

Lab 2: DIY 3D Scanner

Julia Benton & Richard Gao

October 3, 2019

1 Description

In this lab, we built and programmed a 3D scanner using a pan/tilt mechanism. We scanned a letter "J" of known, well-defined geometry and visualized the output from our scanner. We got an introduction to sensors, which transduce a physical, real-world quantity into a more easily measurable, physical quantity, and actuators, which transform one form of energy into mechanical work. In this lab, we used an infrared distance sensor and two hobby servo motors. An infrared distance sensor is an infrared emitter (LED) in addition to an infrared detector (photodiode). The detector measures the intensity of the IR light reflected off of an object in its field of view. The output of the sensor is an analog voltage.

A hobby servo motor is made up of a DC motor connected to a potentiometer (to measure the shaft angle), hardware proportional-integral-derivative (PID) controller (to perform position control), and a gear train (to increase torque and decrease speed). A hobby servo is controlled using a square wave form that is pulsed at a fixed frequency. The time that the waveform signal is set to "on" determines what the position of the shaft should be. This scheme is called "pulse width modulation" (PWM) and the percentage of "on" time during the period is called the "duty cycle".

We also designed and fabricated a custom mount to house the IR sensor and two servos in order to carry out the pan/tilt mechanism and sweep across the "J". Using the 3D scanner, we scanned a cardboard cutout of the letter "J" and rendered the scan results.

2 Process

We closely followed the suggested steps in the lab handout since our team employed the strategy of making sure the building blocks all worked properly before bringing them altogether. First, we connected our sensor to the proper pins on our Arduino and verified that everything was correctly set up using the "AnalogInput" example from the Arduino development environment. Second, we ensured that the serial connection between our the Arduino and our laptop running a python script could relay sensor data using the "AnalogInOutSerial" example in the Arduino IDE and using the sample sketch created by the POE teaching team. Third, we tested the functionality of our servos using the "Sweep" example in the Arduino IDE.

Next, we calibrated our distance sensor by getting a set of voltage readings for known distances and determining the mathematical relationship between physical distance and the sensor reading. We increased the accuracy of our distance measurements by securing a tape measure to the side of the POE table with duct tape to ensure we were measuring a perpendicular to the sensor. We exported our measured data to Matlab and fitted a two term exponential to our plot, seen in Figure 1, using cftool.

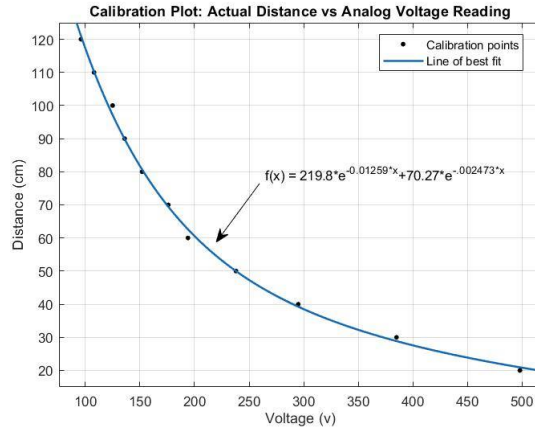


Figure 1: Our calibration curve is a plot of distance (cm) versus voltage (Volts). The line of best fit is a two term exponential.

We then had to test our calibration with distances that were not used. Using the tape measure, we had our code report the estimated distance at known distances we had not tested before. Plotting the known distances and estimated distances against the voltage reading from the sensor we arrived at the error plot seen in Figure 2. Looking at the data points, we determined that our function fits best between 20 and 50 cm, and this is the distance that we will use to scan our letter. It's noted that at further distances, the estimated distance starts to diverge slightly with the actual distance.

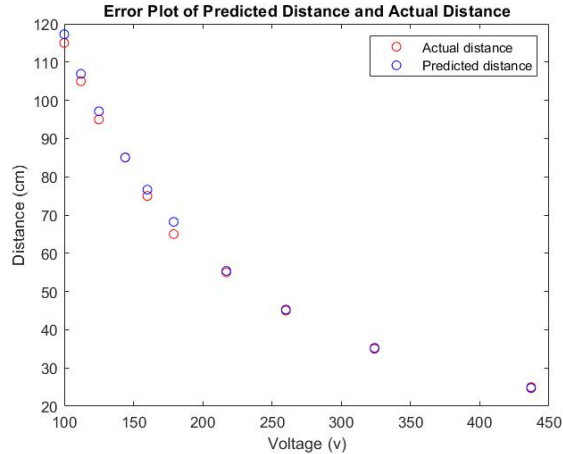


Figure 2: Our error plot is a plot of voltage (V) versus distance (cm). It shows the difference between the predicted distance and actual distance.

Next, we laser cut a cardboard letter "J" with cardboard supports so it could stand freely without a person holding it.

Then, we designed and 3D printed a mount, seen in Figure 3, to hold both servos and the IR sensor. Our setup for the one servo and two servo scan was the same, with us only moving one servo to create the 2D, top-down representation of the "J".



Figure 3: Our setup for both the one servo and two servo scan. We had to employ some creative, on-the-spot engineering to shift the sensor a little bit to be centered.

Putting together all the physical components we arrived at the schematic seen in Figure 4.

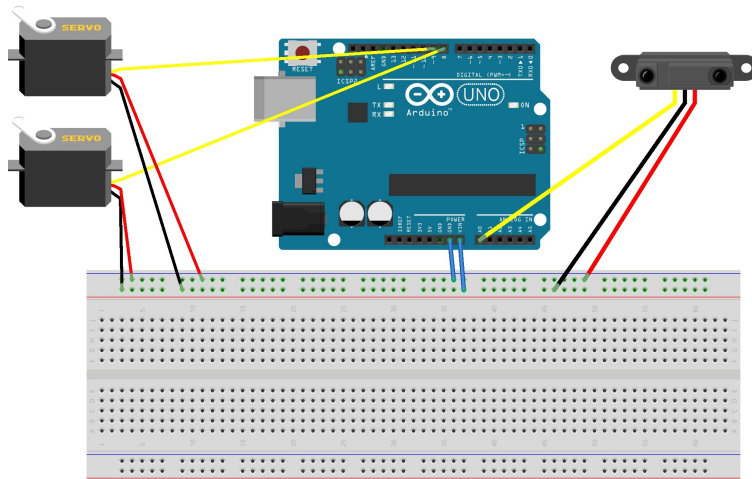


Figure 4: Schematic of our setup

To control the servos and create a scan we wrote an Arduino program that moves the servos in a pan/tilt mechanism and sweeps the sensor across the scanned object. This program moves the pan servo between 65° and 115° , and the tilt servo between 40° and 95° , getting the distance estimate from the IR scanner and sending the distance and servo angles to the python script for processing via serial.

In the Python script, we use this data to compute the x, y and z of this data point by converting

from Polar to Cartesian coordinates. This is because the distance and tilt angles reported from the Arduino are the components for a point in Polar form.

We then first got a 2D scan of our letter "J" by moving just the pan servo and using the x and y coordinates to construct a top-down plot with matplotlib seen in Figure 5. Looking at the plot we can see that there is a fair amount of noise, but generally we can see between -10cm and 10cm in the x axis there is a straight line the represents our letter.

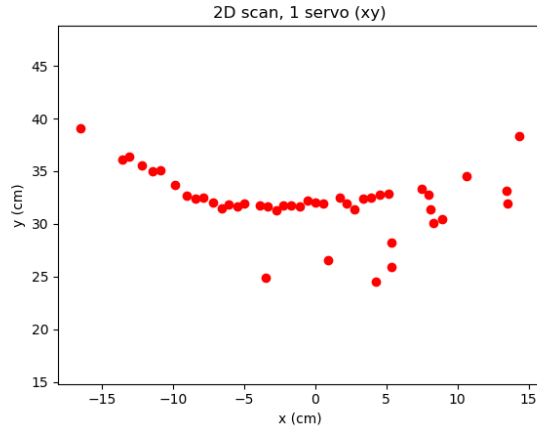


Figure 5: Our 2D, top-down representation of the letter "J" from our one servo scan.

After successfully creating our 2D scan, we then put our efforts into constructing a clean 3D scan. To do this, we moved both the pan and tilt servos, and calculated x, y and z coordinates for each data point. Plotting these point in 3D with matplotlib, we arrived at the visualizations seen in Figure 6 and Figure 7. Like the 2D scan there is a fair amount of noise, however, there is a dense cloud of points that clearly make out our letter "J".

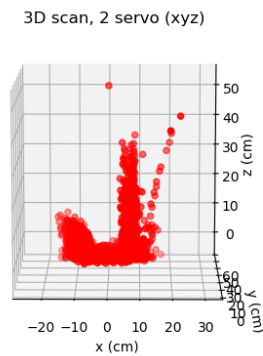


Figure 6: 3D plot of the scan from the front.

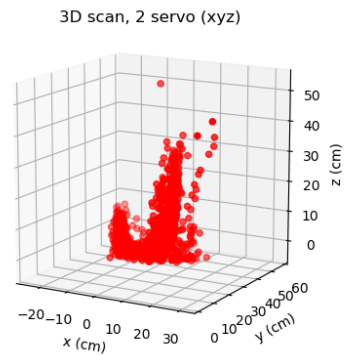


Figure 7: 3D plot of the scan from the isometric view.

You can check out our 3D scanner in action [here](#).

3 Reflection

One of the first things we learned in this lab is that switching which USB port on our laptop our Arduino is plugged into will change which COM is being used for serial communication.

We were also reminded of the importance of checking small setup details, such as plugging wires such as ground and voltage input into the rails in our breadboard and making sure that nothing comes unplugged. Being wary of this, we now know how to save hours of debugging.

A valuable strategy Julia learned from this lab is to try something new, fail, and improve in the next design attempt. She focused on designing the 3D printed mount, which included iterations that were not properly sized due to not considering tolerances or were too thin and snapped when pressure was applied. Julia also learned after having already designed and printed the mount that the IR sensor needed to be centered in the middle, leading to some on-the-spot creative engineering to offset the sensor back into a more central position. Julia learned some important skills in SolidWorks, which she did not have much experience in and gained experience in creating assemblies and mates.

Richard learned how to program the Arduino for servos and how important an accurate calibration curve can be for accurate results. He also learned to improvise and adapt, using scrappy methods to arrive at a working MVP that produces desirable results.

For the future, we can try minimizing the noise produced in the scans by doing some post-processing. One reason there may be so much noise is that our setup would vibrate slightly upon rotations, so the sensor may be occasionally getting inaccurate distances. We can amend this by learning from this attempt, and redesigning our housing.