UNIVERSITY OF WASHINGTON
Department of Aeronautics and Astronautics

Feedforward Control of the Magnetic Ball Levitator

June 4, 2004

Christopher Lum

Table of Contents

Introduction

This paper covers the effects of model based inverse inputs to the magnetic ball levitation system. We will investigate the trajectories obtained using both feedback and feedforward and feedback alone in both simulation and with actual hardware. The calculations of these inverse inputs and their restrictions are also addressed.

Acknowledgements

The author would like to take this time to thank Professor Santosh Devasia of the University of Washington for guidance offered during the course of this project.

Nomenclature

Table 1: Nomenclature and symbols use in this report

Symbol	Value	Description
a	1000	Pseudo-derivative pole
a_1	0.001232	Numerator coefficient for $F_{mag}(x_1,u)$ (see Equation 3)
a_2	0.000179	Numerator coefficient for $F_{mag}(x_1,u)$ (see Equation 3)
α		$\partial F_{mag}(x_1,u)/\partial x_1$ *evaluated at* $\overline{x}=\overline{x}_e, u=u_e$
β		$\partial F_{mag}(x_1,u)/\partial u$ *evaluated at* $\overline{x}=\overline{x}_e, u=u_e$
$C(s)$		Compensator used to stabilize system
$\Delta(s)$		Difference between $G_o(s)$ and $G(s)$
$F_{mag}(x_1,u)$		Magnet force (see Equation 3)
$G_o(s)$		Mathematical system model linearized about 12mm
$G(s)$		Actual system for small perturbations about 12mm
g	9.81 m/s^2	Gravitational constant (m/s^2)
γ	1.092	Denominator parameter for $F_{mag}(x_1,u)$ (see Equation 3)
K_A	3.6	Amplifier gain (volt/volt)
K_P	-5000	Proportional feedback gain
K_I	0	Integral feedback gain
K_D	-500	Derivative feedback gain
$L_n m$		Lie derivative of n in the m direction $\left(L_n m=\left(\partial m/\partial\overline{x}\right)*n\right)$
m	0.0165 kg	Mass of 5/8" diameter steel ball (kg)
n	2	Number of states of system
r	2	Relative degree of system
u		Control voltage to amplifier
u_e		Equilibrium control
v_{sensor}		Voltage output from sensor (volts)
$\overline{x}_e$		Equilibrium state
x_1		Ball distance from tip of magnet (m)
$\overline{x}$		State vector (position and velocity are states)
y		Output of system x_1
y_d		Desired output of system
$\Vert\ \Vert$		Matrix 2-norm
$(\ )^{(r)}$		This denotes the r^{th} derivative of ( ) w.r.t. time

Experimental Apparatus

The general experiment is shown below in Figure 1.

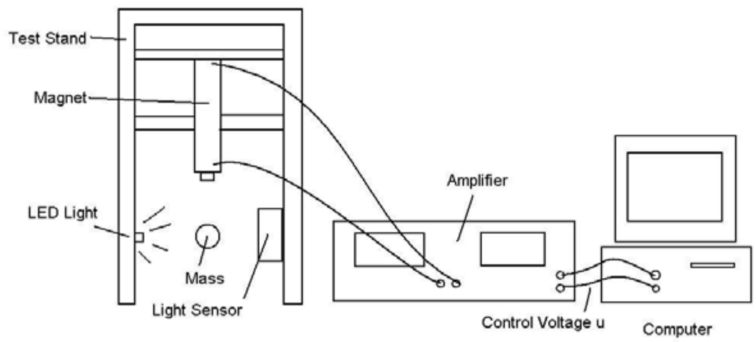

Figure 1: Experimental Apparatus

The Magnetic Ball Levitator (MBL) system consists of a large electromagnet that can be used to levitate a steel ball a certain distance away from its tip. The ball position is obtained using a light emitting diode (LED) and Photo-sensor combination. The LED emits a constant amount of light. As the ball moves up or down, it blocks more or less light to the photo-sensor. The sensor then outputs a voltage that is proportional to the amount of light it receives. The magnet is powered by a amplifier with a constant gain of 3.6 volts/volt and the entire system is controlled in real time using a program called xPC Target which is an additional Simulink toolbox for real time control.

The actual magnet and the LED close up is shown below in Figure 2.

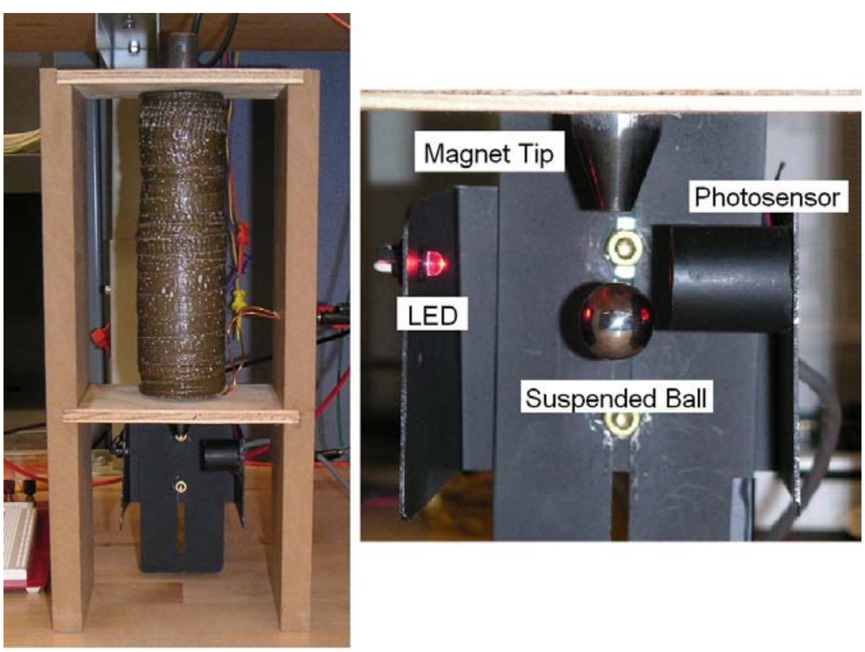

Figure 2: LED and Photo-sensor

The magnetic force needs to be calibrated as a function of the applied voltage and the distance from the tip. The magnetic force is calibrated using a force transducer made by Transducer Techniques Load Cells, S/N 62551 with a capacity of +/-10 lbs. The calibration is performed with the magnet upside down and the transducer measures the magnetic force exerted on the ball (after subtracting the weight of the ball). A set voltage is applied to the magnet and the force on the ball is measured at distances varying from 1mm to 20mm in 1mm increments. This is then repeated for a different voltage. The calibration setup is shown below in Figure 3.

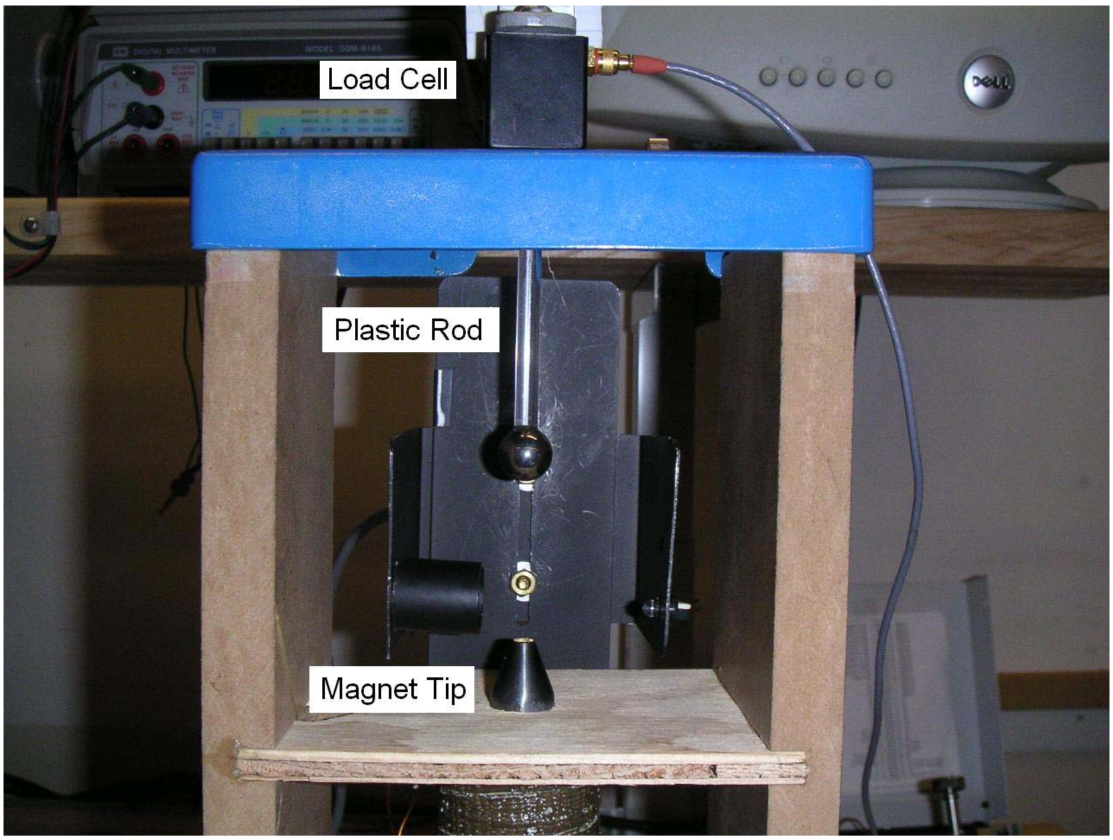

Figure 3: Calibration of magnetic force (apparatus is upside down)

The sensor is also calibrated with the setup shown in Figure 3 except the plastic rod is replaced with a piece of string to minimize the blockage of light to the sensor. The voltage output of the sensor is then calibrated vs. x_1 (ball position) as shown below in Figure 4.

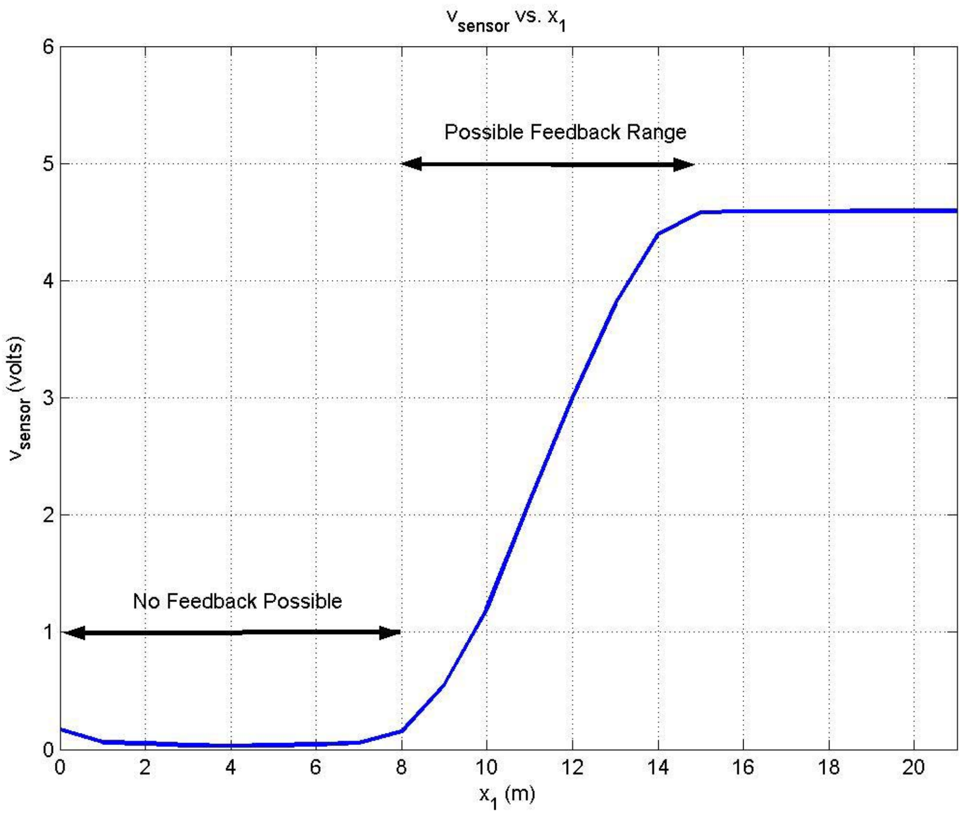

Figure 4: Sensor output voltage vs. ball position

This shows the need for feedforward control. As can be seen, the sensor is only effective at certain x_1 values. If the ball is to close or too far away from the tip, no feedback is possible because a change in ball position does not yield a change in output from the sensor. Therefore, in these regions, we are forced to rely on feedforward control inputs alone to position to the ball into a region where feedback can be applied.

System Modeling

The system equations of motion can be modeled using the reference frame shown below in Figure 5.

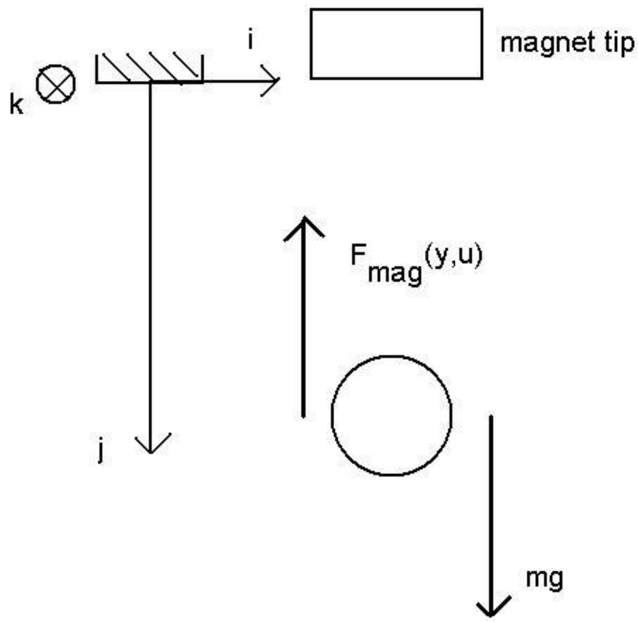

Figure 5: Reference frame for coordinates

The degree of the model can be chosen here. Obviously, a second order differential equation can be used to describe the mechanical side of the system and is governed by Newton's 2^{nd} law. However, the magnet can be thought of as an inductor and resistor combination. Since the magnet force is proportional to the current and we only have direct control over the voltage, this means that there should be another 1^{st} order differential equation that governs the electrical side of the system and describes the dynamics between the voltage and current. If we neglect the current dynamics, the relative degree of the system is 2 and there is no internal dynamics. However, if the current dynamics are modeled, the system still has a relative degree of 3 and still has no internal dynamics. Furthermore, the current dynamics are much faster than the mechanical dynamics and therefore, can be safely neglected. Therefore, since assuming we have direct control over the current does not change the dynamics greatly and does not change the model structure (ie $r = n$), we will choose the state vector to be

$$\bar{x} = \begin{pmatrix} y & y^{(1)} \end{pmatrix}^T$$

Equation 1

With this state vector, the system equation of motion is given by

$$m x_1^{(2)} = -F_{mag}\left(x_1, u\right) + mg$$

Equation 2

One of the main challenges of this project is modeling the non-linear $F_{mag}\left(x_1, u\right)$ term. The calibrated data for $F_{mag}\left(x_1, u\right)$ as a function of x_1 and u is presented in Appendix A and is shown graphically in Figure 6.

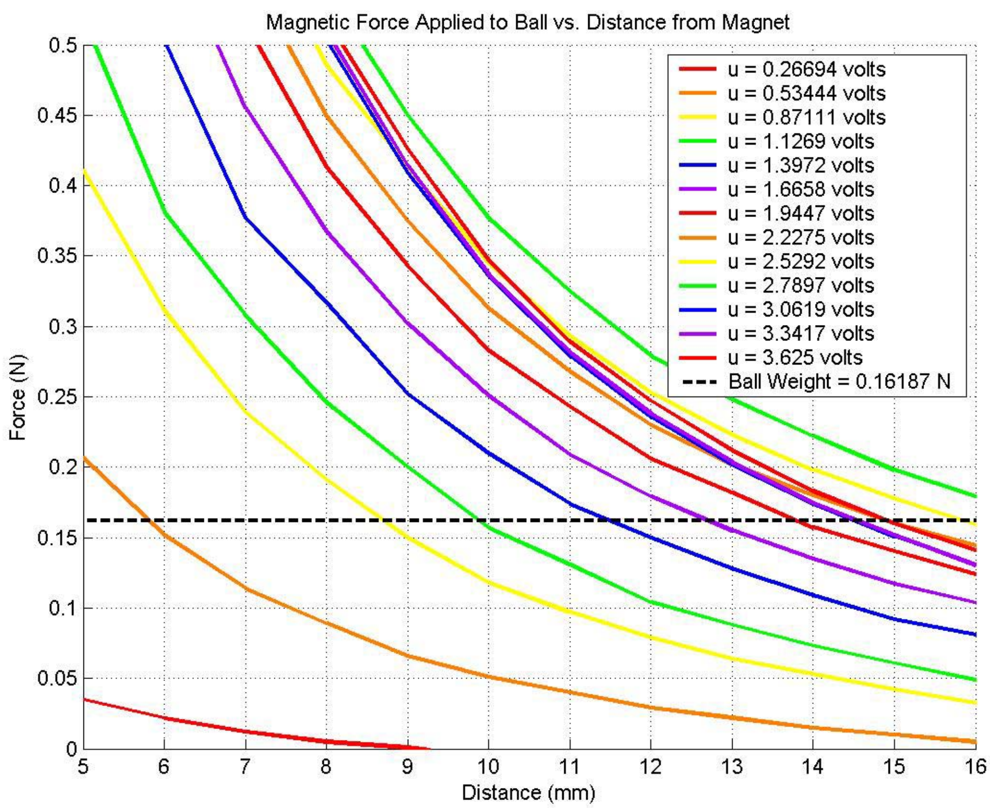

Figure 6: $F_{mag}(x_1,u)$

Obviously, this is a very nonlinear function. We choose to fit it with the following function

$$F_{mag}\left(x_1, u\right) = \frac{a_1 u + a_2}{x_1^{\gamma}}$$

Equation 3

We fit the data shown in Figure 6 to obtain the best-fit coefficients for a_1, a_2, and γ, these values are shown in Table 1. The fitted function and the raw data are shown below in Figure 7.

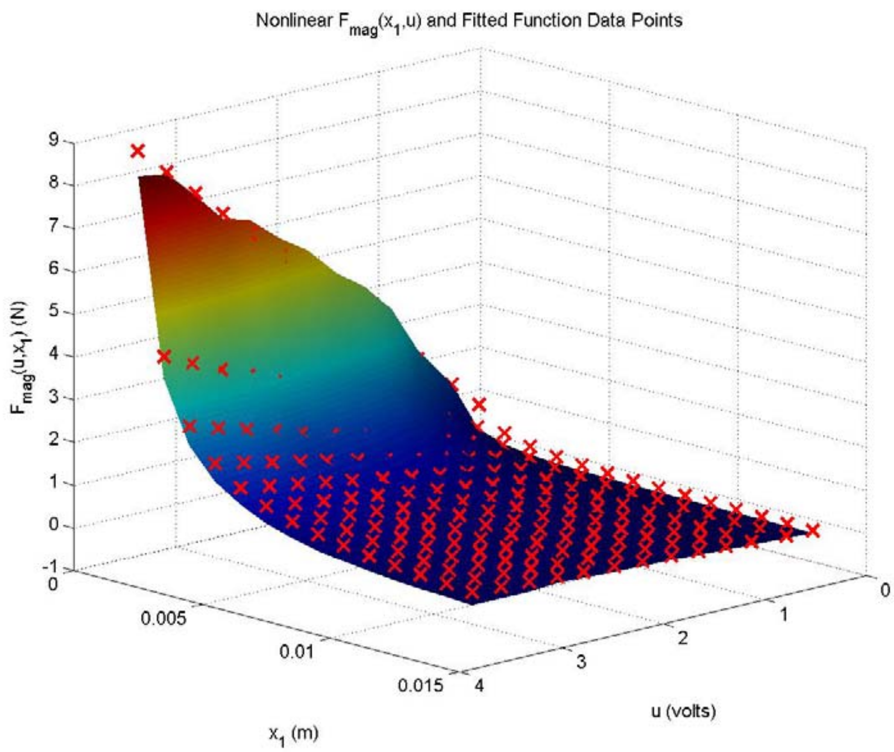

Figure 7: Fitted $F_{mag}(x_1,u)$ and actual data

The experimental data is the underlying surface and the predicted values of F_{mag} are the red x's. As can be seen, the model of the magnetic force matches the actual data fairly well.

The nonlinear equations of motion for this system can now be written as

$$\dot{x} = f(\bar{x}) + g(\bar{x})u$$

Equation 4

where $$f(\bar{x}) = \begin{pmatrix} x_2 \\ -\frac{a_2}{m x_1^{\gamma}} + g \end{pmatrix}$$

$$g(\bar{x}) = \begin{pmatrix} 0 \\ -\frac{a_1}{m x_1^{\gamma}} \end{pmatrix}$$

The output of the system is the position of the ball and therefore

$$y = h(\bar{x}) = x_1$$

Equation 5

The relative degree of this system is determined to be 2. Therefore, since the relative degree is equal to the number of states, then the system has no internal dynamics and the inverse input of the system is given by

$$u_{ff} = \left(L_g L_f\right)^{-1}\left(y_d^{(2)} - L_f^{\,2} h\right)$$

Equation 6

where $L_g L_f = -\dfrac{a_1}{m y_d^{\gamma}}$

$$L_f^{\,2} h = -\frac{a_2}{m y_d^{\gamma}} + g$$

This system can now be linearized about an arbitrary equilibrium point using the standard Jacobian operation. This yields the following structure of the linearized model

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ -\dfrac{\alpha}{m} & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ -\dfrac{\beta}{m} \end{pmatrix} u$$

Equation 7

Notice that α is a negative number and therefore, the linearized system is unstable. Furthermore, the B matrix is negative so if we use negative feedback, the gains of the controller will be negative. Since there are only two states, a PD controller is used and this is actually a full state controller. We choose a compensator of the form

$$C(s) = K_p + K_d \frac{as}{s+a}$$

Equation 8

The root locus of the stabilized system with the system linearized about $x_1 = 12mm$ with $K_p = -5000$, $K_D = -500$, and $a = 1000$ is shown below in Figure 8.

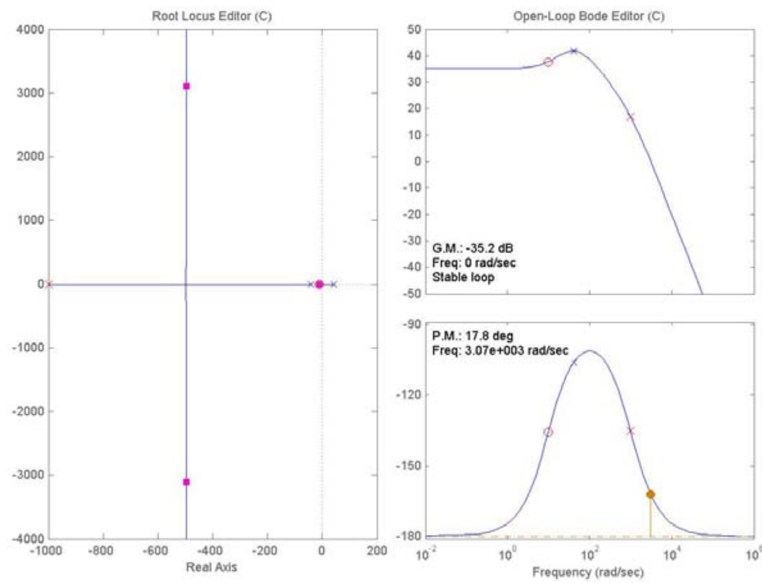

Figure 8: Root locus of stabilized system

As can be seen, this stabilizes the system and we can now use this compensator and feedforward input as shown below in Figure 9. Although this does not appear to be an optimal location of the roots, we choose these P and D gains such that the actual experimental system is stable. Note that this has an integrator in the controller that is currently set to zero for the analysis in this paper. This could be set to a non-zero value if zero steady state error is desired.

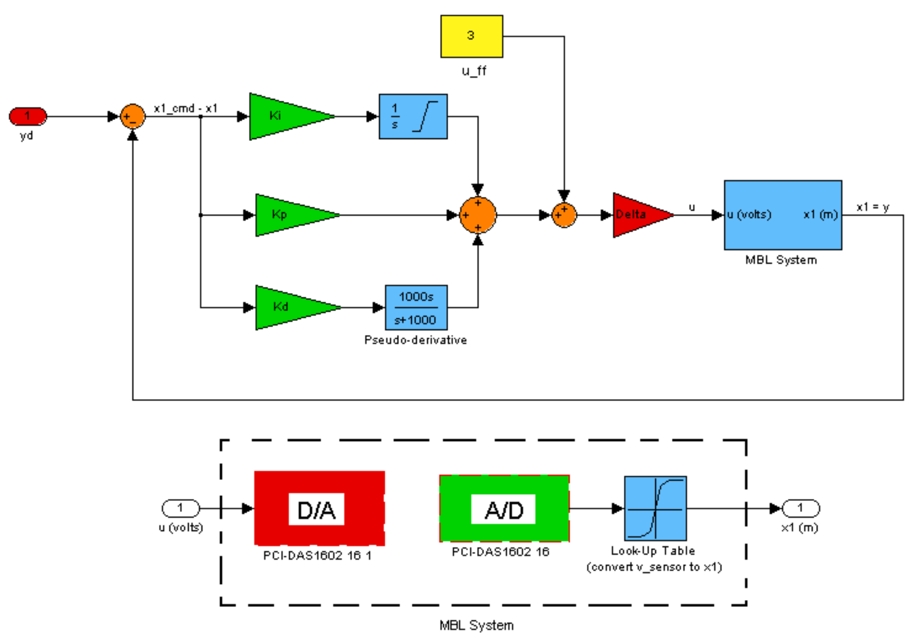

Figure 9: Controller and overall system used to interface with hardware

Model Validation

We can now simulate the response of the system to different inputs. First, to validate the accuracy of the simulation, we can simulate the run the actual system when the commanded input is a series of steps using only feedback and no feedforward input. The trajectories obtained are shown in Figure 10.

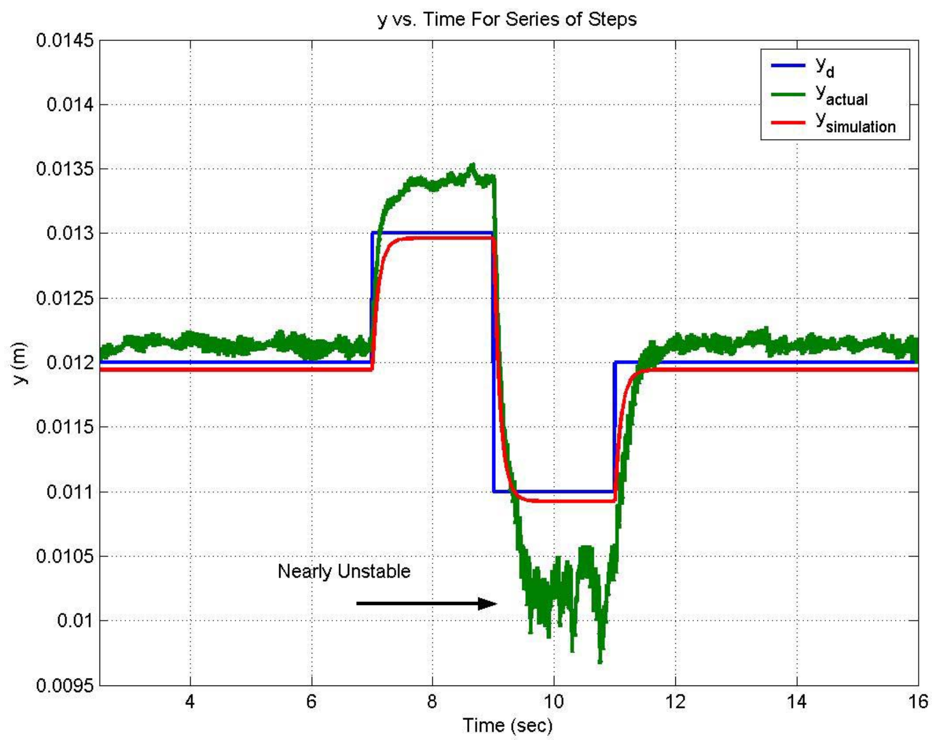

Figure 10: Actual and experimental data for series of steps

As can be seen, the system is obviously stable and able to track these steps to some degree. Notice that the actual system becomes nearly unstable when the ball moves too close to the magnet. This is because the actual system is highly non-linear and this data was obtained using a static K_p and K_d gain. The actual system will require the use of gain scheduling if a larger range of motion is desired. The range of +/-1.5mm from the nominal x_1=12mm is about the most that the static gains can handle before becoming unstable. Although we are using the non-linear simulation, we do not see this phenomenon in the simulation.

The model does a fair job of approximating the actual system in the transient phase but seems to have a discrepancy in steady state. To investigate this steady state error more, we can simulate the and run the actual system when the desired trajectory is to simply hold to ball at a constant 12 mm using both feedforward and feedback control. This is shown in Figure 11.

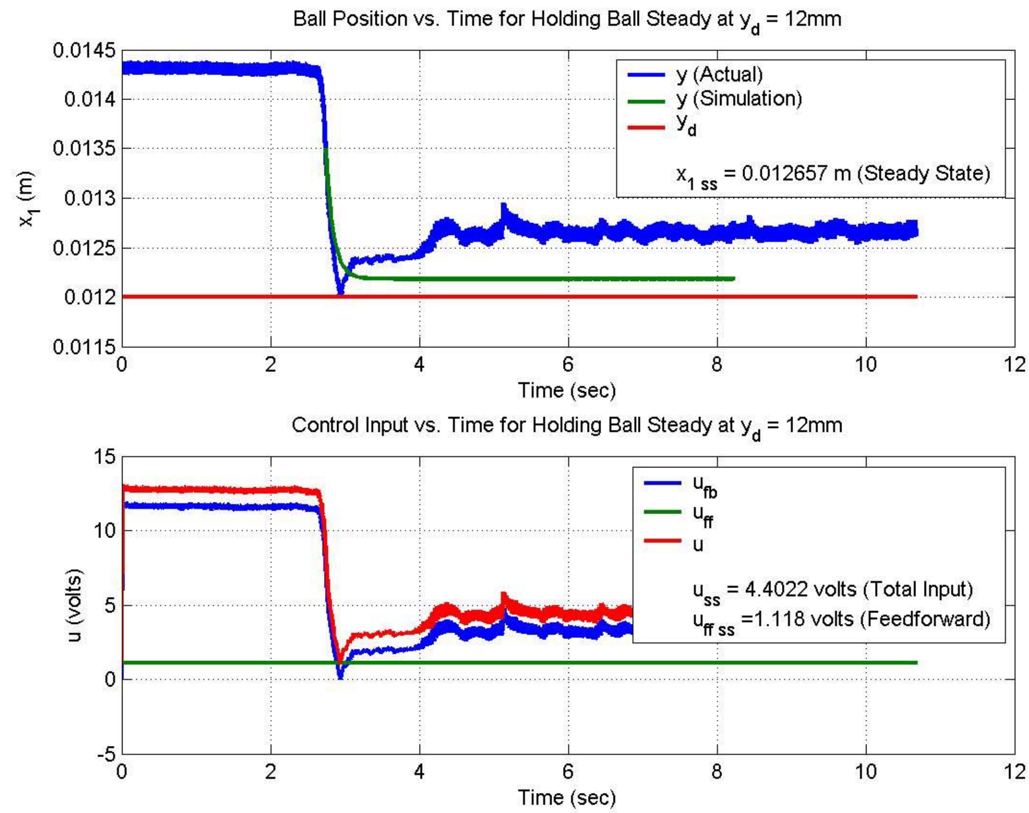

Figure 11: Simulation and actual results holding 12 mm

The response from 0 to roughly 4 seconds shows where the ball is being inserted by hand into the magnetic field. Once it is in the field, the ball stabilizes at a steady state value. As can be seen, the simulation appears to match up with the experimental data to some degree but the steady state error between the simulation and the actual system is still present. Also notice that there is a steady state error in both the simulation and the experiment. This is especially noticeable by looking at the control inputs.

Based on the model of our system, we calculate a feedforward control input of roughly 1.1 volts to hold the ball at 12mm. Since we are holding the ball steady, if we had a 100% accurate model, this would be the trim voltage required to hold the ball at 12mm and the feedback input would be 0. However, notice that the total control required to hold to ball at 12mm is roughly 4.4 volts. Therefore, we can interpret this is overestimating the size of the B matrix by a factor or 4. This leads to the question, given that our model has significant differences, should model based inversion be used?

.

Uncertainty Analysis[1]

When holding the ball at a constant 12mm, the same controller was used in both simulation and in the actual system yet there were still errors. Therefore, we can safely assume that all the variation comes from the fact that we have an inaccurate plant model. The nominal linear model of the system about 12mm is given Equation 9.

$$G_o(s) = \frac{-5.347}{s^2 - 1331}$$

Equation 9

Given that the trim voltage required to hold the ball at 12mm in the actual system was 4 times more than what was calculated, we can reduce the B matrix by a factor of 4 to obtain an estimate of the actual system as shown in Equation 10.

$$G_o(s) = \frac{-1.642}{s^2 - 1331}$$

Equation 10

Note that this is a decent estimate at low frequencies given that this estimate is based on a steady state response (effectively the response at zero frequency).

The difference between the nominal and the actual system gives an expression for Δ(s) as shown below

$$\Delta(s) = \frac{-3.7052}{s^2 - 1331}$$

Equation 11

We can plot the norm of Δ(s) and the norm of $G_o(s)$ vs. frequency to obtain Figure 12.

[1] See Should Model Based Inverse Inputs be used as Feedforward Under Plant Uncertainty? by Santosh Devasia. IEEE Transactions on Automatic control, Vol.47, No.11, Nov. 2002 for more information.

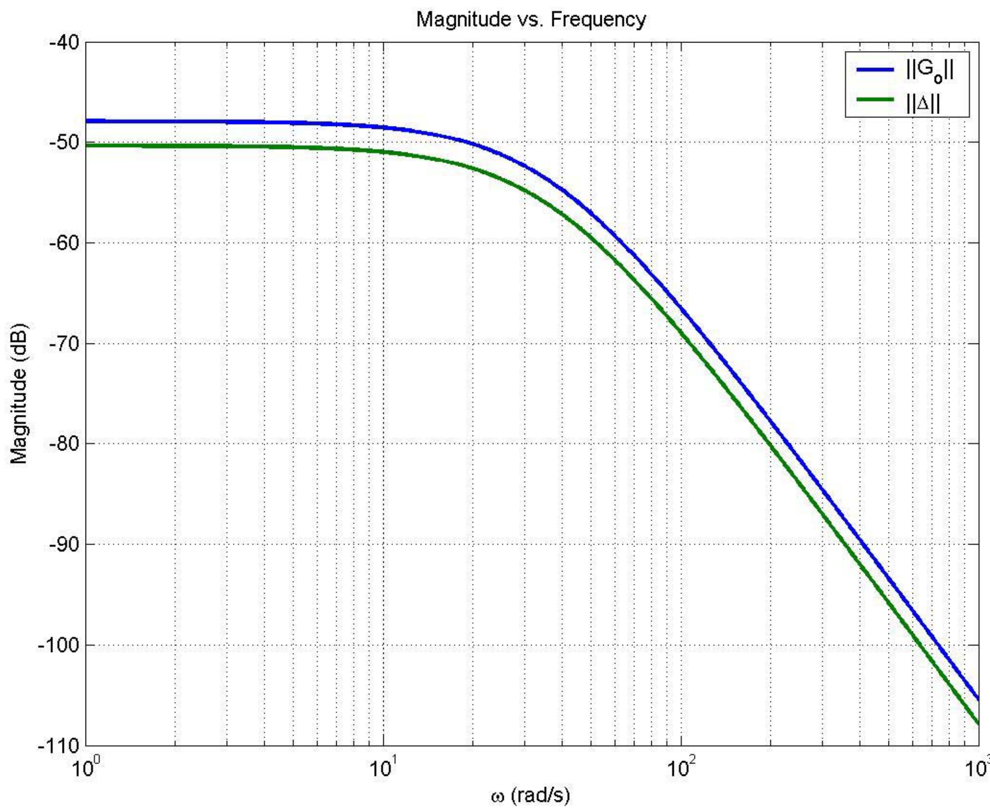

Figure 12: $\|\Delta(j\omega)\|$ and $\|G_o(j\omega)\|$ vs. ω

As can be seen, since we have $\|\Delta(j\omega)\| < \|G_o(j\omega)\|$ for all ω, we see that we should expect an improvement using feedforward and feedback as opposed to just feedback alone.

Note that this is only true for low frequencies considering that the above expression for $\Delta(s)$ was derived using data from a zero frequency signal.

Simulated Response

Now that we are confident in our simulation model, we can use it to design a feedforward input. The original idea was to use feedforward control to start the ball attached to the magnetic tip and then drop it into a position where it can be stabilized using feedback. The desired trajectory and its derivatives are shown below in Figure 13. Recall that a position of $y = 0$ corresponds to the ball stuck to the tip of the magnet and increasing y means the ball is falling away from the magnet. Notice that we start the ball at a non-zero initial condition to avoid singularities in the simulation since the magnetic force function (Equation 3) is undefined at $y = 0$.

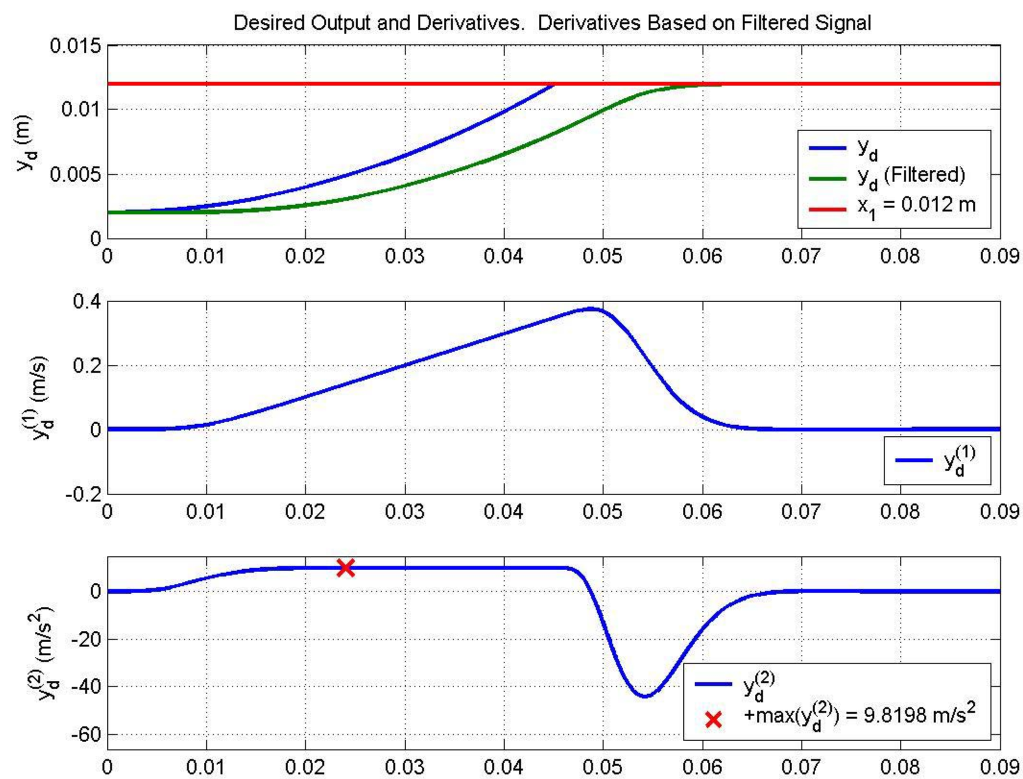

Figure 13: y_d and derivatives for dropping ball trajectory

We need to be wary of the fact that the actual system cannot follow any arbitrary trajectory. This is because the magnet is only able to apply an attractive force, not a repulsive one. This is because if the voltage on the magnet becomes negative (thus the current becomes negative), the magnetic field reverses direction but the steel ball instantly realigns itself so that it is attracted to the magnet once again. Therefore, the maximum positive acceleration possible is the acceleration due to gravity. As can be seen from Figure 13, this physical constraint is satisfied.

With the desired trajectory and its derivatives defined, we can calculate the feedforward input given by Equation 6 and we obtain the plot shown in Figure 14.

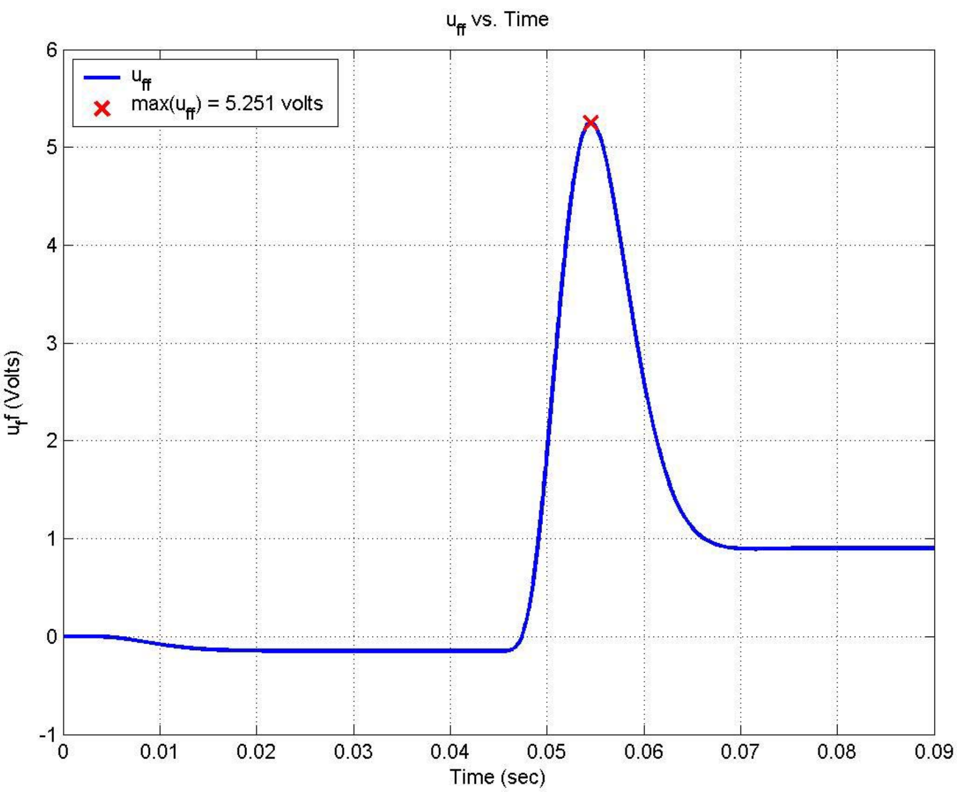

Figure 14: u_{ff} for ball dropping trajectory

Once again, there is a physical constraint on the feedforward input. The D/A card is only capable of supplying +/-10 volts. Therefore, the maximum control (feedforward + feedback) must not exceed +/-10 volts. The system was modeled with an amplifier already in the path with a set gain of 3.6, therefore, we cannot simply add another amplifier if the control input became too large without re-computing the inverse input. As can be seen, the control saturation limits are also satisfied with this trajectory.

We can now apply this feedforward input to the simulation to obtain Figure 15

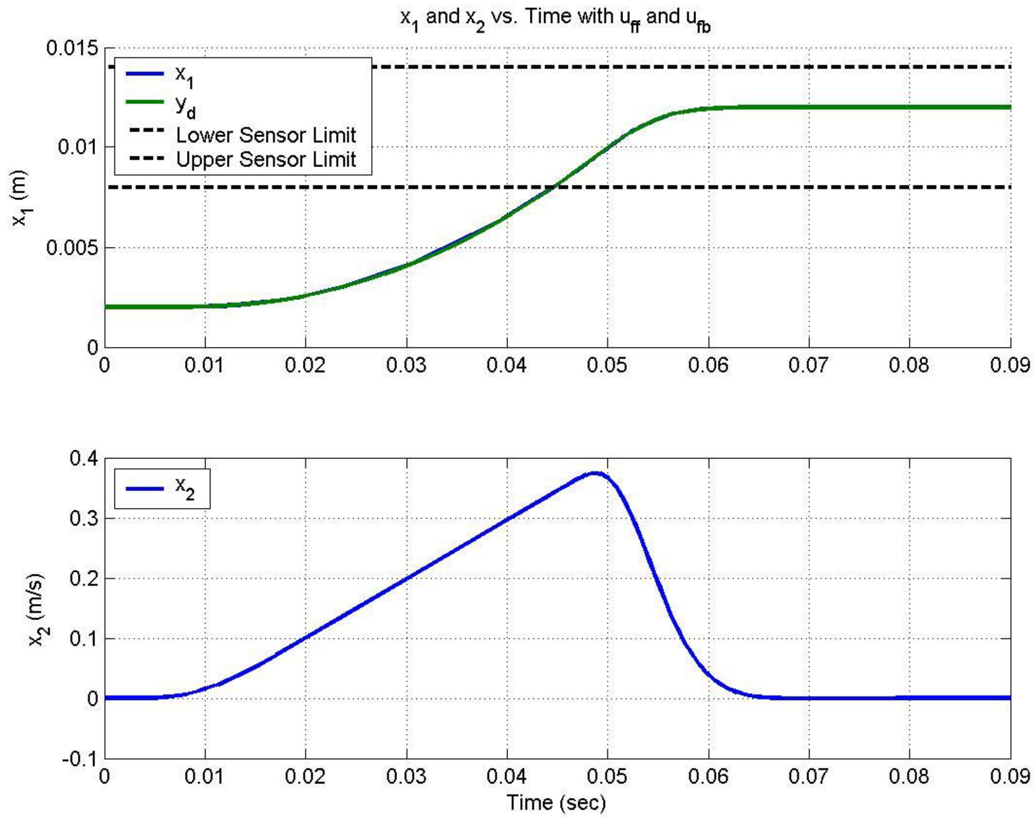

Figure 15: Simulated response of system states

As can be seen from Figure 15, we have to deal with another limitation of the hardware, name the sensor deficiency. As was shown previously in Figure 4, the sensor only is able to supply the ball position when x_1 is between 8 and 14 mm. Therefore, in our simulation, we only apply feedback to stabilize the trajectory if x_1 is between 8 and 14 mm. As expected, we obtain perfect tracking. In fact, we should obtain perfect tracking even if we used only feedforward. However, since the nominal system is unstable, the output would eventually become unbounded as the small numerical errors grew. For this region, we need to have feedback to stabilize the trajectory.

This trajectory is extremely difficult to obtain using the actual hardware. The reason for this is because this requires a very accurate model of the system and as we already showed, the system model is only somewhat accurate. Even if we were able to apply feedback, we showed that feedback was only possible in certain range. When the ball is falling, it goes through the range where feedback is possible very quickly. Therefore, there is only a narrow window of time where we can catch the ball. After many failed attempts on the hardware, we concede to only simulation this response and leave the actual tracking of this trajectory with the hardware for future work.

Experimental Results

Another trajectory that we can implement on the hardware is picking up the ball off a stand. The basic idea is to start the ball at 13mm on a stand and then use feedforward and feedback input to pickup the ball to 11.5mm. The required feedforward input for this maneuver is shown below in Figure 16 (zoomed in to area of interest).

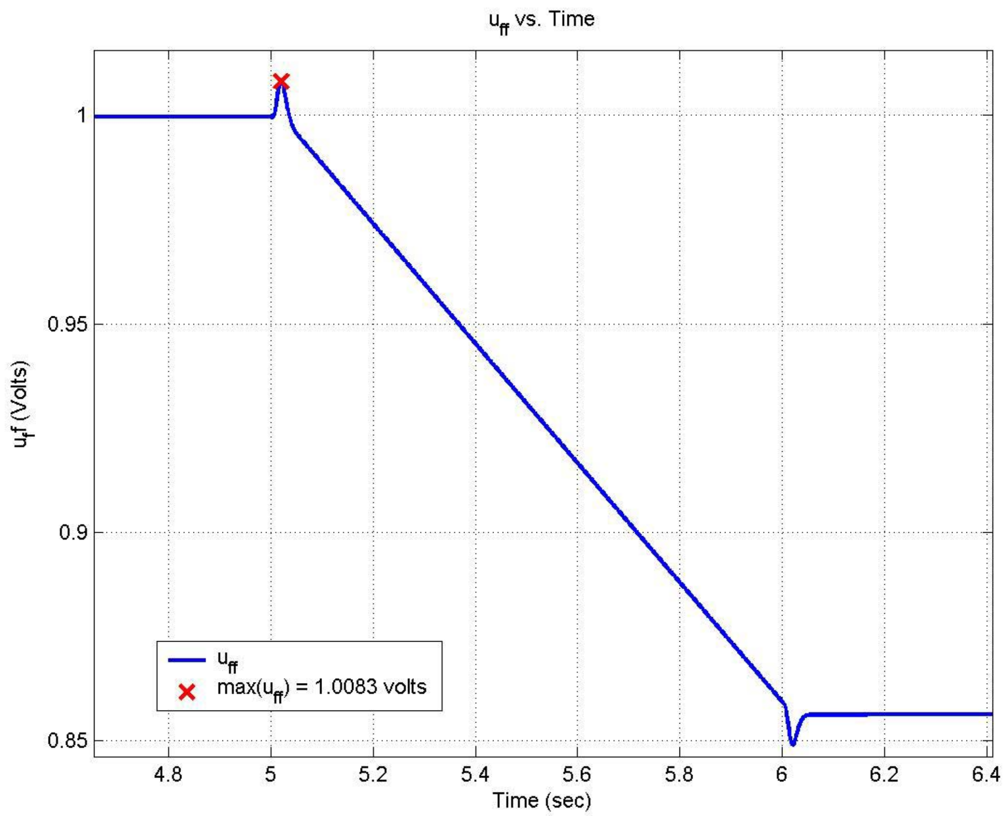

Figure 16: u_{ff} for picking up ball (zoomed in to region of interest)

The two flat regions are basically the trim control required to hold the ball at steady state at 13 and 11.5mm, respectively. The small increase in control at *t=5* seconds is to get to ball to accelerate towards the magnet. After this, the control starts to decrease since it requires less control as the ball moves closer to the magnet. As the ball reaches the final position, the input briefly drops to decelerate the ball before it comes to the final resting position.

We can then apply this feedforward input to both the simulated and actual system to obtain Figure 17.

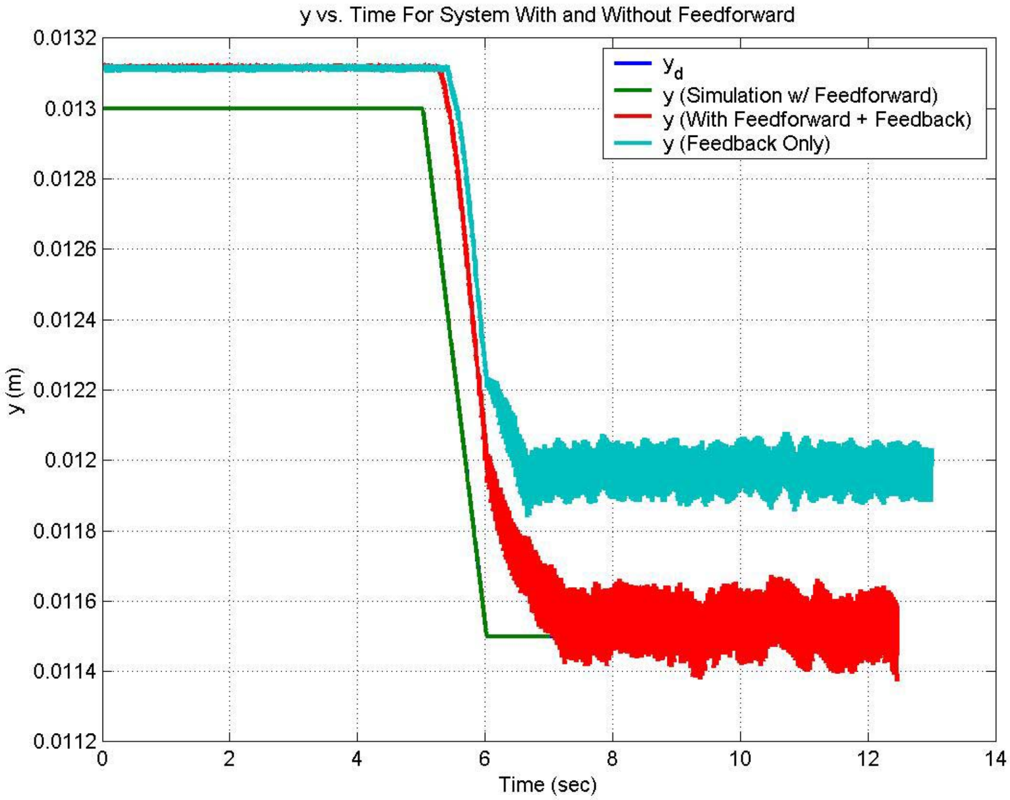

Figure 17: Actual and simulated trajectory for picking up ball

As can be seen, we obtain perfect tracking in simulation (green and blue curves lie on top of each other). With the actual system, we see that when we use feedback alone, we obtain a steady state error. This should be expected considering that we are only using PD control and therefore, a finite amount of error must exist in order for the controller to generate a control signal to hold the ball. However, notice that when we use feedforward and feedback together, the steady state error becomes much smaller and the output tracking greatly improves. This agrees with the results from the Uncertainty Analysis section where we showed that we should expect and improvement when using both feedforward and feedback together as opposed to just feedback alone. The small initial condition error of 0.1mm corresponds to the fact that in the lab, it is very difficult to position to ball to this fine a resolution so we simply live with mismatched initial conditions.

We can also look at the inputs during this pickup procedure for both situations where feedback and feedforward are used and only feedback is used. This is shown below in Figure 18.

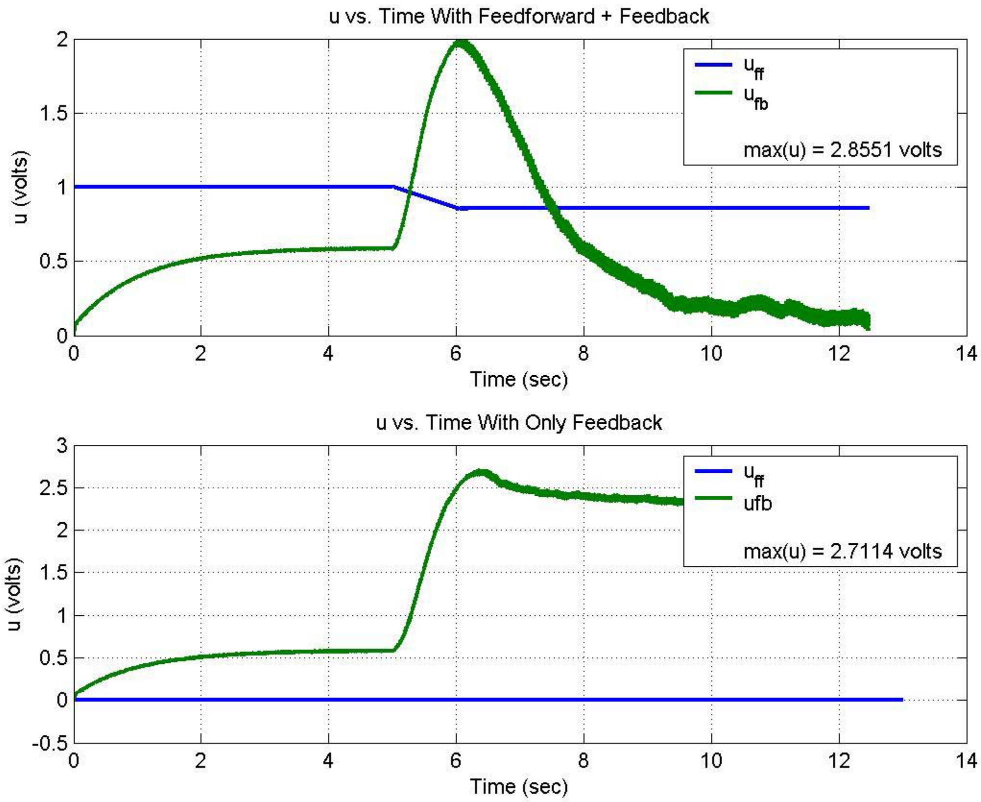

Figure 18: Inputs when feedforward and feedback are used in pickup maneuver

As can be seen, in the case where feedback and feedforward are use (top graph), the feedback input magnitude is smaller and appears to become near zero at steady state. This shows that at steady state, the feedforward input is holding the ball at the correct position. However, we can see that our model is not very good considering that if our model were 100% accurate, the feedback input would be zero for all time. Obviously, this is not the case.

The bottom graph shows the input when only feedback is used. As can be seen, the magnitude of the feedback input is greater in this situation. Also notice that in both situations, the maximum total input is roughly the same. Therefore we see the benefits of feedforward control, namely the maximum control applied is the same, but the output tracking is superior using feedforward control.

Conclusions

Overall, this project was a good experience in showing the application of feedforward control to an actual physical system. Although the mathematical model is fairly simple, attempting to implement this on the hardware increases the challenge of this project significantly.

Experimentally, we can only operate in the range of 13.5mm to 10mm. If we attempt to stabilize the ball outside this region, the system becomes unstable due to the fact that a static controller was used. Although we could use a gain scheduling to obtain a larger operating conditions, we chose to use a static controller in this situation so that the results due to purely feedforward control was more exaggerated in the output.

This system is fairly simply to model in a simulation given that is a simple 2 state system. The main difficulty arises from fitting an accurate simulation to the actual system due to the highly non-linear magnetic force term that is somewhat difficult to calibrate. The physical system produces additional complications because it is somewhat time varying parameters (the magnet heats up significantly and thus, the magnetic force changes). However, despite these uncertainties, we see that we are able to use both feedforward and feedback control to obtain improved output tracking with noticeable success.

Appendix A
Raw Data

Table 2: Raw $F_{mag}(x_1,u)$ in N. u values are across top and in volts. x_1 values are down left side and in mm

	0.26694	**0.53444**	**0.87111**	**1.1269**	**1.3972**	**1.6658**	**1.9447**	**2.2275**	**2.5292**	**2.7897**	**3.0619**	**3.3417**	**3.625**
1	0.355	1.582	2.591	3.675	4.359	4.8	5.488	5.927	6.538	6.713	7.413	8.0675	8.177
2	0.175	0.702	1.341	1.751	2.177	2.467	2.662	2.95	3.123	3.364	3.435	3.4755	3.606
3	0.097	0.442	0.845	1.046	1.33	1.485	1.661	1.751	1.894	2.091	2.14	2.1865	2.189
4	0.061	0.289	0.563	0.735	0.952	1.058	1.186	1.268	1.371	1.476	1.508	1.5025	1.509
5	0.035	0.207	0.411	0.52	0.678	0.76	0.84	0.916	0.974	1.062	1.058	1.07	1.103
6	0.022	0.152	0.311	0.381	0.503	0.583	0.642	0.711	0.758	0.834	0.805	0.8155	0.824
7	0.012	0.114	0.239	0.308	0.377	0.455	0.516	0.555	0.605	0.654	0.623	0.631	0.654
8	0.005	0.089	0.191	0.246	0.317	0.367	0.413	0.449	0.485	0.539	0.503	0.509	0.517
9	0.001	0.066	0.15	0.2	0.252	0.302	0.343	0.375	0.413	0.45	0.409	0.414	0.426
10	-0.005	0.051	0.118	0.157	0.21	0.251	0.283	0.313	0.345	0.377	0.336	0.337	0.347
11	-0.008	0.04	0.097	0.131	0.174	0.209	0.243	0.268	0.293	0.325	0.279	0.2815	0.289
12	-0.011	0.029	0.079	0.104	0.15	0.179	0.206	0.23	0.253	0.279	0.236	0.238	0.247
13	-0.014	0.022	0.064	0.088	0.128	0.155	0.182	0.202	0.223	0.248	0.202	0.2035	0.212
14	-0.015	0.015	0.053	0.073	0.109	0.135	0.157	0.18	0.198	0.222	0.174	0.1745	0.183
15	-0.016	0.01	0.042	0.061	0.092	0.117	0.14	0.16	0.178	0.198	0.151	0.152	0.16
16	-0.017	0.005	0.033	0.049	0.081	0.104	0.124	0.144	0.159	0.179	0.131	0.1305	0.141
17	-0.019	0.002	0.027	0.042	0.07	0.094	0.114	0.13	0.145	0.162	0.115	0.1155	0.125
18	-0.02	-0.001	0.021	0.035	0.063	0.082	0.102	0.118	0.133	0.15	0.101	0.1025	0.109
19	-0.021	-0.004	0.016	0.029	0.055	0.075	0.092	0.109	0.123	0.139	0.09	0.0905	0.098
20	-0.022	-0.007	0.012	0.024	0.05	0.068	0.084	0.101	0.115	0.13	0.08	0.0805	0.088
21	-0.022	-0.009	0.009	0.018	0.045	0.062	0.079	0.094	0.108	0.121	0.072	0.072	0.08

Appendix B
Using Matlab Functions[2]

Function	Comments
sim_ball_MAIN.m[3]	Simulate the ball dropping, pickup, and non-zero initial condition response (MAIN CODE FOR DEMOS)
sim_ball_ode45.m	Function called by ode45 to simulate ball dynamics
numerical_linearization.m	Linearize the experimental data about a equilibrium point to obtain a state space model
nonlinear_Fmag.m	Curve fit the magnetic force term
uncertainty.m	Perform uncertainty analysis
compare_results.m	Compare simulations and actual experimental data

[2] These are a few selected files. Please contact the author for descriptions of all the files.
[3] This is the main Matlab file which can be run for demonstrations