} (Radiance\ Fraction) = \frac{1}{A_{beam}} \left( \frac{\partial}{\partial d} A_{Overlap} \right)$$

Taylor expansion allows us to approximate the Radiance Fraction about a certain distance  $d_0$  as:

$$Radiance\ Fraction = R(d) = R(d_0) + \left. \frac{\partial R(d)}{\partial d} \right|_{d_0} (d - d_0) + \frac{1}{2!} \left. \frac{\partial^2 R(d)}{\partial d^2} \right|_{d_0} (d - d_0)^2 + \dots$$

Dropping the higher order terms leaves us with a linear expression for the Radiance Fraction about a certain distance.

$$R(d) \approx R(d_0) + \left. \frac{\partial R(d)}{\partial d} \right|_{d_0} (d - d_0)$$

Substituting our expression for d in terms of x yields:

$$R(x) \approx R(x_0 + 4.12) + \left. \frac{\partial R(x)}{\partial x} \right|_{x_0 + 4.12} (x - x_0)$$

Substituting our previous expressions in to obtain a single symbolic linear expression for the Radiance Fraction in terms of  $x$ ,  $r_{beam}$  and  $r_{sphere}$  is possible but not particularly practical. The appendix includes MATLAB code for computing  $R'(d_0)$  by computing numerical values for terms before performing the substitution.

To obtain the final expected voltage as a function of x, it is necessary to multiply by the full scale range, in our case 10 volts.

## Electromagnet & Steel Sphere

We can model the dynamic forces of attraction between the ball and the magnet as follows:

$$\Sigma F_x = mg - \frac{Ai^2}{(B+x)^2} = m\ddot{x}$$

**Where:**

$mg$  – is the weight of the ball pulling downwards in the positive x direction

$-\frac{Ai^2}{(B+x)^2}$  – is the non-linear attractive force exerted by the electromagnet

$m\ddot{x}$  – is the net force experienced by the ball

In order to simplify the system we would like to linearize the force of attraction exerted by the magnet around an equilibrium value. This can be achieved using the Taylor series expansion which allows us to represent the function as a sum of derivatives about the point of interest. The Taylor series of a function  $f(a)$  is given by:

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n$$

Which can be expanded to give:

$$f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots$$

Estimating the system up to the first derivative gives us sufficient approximation and results in a linear model. We see that:

$$\frac{Ai^2}{(B+x)^2} \approx \frac{A\bar{i}^2}{(B+\bar{x})^2} - 2\frac{A\bar{i}^2}{(B+\bar{x})^3}\hat{x} + 2\frac{A\bar{i}}{(B+\bar{x})^2}\hat{i}$$

**Where:**

$\bar{i}$  – is the current flowing through the electromagnet at equilibrium

$\hat{i}$  – is the instantaneous current flowing through the electromagnet at the present time

$\bar{x}$  – is the displacement of the ball from the electromagnet at equilibrium

$\hat{x}$  – is the current displacement of the ball from it's equilibrium position.

The expression for the forces acting on the ball can be rewritten as:

$$mg - \left( \frac{Ai^2}{(B+x)^2} - 2\frac{A\bar{i}^2}{(B+\bar{x})^3}\hat{x} + 2\frac{A\bar{i}}{(B+\bar{x})^2}\hat{i} \right) = m\ddot{x}$$

At equilibrium the ball experiences zero net force along the vertical axis as the weight of the ball is negated by the equilibrium force of the magnet by definition.

$$mg = \frac{A\bar{i}^2}{(B+\bar{x})^2}$$

About the equilibrium point, the force acting on the ball can be rewritten as:

$$m\ddot{x} = +2\frac{A\bar{i}^2}{(B+\bar{x})^3}\hat{x} - 2\frac{A\bar{i}}{(B+\bar{x})^2}\hat{i}$$

Taking the Laplace transform of the expression, we get:

$$s^2m\hat{X}(s) = +2\frac{A\bar{i}^2}{(B+\bar{x})^3}\hat{X}(s) - 2\frac{A\bar{i}}{(B+\bar{x})^2}\hat{I}(s)$$

Grouping similar terms, we get:

$$\hat{X}(s) \left( s^2m - \frac{2A\bar{i}^2}{(B+\bar{x})^3} \right) = \hat{I}(s) \left( -\frac{2A\bar{i}}{(B+\bar{x})^2} \right)$$

We can further rearrange the terms to get a ratio of the output to input as follows:

$$\frac{\hat{X}(s)}{\hat{I}(s)} = \frac{-2A\bar{i}(B+\bar{x})}{s^2(m(B+\bar{x})^3) - 2A\bar{i}^2}$$

This is the linearized transfer function of the electromagnet-ball system.



	Expected Radii (mm)	Fit Radii (mm)
Rbeam	2.3650	1.398935
Rblock	6.350	6.070821

Table 1: *Expected and Fit values for beam radius and blocking radius*

### Overall Linear Model

Combining these linearized components allows us to compose an overall linear open loop transfer function from the plant and actuator, the driver circuit, and the sensor:

$$\left( \frac{-2A\bar{i}(B + \bar{x})}{m(B + \bar{x})^3 s^2 - 2A\bar{i}^2} \right) \left( \frac{R_2}{R3(R1 + R2)} \right) * \frac{\partial R}{\partial x}$$

Note that the specific values are explored later in the parameter identification section.

### c.

The sensor has a non-linear relationship between sphere position and output voltage. The primary non-linearity in this system are the two saturations, at zero volts and the full scale value of 10 volts, which occur when the sphere is completely occluding the beam or completely out of the beam respectively. The central region is non-linear, but is well approximated by a linear function. Linear Controls theory may only be strictly applied to linear systems, because the Laplace Transform assumes linear time invariant systems. Because of this assumption we must linearize our sensor about some nominal position. The whole notion of a transfer function does not apply to a non-linear system so attempting to apply Linear Controls without the linearization would be an oversight at best, if the system is pretty close to linear, or a complete disaster if it is not. This non-linearity will cause the actual closed loop response to deviate from the results we would expect given the systems transfer function as we deviate from the nominal position.

### d. Parameter Identification

#### Break Beam Sensor

To determine the model parameters of the break beam sensor model, the beam radius and the effective sphere radius, we acquired data measuring the relationship between the displacement between the optical axis of the phototransistor and the steel sphere and sensor voltage. We sat the sphere on top of the stand and changed the height of the sphere stand and the magnet stand by known amounts, recording the voltage from the phototransistor at each displacement value. Theoretically we would expect the beam radius to be very close to the radius of the phototransistor while we would expect the blocking radius to be very close to the radius of the sphere. Non-linear optimization of the model revealed that that the actual values, shown in table 1 were reasonably close to our expected values. The curve resulting from this fit plotted alongside our data is shown in figure 6.

There is a fixed offset between the displacement, measured from the optical axis of the phototransistor to the center of the sphere, used in figure 6, from the x, measured from the bottom of the magnet to the top of the sphere, used in our controller. This relationship can be obtained by examining the physical dimensions of the system, noted in table 2 yielding the following expression for x in terms of d.

$$x = \Delta + h_{magnet} - h_{stand} - r_s \quad \Delta = d + h_{stand} - h_{sensor}$$

Evaluating this equation and solving for x yields the following relationship:

$$x = d - 4.12 \text{ mm}$$

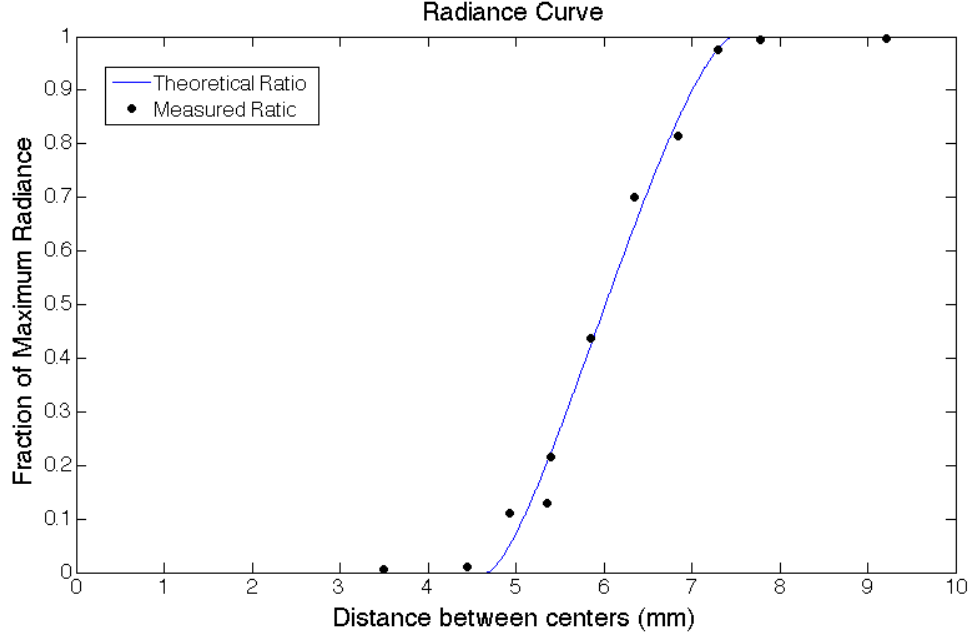


Figure 6: *Normalized Sensor Voltage Output vs distance from the optical axis*

	Dimension (mm)
Magnet Height ( $h_{magnet}$ )	130.73
Sphere on Stand Height ( $h_{sstand}$ )	121.5
Sensor Height ( $h_{sensor}$ )	128.5
Sphere Radius ( $r_s$ )	6.35

Table 2: *Physical Dimensions of the System*

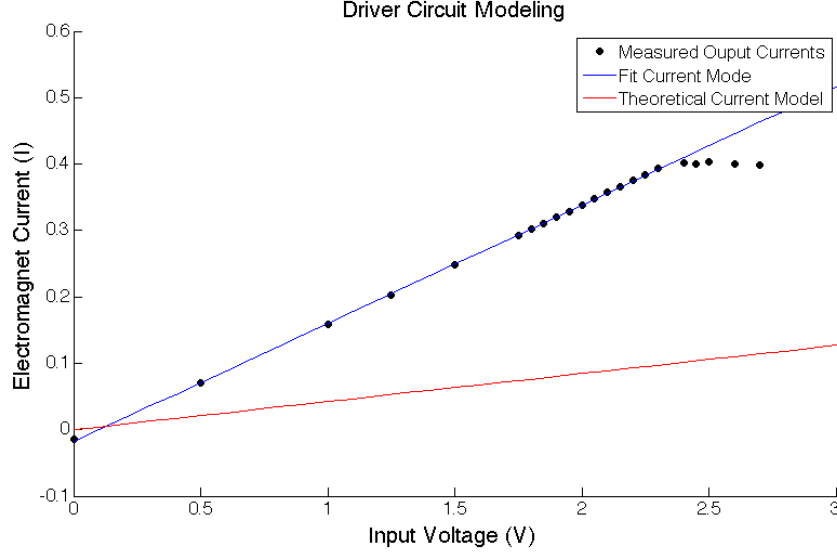


Figure 7: Comparison of actual voltage current relationship with the fit model and the theoretical model

Component	Resistance ( $\Omega$ )
$R_1$	46200
$R_2$	999
$R_3$	0.5

Table 3: Measured Resistances in driver circuit.

### Driver Circuit

Theoretically the conversion from voltage to current for this system should be very straight forwards. We used a multi-meter to measure the resistances, noted in table 3, and then plugged these values into our model yielding the following relationship between current and input voltage.

$$I_{EMExpected} = \frac{1}{R_3} \left( \frac{R_2}{R_1 + R_2} \right) V_{in} = \frac{1}{0.5 \Omega} \left( \frac{999 \Omega}{42.3 k\Omega + 999 \Omega} \right) V_{in} = 0.04233 V_{in}$$

However an examination of the actual current and voltage relationship revealed that the above relationship did not hold. These current measurements were obtained by using the current measuring functionality of the multi-meter. Our observations revealed a steady state current even when the input voltage was 0, so we fit a more general linear model to the data excluding the input voltages above the current saturation. It is also worth noting that we expected the electromagnet current to saturate at 510 mA rather than the 400 mA seen here.

$$I_{EMFit} = D V_{in} + C = 0.1781 V_{in} - 0.0179 \quad \text{if } V_{in} \leq 2.3 V$$

### Electromagnet

$$F(I, x) = \frac{AI^2}{(B + x)^2}$$

Examining our model for the force exerted by the electromagnet we see that we have two unknowns, A and B. We conducted an experiment where we measured the amount of voltage sent to the driver to lift the steel

sphere from a particular height. We then attempted to perform non-linear optimization to determine A and B from this data however this computation diverged due to poor initial guesses for A and B. To determine better initial guesses we symbolically solved for A and B as a function of two of these data points and then used these values as our initial estimate.

At equilibrium we know that the force exerted by the electromagnet must balance out the force of gravity on the sphere, this yields the following two expressions. In our experiment, the moment that the force exerted by the electro magnet exceeds the force of gravity, the sphere will stick to the magnet because the system was uncontrolled. But in that moment we know that the system must have passed through the equilibrium current giving us a good measurement of the correct amount of current.

$$mg = f = \frac{AI_1^2}{(B + x_1)^2} = \frac{AI_2^2}{(B + x_2)^2}$$

We can solve the first expression for A.

$$A = \frac{F(B + x_1)^2}{I_1^2}$$

Then substituting the previous expression for A back into the 2<sup>nd</sup> equation yields the following expression for B:

$$B^2(1 - R) + (2x_2 - 2x_1R) + x_2^2 - x_1^2R = 0 \quad R = \frac{I_2^2}{I_1^2}$$

Choosing two of our x, I pairs allowed us to solve for plausible values of A and B which we used as initial guesses for the optimization allowing it to converge to the correct result. Notice that because our model has both  $I^2$  and  $x^2$  terms, we expect multiple solutions for A and B. The  $I^2$  term physically means that the direction of current through the electromagnet is irrelevant to the force exerted. However the x term is added to our B term before being squared. Physically the B term prevents the force exerted by the electromagnet from going to infinity when x equals 0, but our model has no notion of negative x values, even though it is perfectly legal to plug negative values in for x. Small negative x values correspond to the steel sphere being within the magnet in real life, but in our model does not correctly handle this case. In fact the model suggests that if B is a positive value then for some negative value for x the force exerted goes to infinity. To handle these multiple solutions we attempted the optimization with both pairs of solutions and choose the one that minimized the overall error.

It is important to note the difference between the two different fits presented in figure . Both are the result of non-linear regression to solve for our unknown parameters A and B. However the errors that each minimize is different. The Current fit, minimizes the expected amount of current draw at each of our measured x values from the actual amount of current drawn by the electromagnet. The Force fit minimizes the error between the expected amount of force, computed for each pair of observed current and position, and the actual amount of force required to lift the steel sphere. The fit error described in table 4 refers to the error in each fits respective space. So the force fit has error in units of Newtons while the current fit has error in units of Amps so these two errors are not directly comparable. The required current was computed assuming the specified amount of force needed.

Optimizing the error in force is the more correct answer for this problem because that is the error we wish to minimize. The force exerted is the dependent variable while the current and height are both independent variables. Given a constant force we can link the height to the current, but this is only an approximation and assumes no external disturbances.

## Steel Sphere

The steel sphere has two parameters of consequence for our modeling, its diameter and its mass. Both of these values were obtained through direct measurements using calipers and a scale respectively shown in table 5.

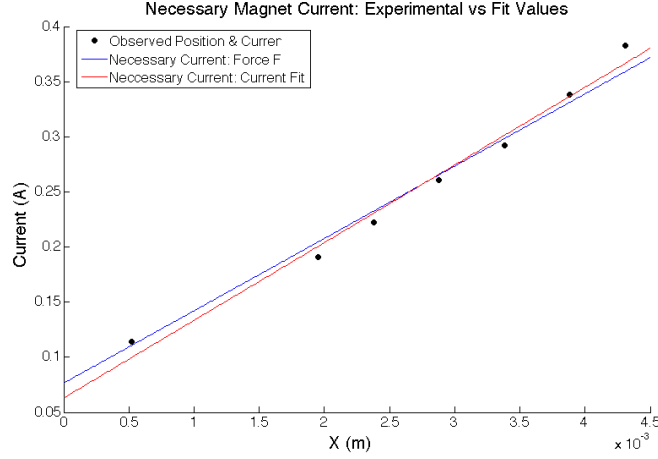


Figure 8: *Electromagnet Current vs X position*

	A	B	Error
Force Fit	2.2795e-05	0.0011656	0.0223076 N
Current Fit	1.9752e-05	8.9294e-04	0.0272997 A

Table 4: *Two different parameter fits to our data*

## Open Loop Transfer Function

Substituting numerical values from our physical system yields the following open loop transfer function:

$$\frac{-78511.7}{s^2 - 6201.6}$$

e.

Figure 9 shows the overall Simulink model of the nonlinear system, and figures 10 - 12 show the Simulink models of the individual sub-systems. The Matlab code for the non-linear model of the Photo-transistor can be seen in the appendix<sup>1</sup>. Figure 13 shows the overall Simulink model of the linear system while figures 14 - 16 show the linearized Simulink models of the individual sub-systems.

f.

Figure 17 shows the time response of the nonlinear and linearized models when both are set to the nominal starting position ( $\bar{x} = 0.00323\text{m}$ ). The time scale has been reduced to better show the behavior of both systems. In both cases the ball oscillates about a distance closer to the electromagnet than the starting

<sup>1</sup>Appendix: Matlab code for Photo-transistor: irradianceEstimator.m

Steel Sphere Mass (g)	10
Steel Sphere Diameter (mm)	12.7

Table 5: *Steel sphere parameters*

MagLev System - NonLinear and Uncontrolled

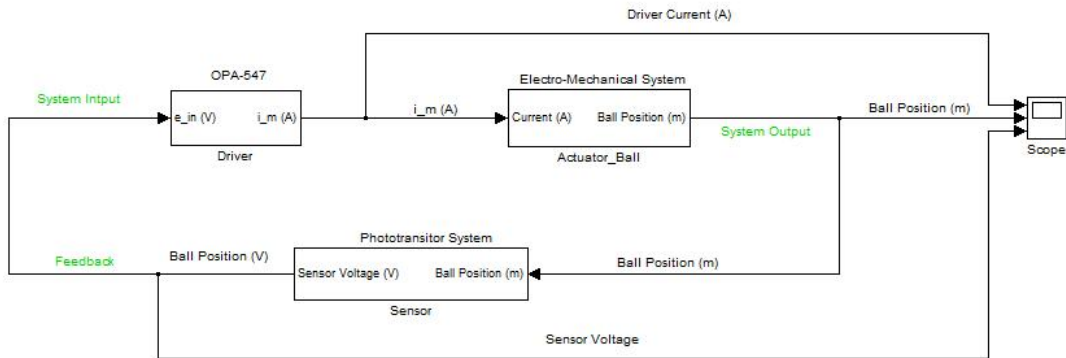


Figure 9: *Nonlinear Simulink model*

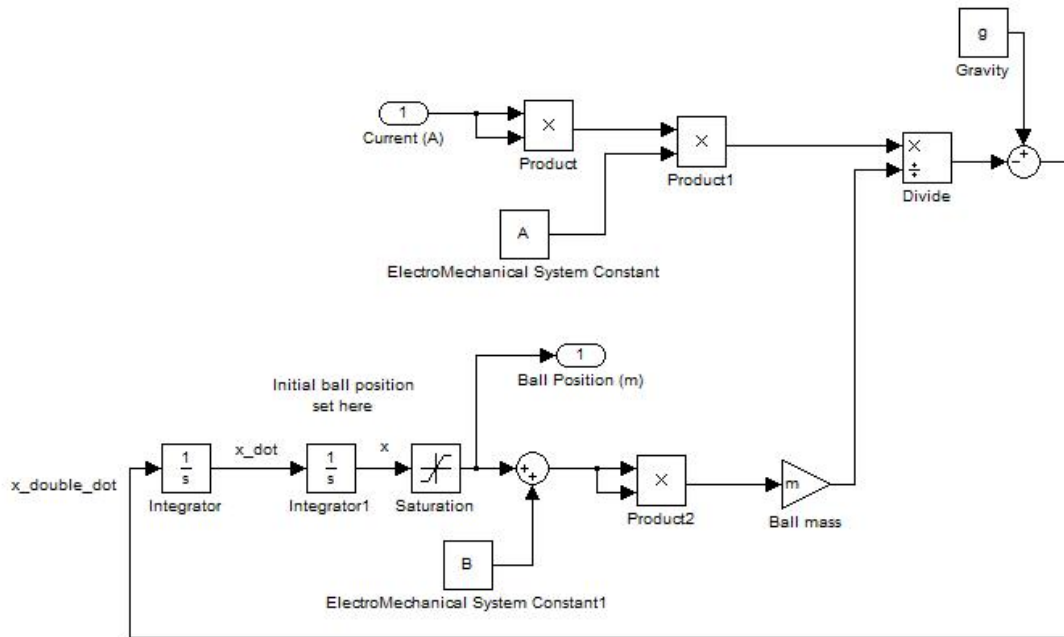


Figure 10: *Non-Linear Simulink model of Actuator-Ball Sub-System*

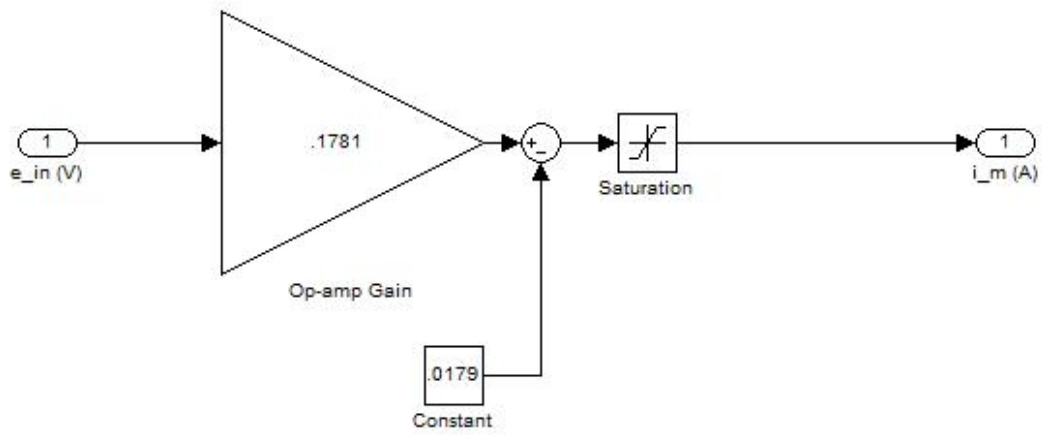


Figure 11: *Non-linear Simulink model of the OPA-547 Op-Amp*

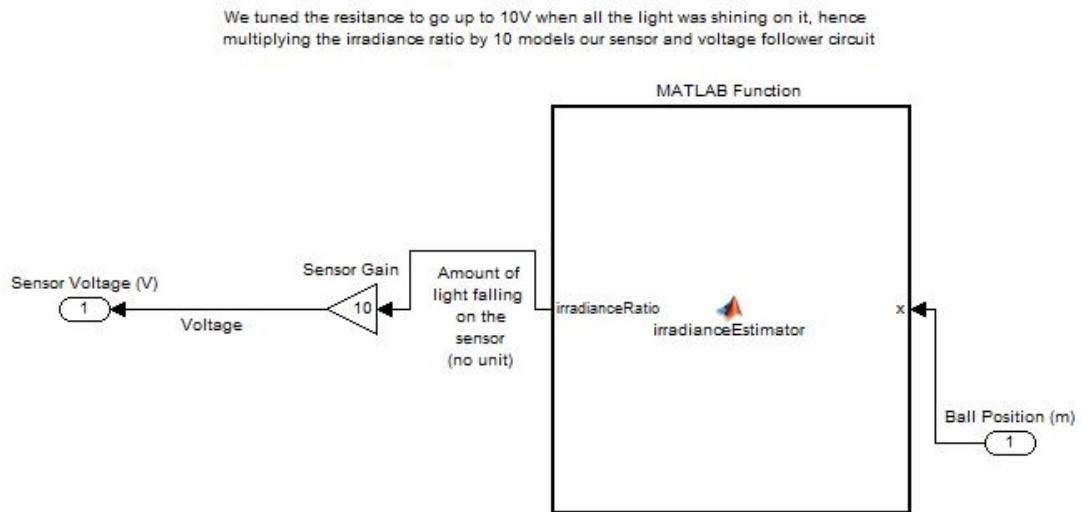


Figure 12: *Nonlinear Simulink model of the Phototransistor*

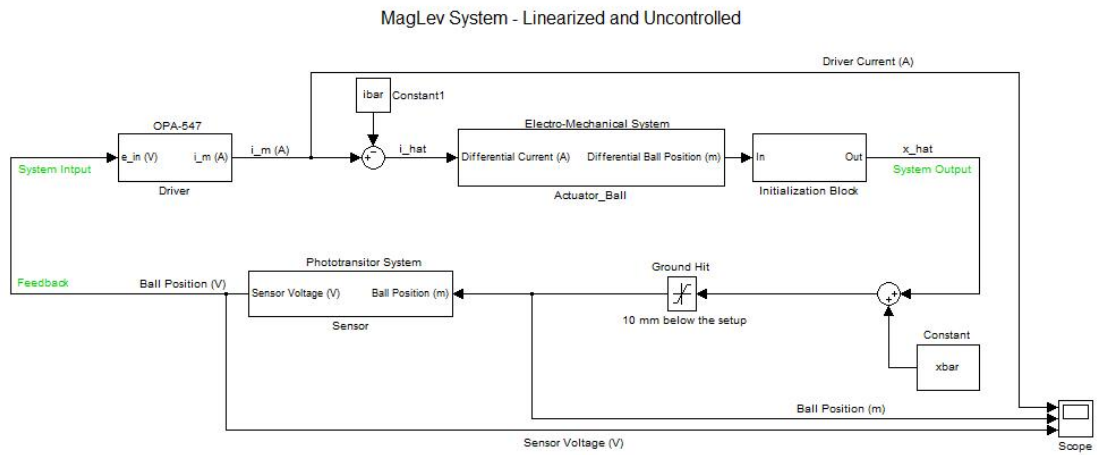


Figure 13: *Linearized Simulink model*

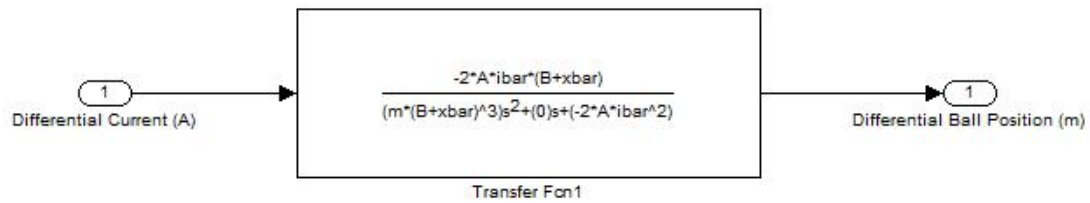


Figure 14: *Simulink model of the linearized Actuator-Ball Sub-System*

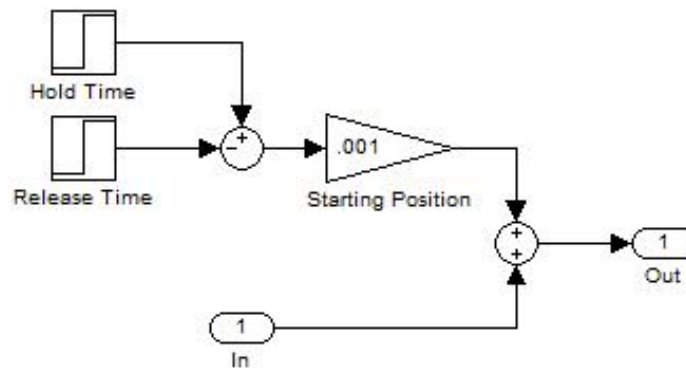


Figure 15: *Simulink Model of Initial Position Offset*



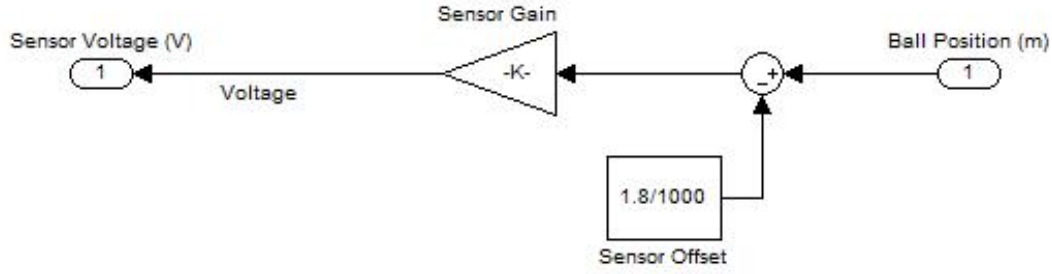


Figure 16: *Linearized Simulink model of the Phototransistor*

point, with the ball in the nonlinear model ending up closer than the ball in the linearized model. In the case of the nonlinear model, the ball strikes the magnet before it begins oscillating, and the amplitude of its oscillations is less than the amplitude of the linearized model.

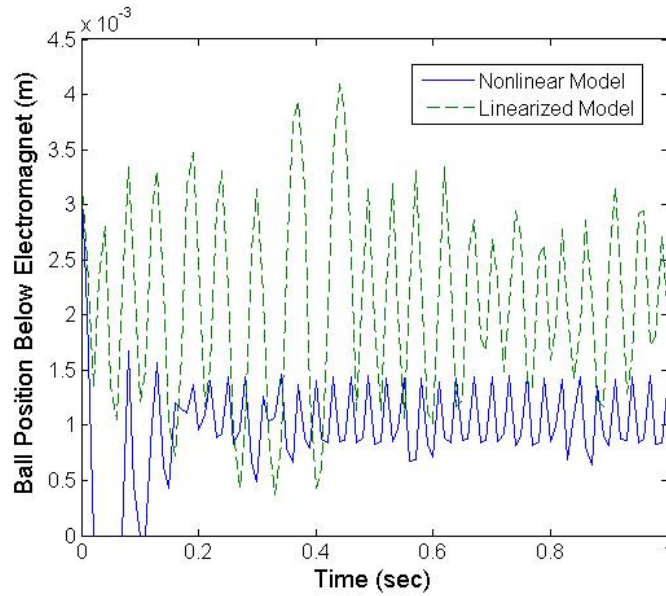


Figure 17: *At the nominal starting height, both systems are able to levitate the ball, but also show sustained oscillations*

In figure 18, the nonlinear and linearized models are both set to a position one mm above the nominal starting position. The time scale has been reduced to better show the behavior of both systems. The linearized model again oscillates about a distance closer to the electromagnet than the starting point, but with smaller amplitude than previously. In the case of the nonlinear model, the ball strikes the magnet, is released, is re-attracted to the magnet, and then falls away.

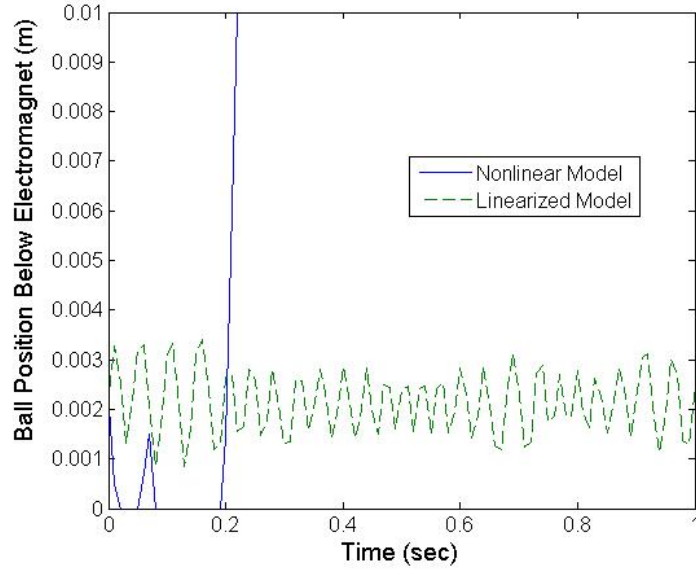


Figure 18: *When starting 1 mm further away from the electromagnet, the linearized model again predicts that it can still levitate the ball (with sustained oscillations) while the non-linear model again forecasts that the ball will fall*

In figure 19, the nonlinear and linearized models are both set to a position one mm below the nominal starting position. The time scale has been reduced to better show the behavior of both systems. The behavior here is very similar to the previous case. The linearized model oscillates about a distance closer to the electromagnet than the starting point, but with greater amplitude than with the closer starting point. The ball in the nonlinear model again strikes the magnet, and is then released and falls away.

When starting at the nominal position, the two models exhibit the same general behavior. However, changing the start point by greater than 0.01mm causes the nonlinear model to become unstable and fall away, while the linear model remains stable. As the model is uncontrolled, we expect the system to be unstable. However, the linearized model does not include the squared current and distance terms, so it is more *optimistic* in terms of keeping the system stable. While the linearization does not agree with the nonlinear model over a large range, the general behavior is similar very close around our set point. Therefore, we believe our linearization assumptions are sufficient for the model, and the addition of a controller will improve the agreement between the models.

## g.

Investigating the fidelity of our models could be done with some simple experiments using the test apparatus and simple physics models. For example, we could input a given set point to the model, but set the current to the driver block at a constant zero. The resulting output of position vs. time should match that of general falling body physics equations. Similar experiments could be done by inputting a constant voltage command to the real and simulated drivers and comparing the Simulink output to the position measurements within LabView. These simple tests would give us some indication of if our models are accurately representing the true motion of the physical system.

More specific tests could be conducted on the individual sub-systems. For example, using the stand and shims we could place the ball at known heights and record the sensor output. We could then set the model to the same heights and record the output of our sensor models. The same type of test could be conducted

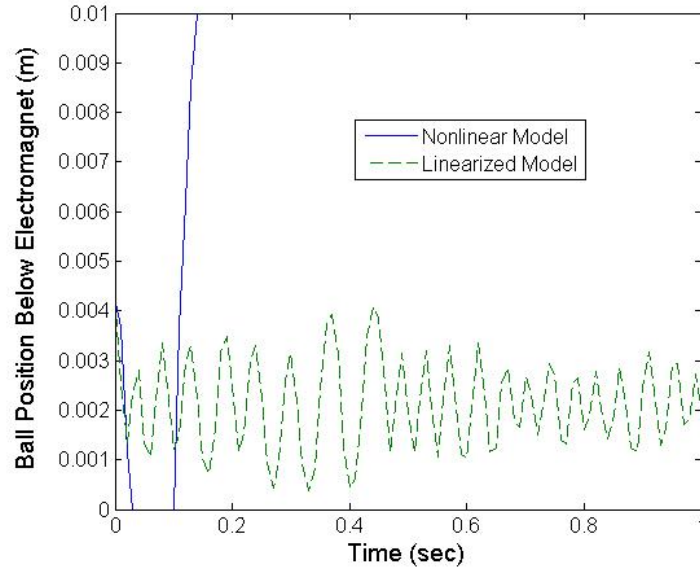


Figure 19: When starting 1 mm closer to the electromagnet, the linearized model predicts that it can still levitate the ball (with sustained oscillations) while the non-linear model forecasts that the ball will fall

on the driver.

## Question 2

Using the linearized model from question one, the root locus plot in figure 21 shows that the physical system will always be unstable in open-loop, because there will always be a pole in the right-half plane (RHP).

A lead controller adds a pole and a zero to the system, with the pole being greater in magnitude than the zero (i.e. further from the imaginary axis) and both in the left-half plane (LHP). Because there were no design requirements given (settling time, overshoot, etc.), specific values were not as critical as they may have been otherwise, and we chose our controller pole to be at -80 and the controller zero to be at -20. Additionally, to ensure the proper sign throughout the system, a factor of -1 was applied to the controller. The root locus plot below shows the result of applying the controller to the system. We see that the system will be stable if the gain is large enough, because all of the closed-loop poles will be within the LHP. The gain magnitude must be greater than 0.096 for the rightmost pole to cross into the LHP, and greater gains reduce the settling time and steady-state error. But as the gain increases, the imaginary poles move further away from the real axis (lower damping). Therefore, we selected a gain of 0.37 as a compromise value. Our lead compensator is given by the equation below, and its effect on the physical system is shown in the following plot (figure 21).

$$C(s) = -0.37 \frac{(s + 20)}{(s + 80)}$$

**a.**

From the root locus plot and Matlab sisotool analysis, the system must have a gain greater than 0.096 in magnitude (-20.35 dB). As mentioned previously, this gain must be negated to ensure the proper sign of the

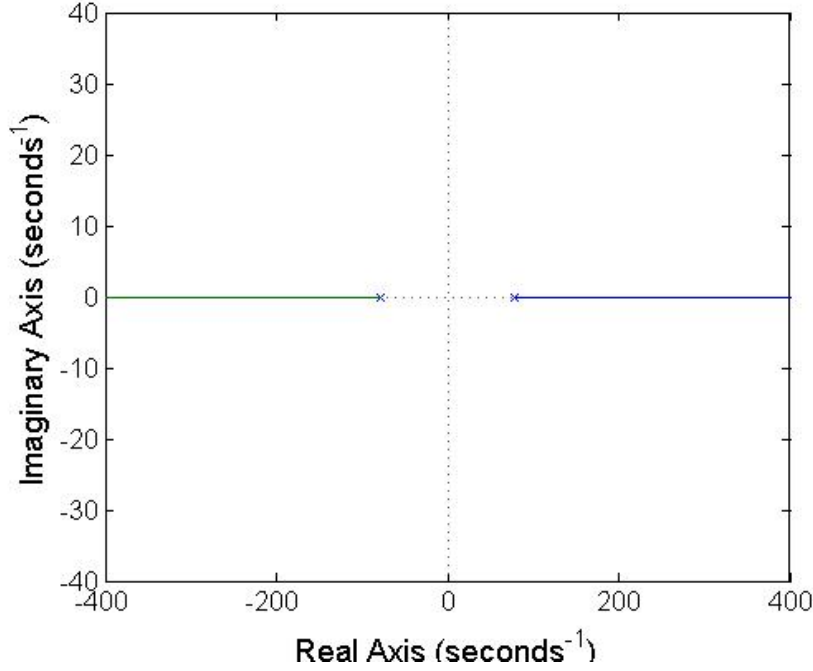


Figure 20: *Open-Loop Physical System is Always Unstable.*

signal in the system. With the controller,  $C(s)$ , the closed-loop transfer function is given by:

$$\frac{2.367 * 10^4 s + 4.732 * 10^5}{s^3 + 79.995s^2 + 1.746 * 10^4 s - 2.290 * 10^4}$$

From the closed-loop bode plot (figure 22), the bandwidth (frequency at which the magnitude is -3 dB) is expected to be 468.22 rad/sec (74.52 Hz).

**b.**

Figures 23 and 24 show the Simulink models of the nonlinear and linearized models, respectively, with the controller applied. Figure 25 shows the Simulink model of the controller itself, which is the same for the linearized and nonlinear models. All other sub-systems in the model are the same as shown in Part 1 (e).

Figure 26 shows the time response of both models when the ball is started at the nominal height ( $\bar{x} = .00323m$ ). Unlike the uncontrolled systems which oscillated continuously, when the controller is applied both systems settle on a steady position levitated below the electromagnet (the nonlinear model has some initial oscillations before settling).

If the starting position of the ball is moved closer to the magnet, both systems remain stable until a position 0.8mm above the set point, when the linearized model becomes unstable. An example of this is shown in figure 27. In contrast, when the starting position is moved away from the magnet both systems remain stable until a position 1.9mm below the set point, when the nonlinear model becomes unstable. An example of this is shown in figure 28.

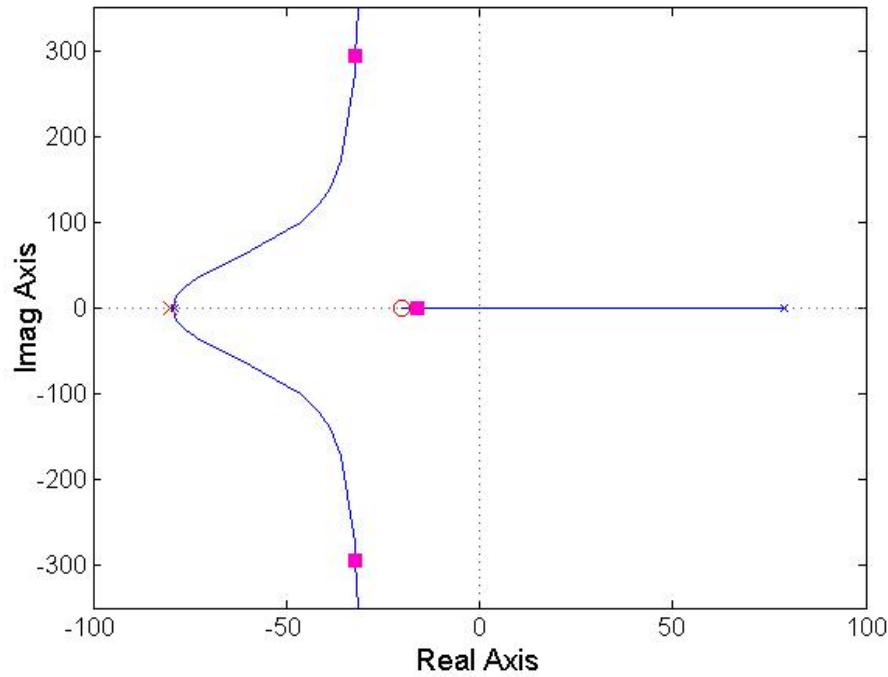


Figure 21: *With Sufficient Gain, Lead Controller Stabilizes the System.*

Within the region where both models are stable, their overall behavior is similar, with both undergoing some oscillations before settling at different steady state points (the steady-state values are within 0.1mm of each other). The main difference between the models is that the linear model has a longer settling time, regardless of whether the starting point is above or below the set point. Another major difference is that when the start point is moved closer to the electromagnet, the linear model becomes unstable first, on the other hand, it is the nonlinear model that becomes unstable first when the start point is moved further from the electromagnet.

### c.

The controller above does match the controller we used in the lab exercise during the previous week. Initially we had designed a new controller for this exercise, with different pole and zero locations, and a different gain value. However, both the nonlinear and linearized Simulink models were unstable under all of the conditions and variations we attempted. After discarding the new values and using the same poles, zeros, and gains that we had implemented in LabView on the test apparatus, the models became stable. This may be because prior to designing the controller for LabView, we conducted numerous tests to identify the parameters of the real system and modeled the various sub-systems.

### d.

The block diagram in figure 29 illustrates how we ensure we have negative feedback in our system. If we start at the far right and assume that the ball just moved slightly above the set point (closer to the electromagnet), the sensor outputs a voltage related to this position which will always be positive. This voltage is then converted to position (m) and subtracted from the set point. Because the ball is above the set point, the error will be positive. The error signal then passes through a gain block of -1 to flip its

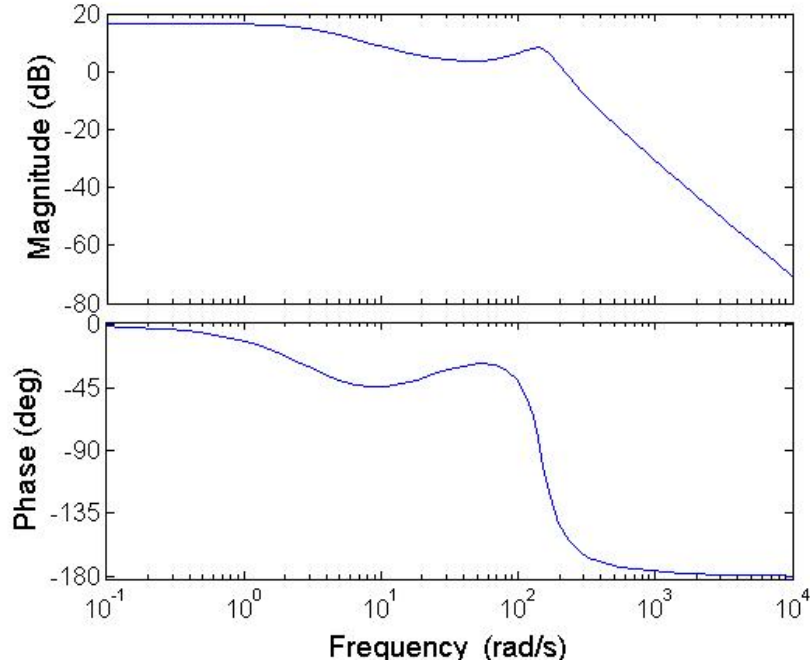


Figure 22: *Bode Plot of the Closed-Loop System.*

sign to negative. The controller takes in the negative error and outputs a negative voltage. By adding this negative voltage to the bias voltage, a lower command voltage is sent to the driver than previously (the starting assumption was that the ball had been at the setpoint, but just moved up). This results in a smaller current being sent to the actuator, which results in less force from the electromagnet. Less force from the electromagnet results in the ball falling further away from the magnet, back towards the set point. This is the behavior that we want.

Similarly, if the ball is below the setpoint, the error will be negative, and the inverting unity gain block will send a positive error into the controller, which will then output a positive voltage. Adding this voltage to the bias voltage results in a larger command voltage than when the ball is at the set point, so the current to the actuator will increase. The increased current will result in a larger force from the electromagnet, which will pull the ball back up towards the set point. Again, this is the behavior we want, so the signs within our system are correct. Because the physical and electronic sub-systems have restrictions on their signs (e.g. the sensor is tuned to always output between 0 and 10 volts), inserting the negation before the controller is the most important step to ensure we have stabilizing feedback.

## Question 3

### Predicting the Range of Controllability

In our system, we use reference position commands instead of voltage commands because it makes semantic sense. For our physical system, the original set point was 3.23mm, and we found the controllable position command ranged from 3.03mm to 3.53mm. For the linear model, the control range goes from 2.53mm to 5.23mm. The non-linear model is controllable from 0mm to 5.13mm.

We see overlapping agreement between the virtual models and the physical system around the set point.

The physical system has the least range of controllability. The difference between our models and the real system arise due to a range of factors that we did not take into account such as op-amp short-comings, air viscosity, heating in the electromagnet's coil causing a change in properties, etc. We additionally see that our non-linear model has higher fidelity in predicting what the system would do compared to the linear model, but both grossly optimistically over-estimate the range of controllability. Thus we can conclude that we could *not* have gauged the controllability of the electromagnet system using the two models.

The frequencies used in the empirical observations started from 0.01Hz and went all the up to 13Hz. A total of 19 data points were sampled. More samples were selected at higher frequencies since the behavior of the system was more interesting behavior was predicted by the linear model. However, the physical system became unstable at higher frequencies and we were limited by this. The bode plot for these data is shown in figure 30. Compared to the one generated from our linear theoretical model, we see that the predicted magnitude change is similar to the actual change in magnitude: as the frequency of input signals increases, they start steadily, decrease and then increases a little bit. However, when we look at the phase, although the trends for both diagrams are similar, they have some noticeable difference. This is probably due to non-linearity in the system and some factors that we have not modelled. This can also account for why our observed bandwidth and predicted bandwidth vary.

## Appendix

### Matlab code: radianceDerivative.m

```
function [dRdd] = radianceDerivative(rbeam, rsphere, d)
% compute derivative of the radiance curve at a particular distance

assert((rbeam >= 0) && (rsphere >= 0), 'Expected non-negative numbers')

if ((rsphere + abs(d)) <= rbeam)
    % handle the block is completely in the beam and the beam is bigger
    dRdd = 0;

elseif ((rbeam + abs(d)) <= rsphere)
    % handle case where block is larger or same size than beam and completely in the way
    dRdd = 0;

elseif (abs(d) < (rbeam + rsphere))
    Abeam = pi * rbeam * rbeam;

    % y stuff
    y = (d*d + rbeam*rbeam - rsphere*rsphere) / (2 * d);
    dydd = 1/2 - (rbeam*rbeam)/(2*d*d) + (rsphere * rsphere) / (2*d*d)

    % intersection distances
    dbeam = y;
    ddbeam = dydd
    dsphere = d - y;
    ddsphere = 1 - dydd

    % occluded areas
    A0beam = cSarea(rbeam,dbeam)
    dA0beam = cSDarea(rbeam, dbeam, ddbeam)
    A0sphere = cSarea(rsphere,dsphere)
    dA0sphere = cSDarea(rsphere, dsphere, ddsphere)

    % total overlap
    AOverlap = A0beam + A0sphere;
    dAOverlap = dA0beam + dA0sphere;

    % change in radiance fraction
    dRdd = dAOverlap / Abeam;
else
    % handle if they aren't even touching
    dRdd = 0;
end

end

function [area] = cSarea(r, h)
% computes the area of the circular segment where the upper boundary is a
% circular arc and the lower boundary is a cord of the circle with radius r
% h is the perpendicular distance from the center to the cord
```



```

area = r * r * acos(h / r) - h * sqrt(r * r - h * h);

end

function [dA] = cSDarea(r, h, dh)
dA = (- (r / sqrt(1 - (h/r)^2)) - sqrt(r*r - h*h) + ((h*h) / sqrt(r*r - h*h)))*dh;
end

```

## Matlab code for Photo-transistor: irradianceEstimator.m

```

function irradianceRatio = irradianceEstimator( x)
% computes the fraction of irradiance as a function of the radii of the two
% intersecting circles and the distance between their centers, d
% This can be used to model the irradiance of our phototransistor if we
% make several assumptions:
% - ignore diffraction
% - assume parallel light rays from LED to photo transistor so that we can
% model the * note we should use the minimum radius of the LED and the
% phototransistor - since - the parallel ray assumption means that distance
% from the ball to the LED doesn't matter

% let r1 be the beam radius and r2 be the blocking radius

% details at http://mathworld.wolfram.com/Circle-CircleIntersection.html

% estimated rblock = 6.35 mm
% estimated rbeam as = 2.3650 mm radius of the phototransistor

% regressed rbeam = 1.5048

%assert((rbeam >= 0) && (rblock >= 0), 'The radii should be non-negative');

r1 = (1.5048/1000);
r2 = (6.35/1000);

d = x + 4.12/1000;      % - (2.23/1000) + 12.7/2000;

if ((r1 > r2) && ((r2 + abs(d)) < r1))
    % handle the block is completely in the beam and the beam is bigger
    A1 = pi * r1 * r1;
    A2 = pi * r2 * r2;

    irradianceRatio = (A1 - A2) / A1;

elseif ((r2 >= r1) && ((r1 + abs(d)) <= r2))
    % handle case where block is larger or same size than beam and completely in the way
    irradianceRatio = 0;

elseif (abs(d) < (r1 + r2))
    % handle case where two circles intersect each other
    x = (d * d + r1 * r1 - r2 * r2) / (2 * d);

```

```

d1 = x;
d2 = (d * d - r1 * r1 + r2 * r2) / (2 * d);

A = cSarea(r1, d1) + cSarea(r2, d2);

Amax = pi * r1 * r1;

irradianceRatio = (Amax - A) / Amax;

else
    % handle case with no overlap
    irradianceRatio = 1;
end

end

function [area] = cSarea(r, h)
% computes the area of the circular segment where the upper boundary is a
% circular arc and the lower boundary is a cord of the circle with radius r
% h is the perpendicular distance from the center to the cord

area = r * r * acos(h / r) - h * sqrt(r * r - h * h);

end

```

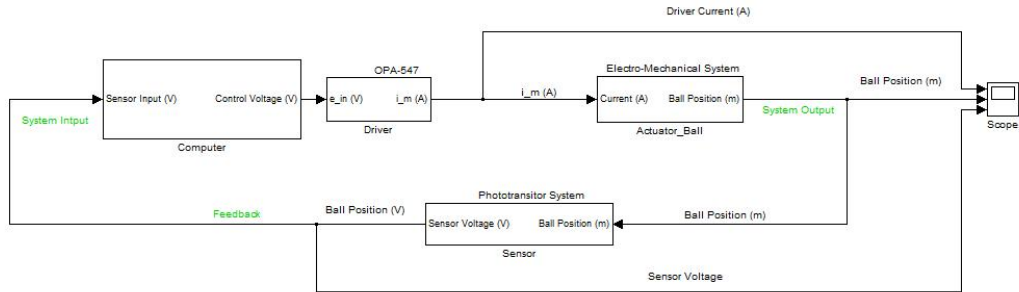


Figure 23: *Nonlinear Model with Controller*

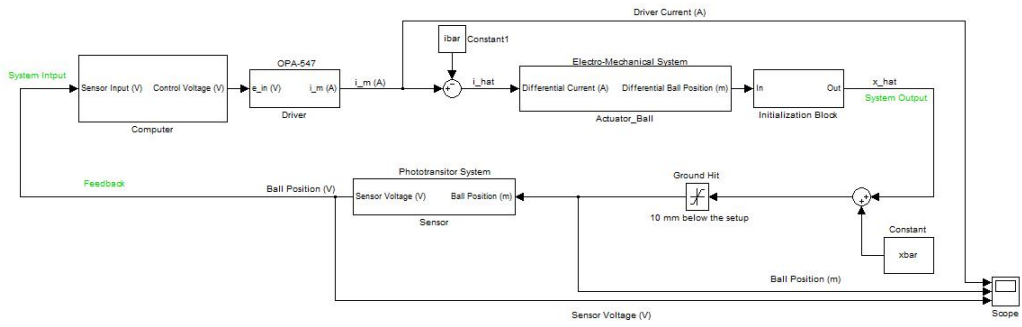


Figure 24: *Linear Model with Controller*

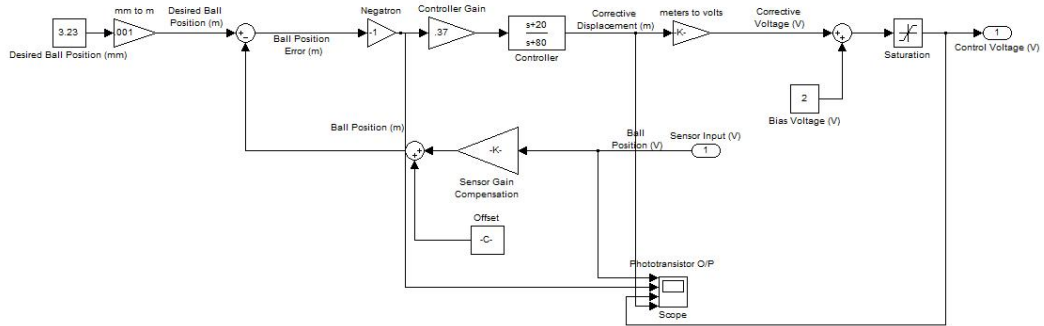


Figure 25: *Simulink Model of the Controller sub-system*

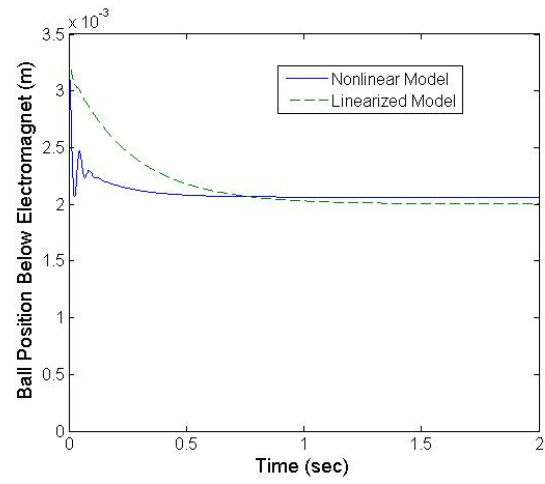


Figure 26: *With the controller, both models have steady levitation positions when starting at the nominal distance.*

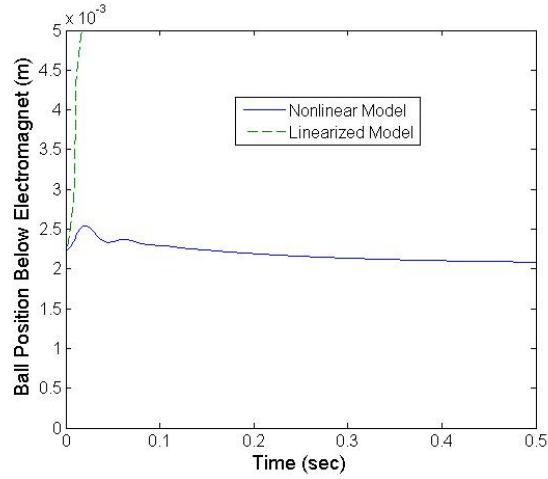


Figure 27: When starting 0.8mm above the set point, the linearized model is unstable and the ball falls.

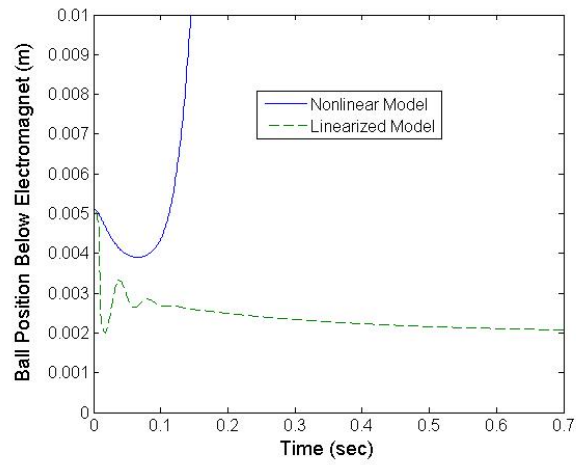


Figure 28: When starting 1.9mm below the set point, the nonlinear model is unstable and the ball falls.

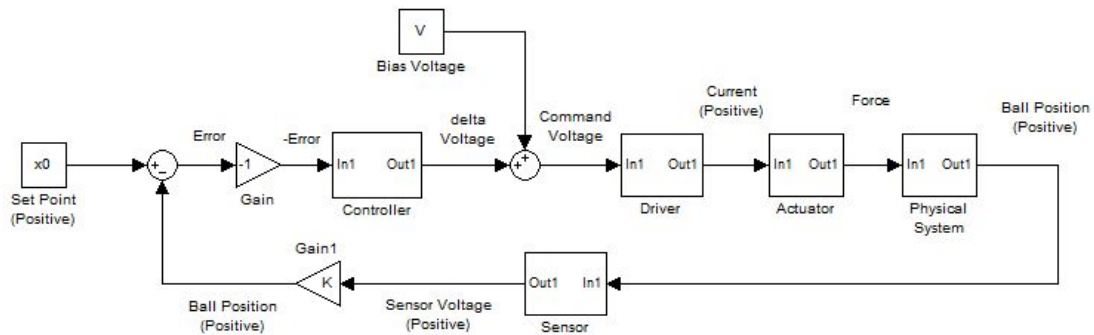


Figure 29: Block Diagram of System: Signs Ensure Negative Feedback.

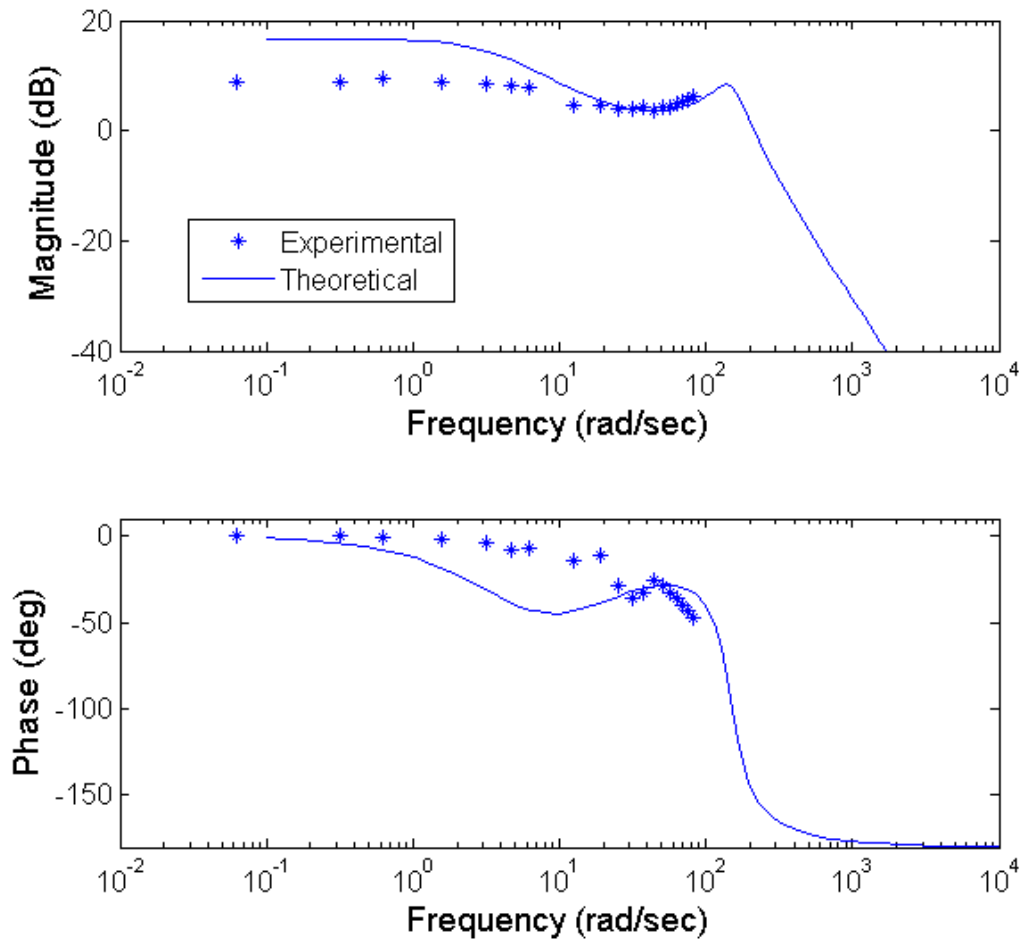


Figure 30: *Bode plot comparison of predicted and observed values.*