Final Exam Extra Credit

PID Controlled Magnetic Levitator

By: Cameron Hanson

ECE 4330

Dr. Mohamad Hassoun

Winter 2019

I have neither received nor provided any assistance on this work. ______________________

Introduction:

This project required us to implement a PID controlled system and because I wanted to do something new, I decided to try and create a magnetic levitator system. This system would have an electromagnet that tries to keep a neodymium magnet suspended in the air at a set high determine by a potentiometer.

Theoretical Results:

In theory, by using an electromagnet, we can create a force that will hold the neodymium magnet in the air. We also only need one electromagnet because gravity is acting against the magnet and trying to pull it down. Therefore, we do not need to worry about having to figure out how to change each signal, so it could work with a two-electromagnet system. From the research I have done, this is an unstable system as it sits because it is not a linear system since the force acting on an object by a magnetic field decreases by the square of the distance. So, to make the system stable, a PID controller is added.

Experimental Results:

This project has gone through several iterations to become what it is now. The first setup we had, used a setup like what we have now (discussed in a bit) but we used too sensitive components and didn't compensate for it. The second iteration was changed so that it only required a single sensor at the bottom of the travel for the magnet. This however, led to some unfortunate consequences due to the electromagnet affecting the sensor reading. This became very apparent when we would try and use the differentiator portion of the PID controller. Figure 1 shows the setup used for the system.

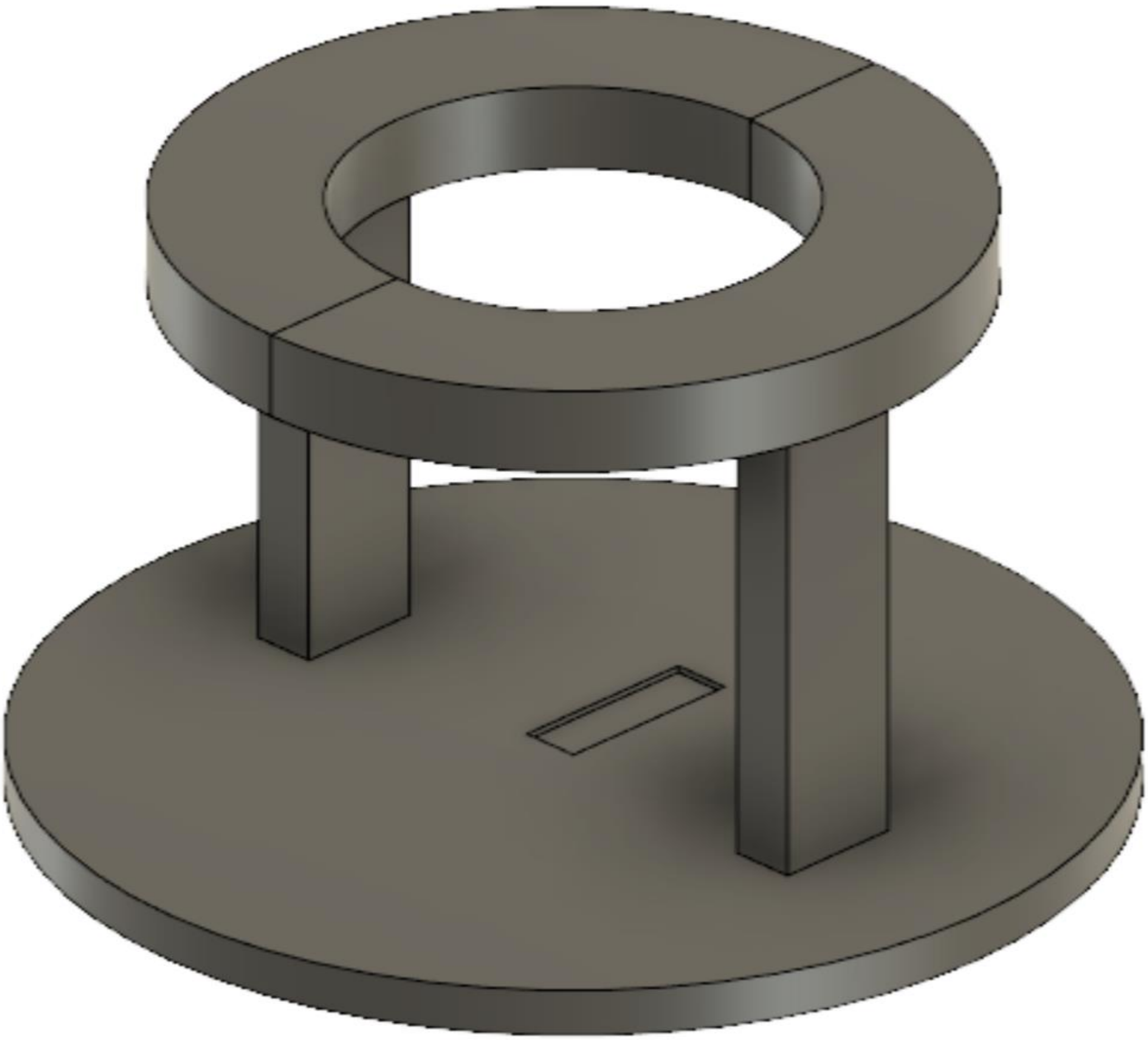

Figure 1. Version one model

The sensor when in the hole in the middle of the larger thinner cylinder and the electromagnet was friction fit into the ring. Here, we had set the system up so that as the magnet gets closer to the bottom of the system, the voltage output from the sensor was higher.

Now come the issues, firstly, this system wasted a lot of power since it was constantly on with a high duty cycle. This heated up the electromagnet a lot and almost caused it to catch fire. We were never able to get stead stable oscillations like the Ziegler Nichols method suggests. Nonetheless, we pressed on in the hopes of stability. We tried all different combinations, but we kept getting this pulsing of the duty cycle from a low value to a high value and we could not figure out why. It wasn't until I disconnected the electromagnet from the system that I figured it out. What ends up happening is as the electromagnet turns on, it affects the sensor and increases the output voltage. The controller then thinks that the magnet is falling and so it increases the current to the electromagnet to bring it back to zero. However, this just makes the problem worse. Eventually the electromagnet saturates at 100% duty cycle and so limits the output voltage of the sensor. The differentiator then stops acting because the sensor output is no longer rising. Once the output voltage reaches the set point, the cycle continues again because the duty cycle is not zero. This just repeats and meant that stability could never be achieved with the setup.

Next, we ordered less sensitive hall effect sensors, so that we don't saturate them with the electromagnet because we were having issues with the hall effect sensors being directly on the electromagnet. Figure two shows the latest iteration of the system.

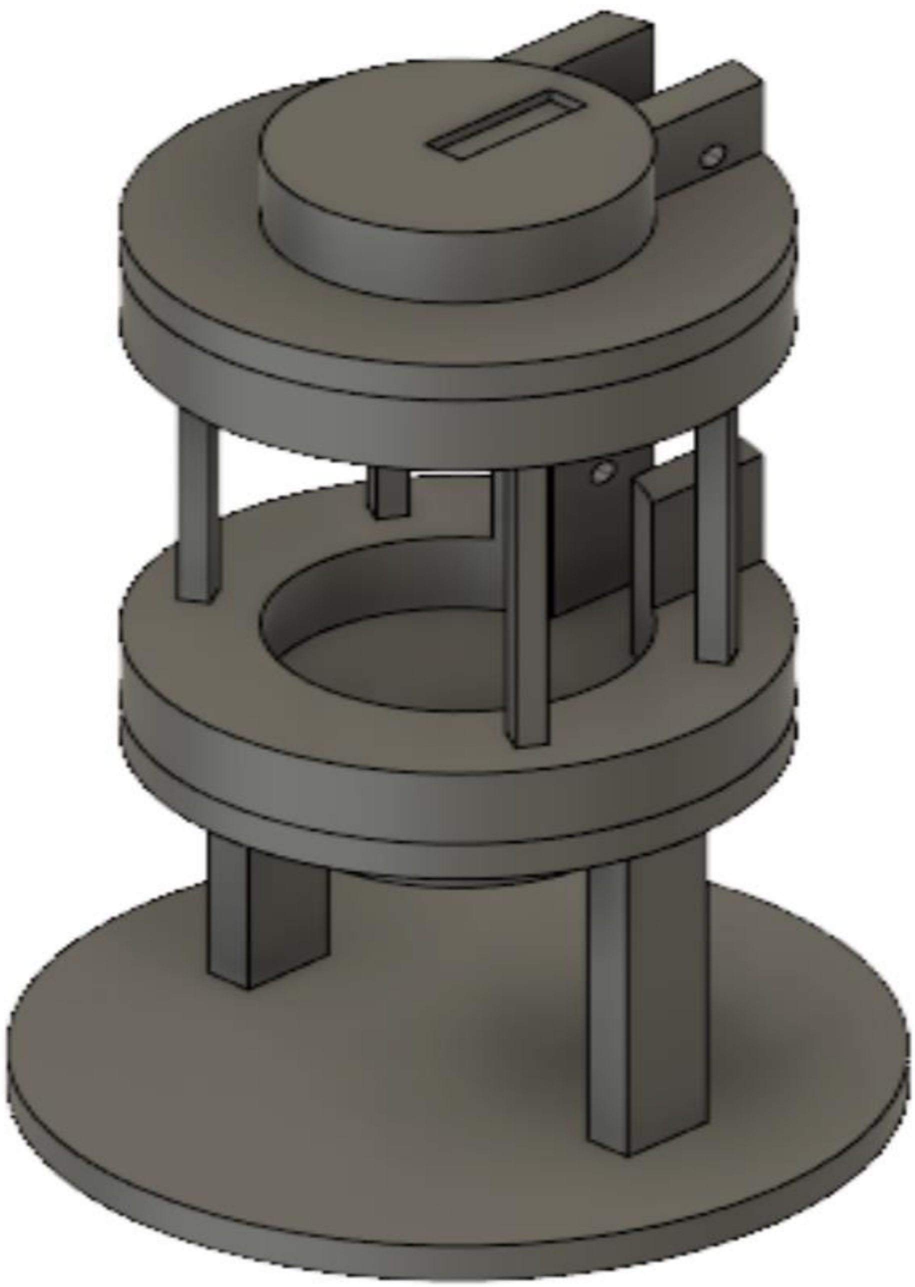

Figure 2. Third iteration model

In this iteration, a few things were changed. Firstly, we went back to our original design of one sensor on top and one sensor on the bottom. This helped because we were able to cancel out the voltage change that was caused by the electromagnets field. We also added a 15mm buffer to the bottom of the system to stop the magnet from sticking to the iron core of the system which in the last iteration caused a lot of problems. The sensors are equidistant from the electromagnet and so allows for direct cancelation just be using a x1 gain difference amplifier. When trying to get the right PID values, we just attached the PID controller to the system and it worked the first time. Somehow, by playing with the values with the second iteration were in tune with the values needed for the third iteration. Now, it didn't stay in the air for long as the values were not perfect. It did require a bit of tuning to get it to work exactly like we wanted.

Unfortunately, I left the paper we used when we changed values to see what happened with the toolbox and system at school, and as of writing this report am not able to attain. I id however, have to turn the P value up a bit to get it to pick the magnet up off the bottom plate, but I turned it up too much because it started to shake violently, and, the oscillations became so large that it began to fail, and it shot out. Once I found the right P value, I started playing with the I value to bring the magnet, so the error was almost 0.2V (which the f(t) was set to), The D value was the one that did the most for the controller. It stopped the oscillations that increasing the I value caused and now, apart from some spin and slight wobble, stays fairly still.

I do not know the values of P, I, or D because I was afraid to take it apart to measure the values since I wasn't sure it would work again if I did that. So again, at the time of writing this, I do not know the values for this system that caused it to work.

Circuit Schematic:

Figure 3 shows the overall circuit schematic

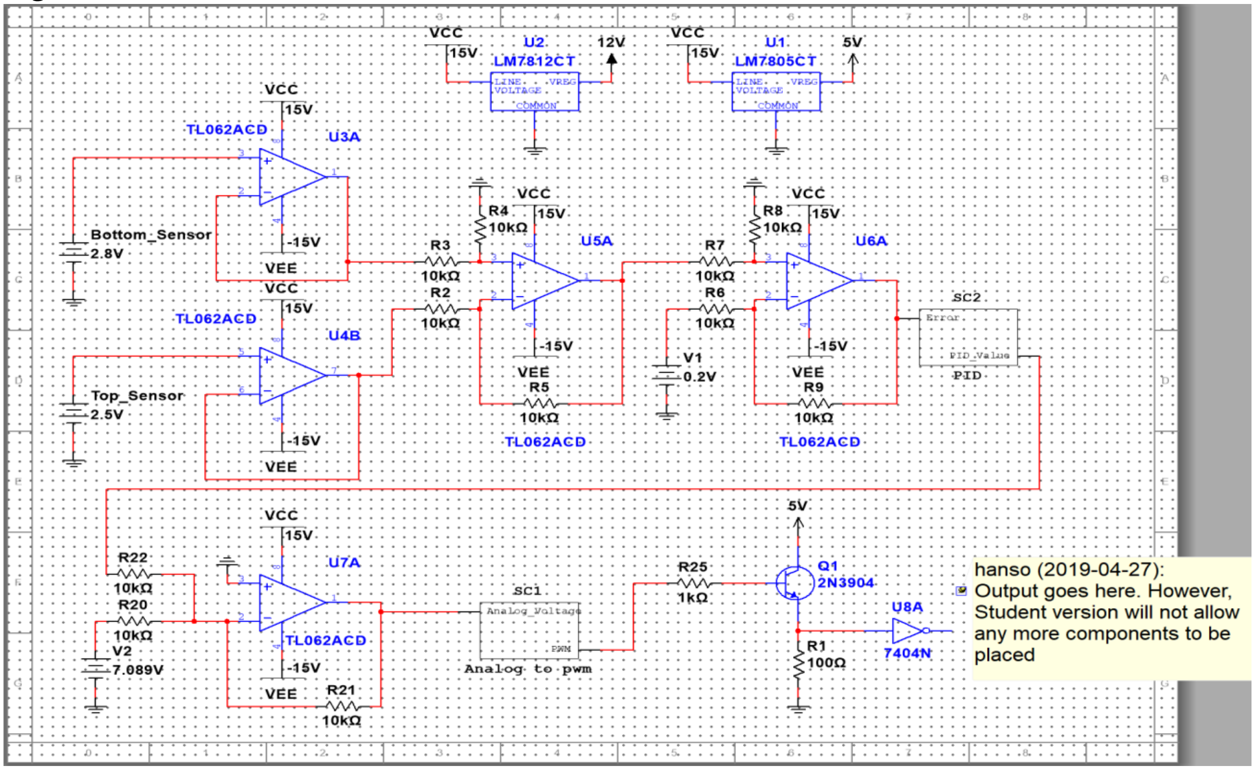

Figure 3. Overall circuit schematic

First, we buffer the two sensors because they have a max current output of 5mA. Then we go into a difference amplifier with a gain of x1. The difference amplifier will cancel out any increase or decrease in sensor reading due to the electromagnet being on. Next, we pass this through another difference amplifier that will subtract the f(t) signal. We do this because we want a positive voltage when the magnet gets closer to the bottom sensor. This seems odd to do, but it will be fixed by a later stage. This is the error signal that gets passed off to the PID controller. Once it comes back from the PID controller. We take that and add it to a 7.809V signal to shift it up by 7.089V. This is needed for the voltage to PWM converter. Next, we pass it to the analog to PWM converter and the PWM signal that gets passed, will be buffered from 12V to 5V peak signal. Finally, we put this into an inverting logic gate. This effectively converts the duty cycle from say 5% to 90%. It becomes a “1-Duty cycle” operation. We need this because as the magnet gets closer to the sensor, we want the electromagnet to turn off, so it stops pulling on the magnet and it falls due to gravity.

This raises some questions like, why did we use y(t)-f(t) instead of f(t)-y(t) like normal? The answer to this is that we didn’t have to do this, we didn’t think to try it the other way. Using f(t)-y(t) probably could have been used to remove the not gate at the very end since if the magnet was farther from the sensor, the error would be positive like we would want without the not gate.

Another question is that why we used a not gate in the first place, when we could have just used the transistor as the not gate? Again, it is because we didn't think about doing it that way.

Both could be future improvements. However, the first is unlikely to change much if the second option is implemented. The reason is the we still need a voltage buffer between the PWM converter and the H-Bridge circuit since the H-Bridge only takes in 5V max.

Figure 4 shows the PID controller.

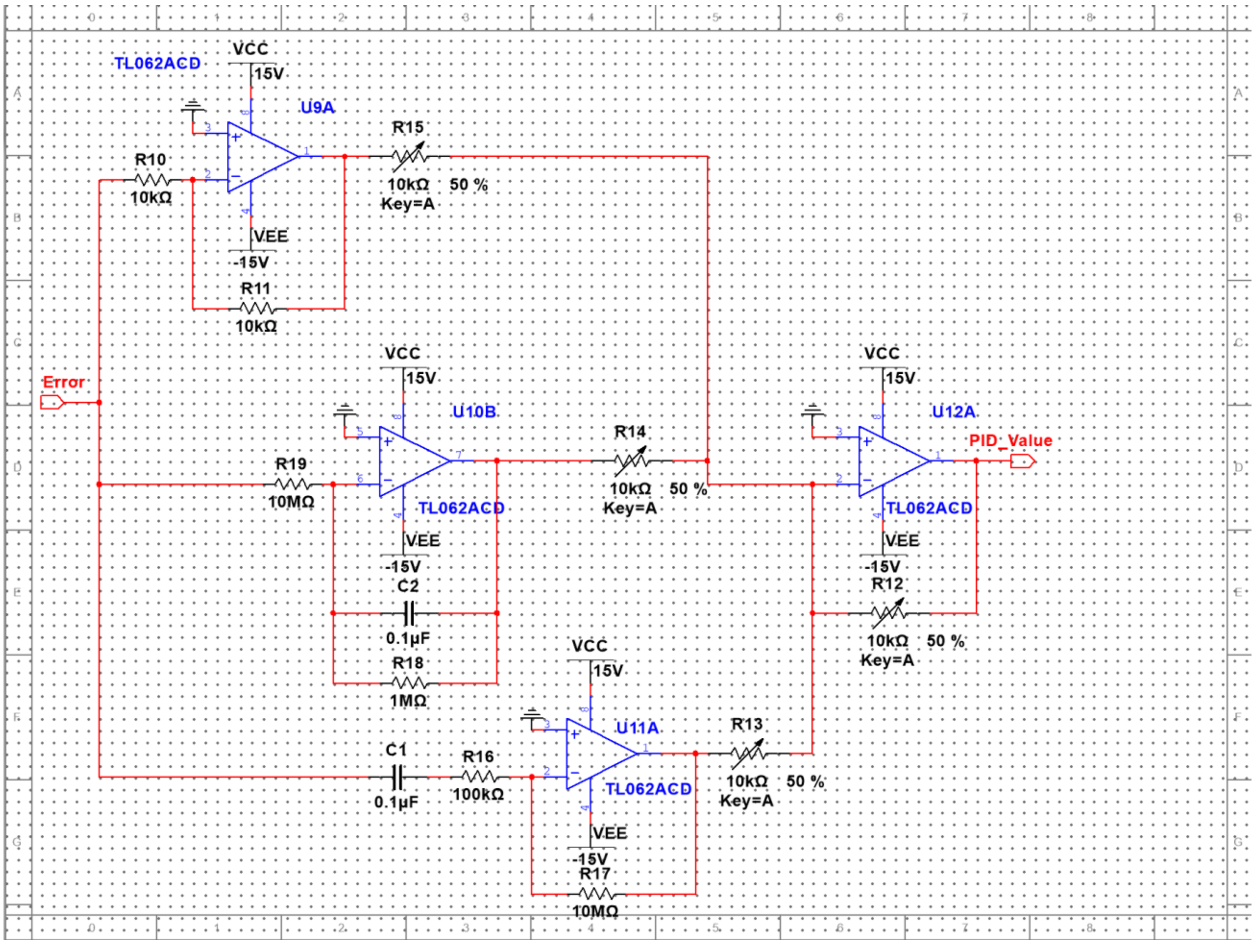

Figure 4. PID controller

This is a fairly stock standard PID controller. We made a few modifications to make things easier to use. First, we made the gains of each stage equal to one. This forced the system to stay unsaturated unless the error got really large which doesn't happen. We changed the summing amplifier to implement the gains of P, I, and D as well as added a potentiometer to the feedback resistor to create really large and small gains. The filter frequency for the differentiator was set to be 100Hz which we thought was more than enough to filter out the frequency of the electromagnet which is set somewhere around 37kHz.

Figure 5 shows the analog to voltage converter that was used.

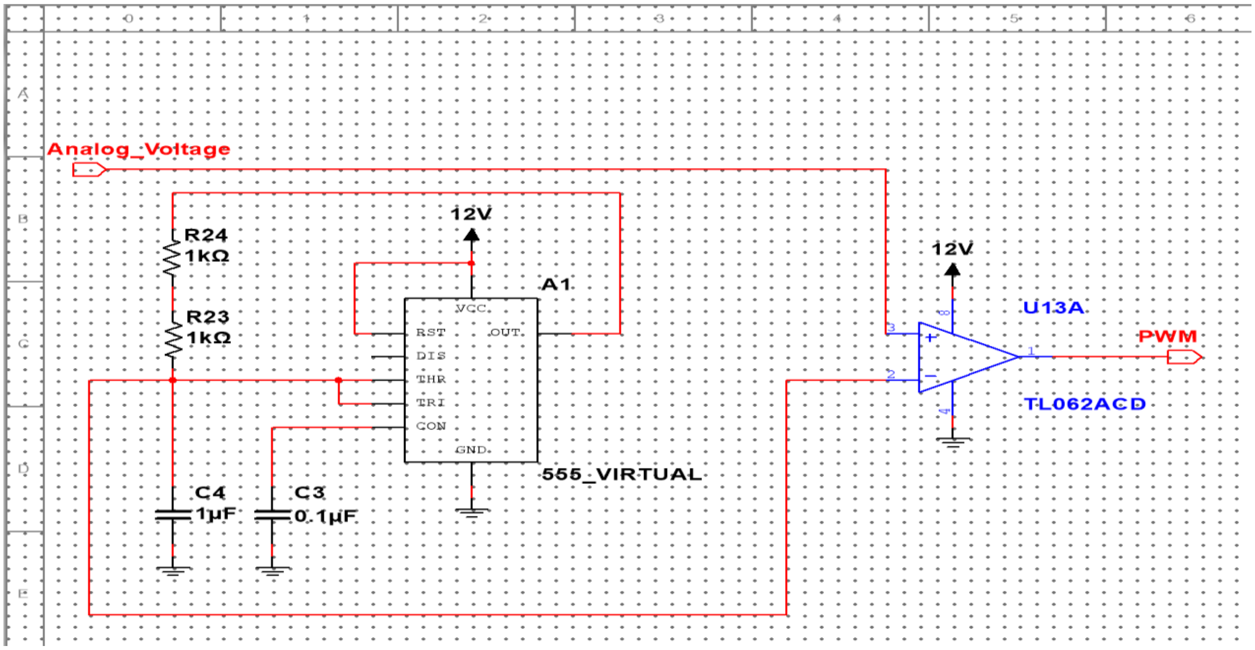

Figure 5. Analog to PWM converter

This circuit is also simple and just consists of a 555-timer circuit and a comparator. This circuit works by creating a charging and discharging capacitor that goes through 1/3Vcc at its lowest and 2/3Vcc at its highest. Therefore, we needed to shift the PID error signal up by 7.089V. The oscillating voltage is fed into the inverting terminal of the comparator and the analog voltage is fed to the non-inverting terminal. When the analog voltage is higher than the current voltage on the capacitor, the comparator voltage is high (12V) and when it is lower, then the output is low (0V). One thing to note is that the inversion of the duty cycle could have been done here as well by reversing the inverting and non-inverting terminals. However, we wanted to keep the block as close to the original source for easy troubleshooting.

Conclusion:

In conclusion, I learned that we can take almost an unstable system and control it to maintain a set point. This project took many iterations to work but we have finally found a system that works and can be controlled. There are a few minor improvements that could be made to remove components and cheapen the overall cost of the unit.

Appendix A:

Link to youtube video: https://youtu.be/y8KLMKgDzNk