Finger Tracking using Magnetometers

Alexander Mastrangelo and Paul Yoo, University of Washington

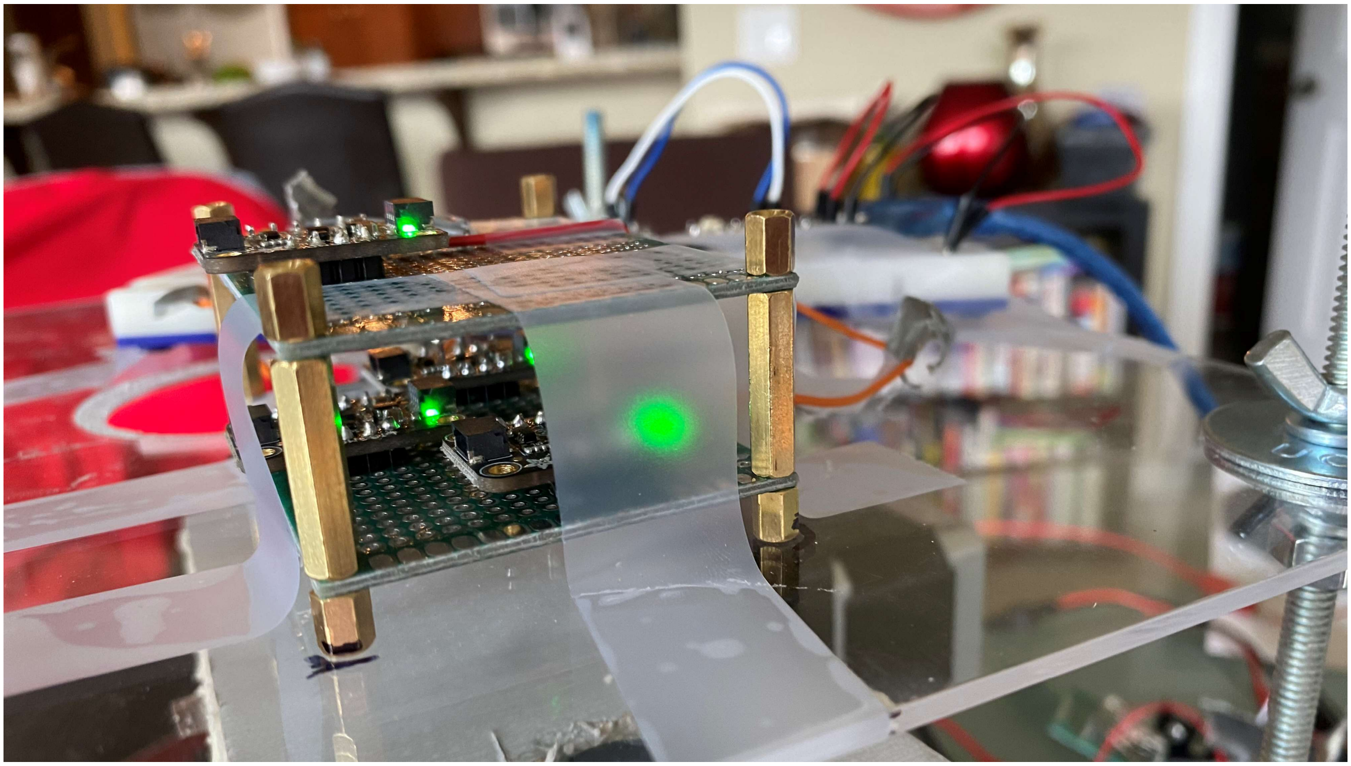

Figure 1: The finger tracker mounted to an XY table for taking measurements.

ABSTRACT

We designed, assembled and testes a system with 4 magnetometers to track the position of an electromagnet beacon driven with alternating current representing the position of a fingertip moving in space. Using this system, we measured the displacement of the electromagnet along a flat surface ~50 mm away from the magnetometers and we were able to track the position inside a 40x40 mm^2 region with about 2.6 mm mean error. The system can be utilized for hand and finger gesture recognition, virtual typing and instrument playing and 3D virtual gaming controls.

Introduction

Position tracking and mapping of the human body is a basic requirement that permeates virtual reality environments. The purpose of this project is to build a high-accuracy finger tracking for use in VR. We believe good finger tracking is important because it allows players to interact with VR environments without the needing a controller. Although there are many approaches for tracking position [1], magnetic tracking is a good solution for tracking fingers because they suffer from occlusion. This means that other approaches for tracking position such as lighthouse and marker-based tracking are difficult to use for solving this problem.

We based our system on the layout of the magnetometers used in the Finexus design by the University of Washington and Oculus [2]. As for the electromagnet, it is surprisingly difficult to get an electromagnet that can be mounted to a finger, so we had to build our own out of a solenoid and magnetic pins.

1.1 Contributions

1. Built a magnetic finger tracker
2. Implemented a model to solve the position of each finger
3. Measured accuracy of the model using an XY-table
4. Implemented new driver for complex wavelet drive

2 Related Work

This project is basically a reimplementation of the Finexus project, except with less time, less budget and less experience. As such its very much a subset of their work [2]. That said the implementation details of the hardware are very different.

3 Experimental Method

To implement the tracking, we mounted 4 magnetometers to perf boards, as shown in Figure 1. These magnetometers are driven through the i2c bus by a Teensy 4.0 Arduino. Given that the magnetometers (Adafruit LIS3MDL) share the same i2c bus, an i2c multiplexer is used to enable/disable magnetometers.

Using these magnetometers, we measured the magnetic field of an electromagnet that is driven by alternating current at 80 Hz. Ac current is used is to filter out–of-band disturbances such as the Earth's magnetic field, and other magnetic noise. He system can also be used to track multiple electromagnets driven at different frequencies [2], but that was not implemented in this work

Once we have the waveforms of the magnetic field measured, we analyze these signals using through the Fourier transform to extract the amplitude of the 80 Hz component of the electromagnets' field.

4 Implementation Details

To make this system work we had to mount the magnetometers at known positions relative to each other. This was done very carefully as any uncertainty here will break the model. Thankfully the perfboard matched this requirement exactly, as we can just place them at a fixed offset relative to each other. This only works for 3 of the 4 magnetometers required, because the last one is on a different plane than the first 3. In our implementation we chose 8 holes to be the distance between sensors, which is exactly 20.3mm.

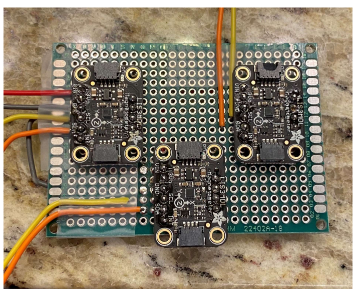

Figure 2: The magnetometers mounted to the perf board

The perf board used had mounting holes on the sides, which were used to connect the top and bottom boards. It's important to note that the wires and spacers used were nonmagnetic, as to avoid introducing any soft iron effects into the system [4]. The four magnetometers were mounted similar to the Finexus configuration [1], with coordinates of (0,0,0), (20.3, 20.3,0), (20.3,-20.3,0), and (0,0,21.6).mm respectively. The power and voltage pins of each chip were shorted, and the i2c pins were routed out the sides.
Communication with the magnetometers was established over the i2c bus. Since all the magnetometers shared the same i2c bus, we used a TCA9548A to split the i2c bus; allowing us to talk to all the sensors, even though they all share the same address.

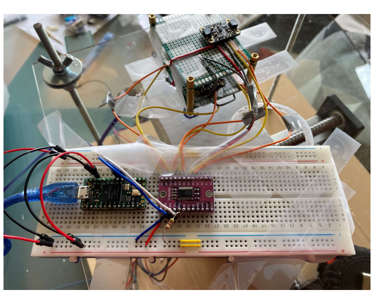

Figure 3: Photograph of the Teensy 4.0 and I2C multiplexer

Prior to any measurements the magnetometers need to be calibrated to normalize the scale factors and eliminate biases. The calibration procedure is simple, we rotate the sensor in random directions, while keeping them away from anything magnet. We used MotionCal by PJRC [4] to solve for the calibration matrix and biases for each sensor, and PySerial to stream it to the PC [3].

To drive the electromagnet, we built a voltage controlled current source. It uses a rail to rail LM358 op amp, and a TIP31 power BJT to push a current through the magnet defined by a signal generator and a resistor. The current source implementation assures high fidelity of the resulting magnetic field waveform and permits the use of other non-sinusoidal waveforms (our intent was to use wavelets, but we ran out of time).

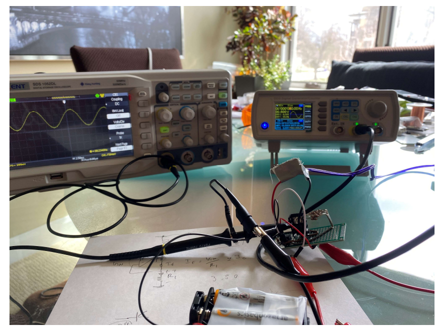

Figure 4: The signal generator providing the sin waveform to the amplifier, and the oscilloscope measuring the current through the electromagnet.

Acquiring an electromagnet small enough for this purpose was surprisingly difficult. We used a solenoid, and to implement the electromagnet we ripped a solenoid apart to get the coil, and then used an iron dolly pin as the core of the electromagnet.

Finally, to drive the electromagnet we first used a signal generator to make a 80 Hz sin wave. This sin wave was then fed into the following amplifier circuit to control the current though the electromagnet. The current waveform is provided by an arbitrary waveform generator (Amazon, Koolertron 30 MHz DDS) driving the voltage controlled current source.

4.1 Solving for the Electromagnet 3D position

We use a modified system of equations from the Finexus tracking system to solve for the position of the electromagnet [1].

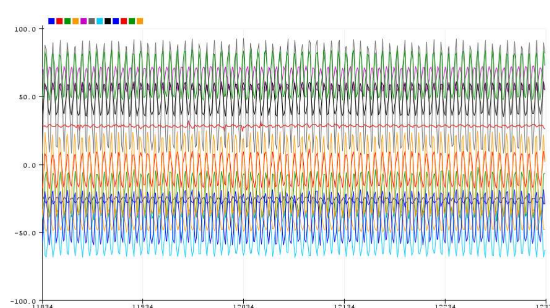

Figure 5: All 12 components of the magnetometers when near an electromagnet oscillating at 80 Hz.

To obtain the magnet position we first acquired 1000 samples of measured data for each magnetometer at a sampling frequency of about 600 Hz. An example f the recorded waveforms is shown in Fig. 5. Each of the xyz components of the magnetometer measurements are analyzed through the Fourier transform, to obtain the amplitude of the in-band field induced by the electromagnet for each sensor. Fig. 6 below shows an example of the Fourier spectra [7] of the recorded signals. For our localization analysis we are only interested in the amplitude at 80 Hz.

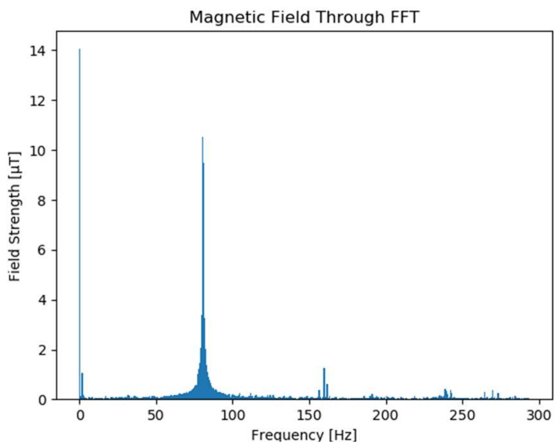

Figure 6: Fourier spectra of a single component of one of the magnetometers. Notice the peaks at 0 Hz and 80 Hz. The 0Hz component is the result of the earth field and the fixed DC biasing field of out current source crcuit.

We record the in-band magnitude of the magnetic field at each sensor. resulting from the electromagnet. We next follow the localization procedure outlined in [1].

Once we obtain $\mathbf{H} = [H_x, H_y, H_z]^T$ which represents the strength of the magnetic field in the x, y, and z directions, we now have to solve for $||\mathbf{H}||^2 = K * r^{-6} *(3\cos^2\theta + 1)$ which is the simplified equation for the total magnetic field strength [1] produced by the electromagnet magnetic dipole. The variable *r* here is the distance from the magnetic field and θ is the tilt angle between the sensor and the magnetometer and K is a constant. However, since this is an under-constraint system with two unknowns and one equation, we need to introduce more equations [1]. We can do this by rewriting *r*, and θ in terms of the magnetometer's 3D position x, y, z. Since we know the relative positions of the four magnets we can get the following equations:

$\mathbf{r_1} = [x^2 + y^2 + z^2]^{1/2}$
$\mathbf{\cos\theta_1} = z / r_1$

$\mathbf{r_2} = [(x - 20.3)^2 + y^2 + z^2]^{1/2}$
$\mathbf{\cos\theta_2} = z / r_2$

$\mathbf{r_3} = [x^2 + (y - 20.3)^2 + z^2]^{1/2}$
$\mathbf{\cos\theta_3} = z / r_3$

$\mathbf{r_4} = [x^2 + y^2 + (z - 21.6)^2]^{1/2}$
$\mathbf{\cos\theta_4} = z / r_4$

Now if we substitute each pair of equations back into $||\mathbf{H}||^2 = K * r^{-6} *(3\cos^2\theta + 1)$ we can get the following over-constraint system:

$||\mathbf{H_0}|| = K * [x^2 + y^2 + z^2]^{-3/2} *(3z / [x^2 + y^2 + z^2]^{1/2})$

$||\mathbf{H_1}|| = K * [(x-20.3)^2 + y^2 + z^2]^{-3/2} *(3z / [(x-20.3)^2 + y^2 + z^2]^{1/2})$

$||\mathbf{H_2}|| = K * [x^2 + (y-20.3)^2 + z^2]^{-3/2} *(3z / [x^2 + (y-20.3)^2 + z^2]^{1/2})$

$||\mathbf{H_3}|| = K * [x^2 + y^2 + (z-21.6)^2]^{-3/2} *(3z / [x^2 + y^2 + (z-21.6)]^{1/2})$

Where K depends on the parameters of the electromagnet [1,5]. We next minimize the error function.

$$\epsilon = \sum_{i=0}^{3} (\|H_i\| - \|M_i\|)^2$$

Where M_i is the measured value for the magnetic field for each sensor, and H_i is the theoretical value. We now have an overdetermined system of nonlinear equations which can be minimized with the python scipy.optimize.minimize function with the Broyden algorithm, which does not require the Jacobian computation [6]. The value of the electromagnet position that minimizes the error is regarded as that best fitting magnet location.

5 Evaluation of Results

To evaluate the accuracy of the finger tracker, we required a way to know the exact position of the electromagnet. To do this we acquired data with an XY table shown in Fig. 7 below (that I had built earlier in my oculometer project!). The electromagnet was mounted to the moving part of the table, and the tracker was mounted above on a sheet of plastic. The electromagnet was then move along the horizontal plane to a preset number of locations on a 5x5 grid using a python stepper motor driver. We recorded the magnetic field signal for each of the four magnetometers for all positions. This data was recorded with the electromagnet about 50mm below the magnetometers.

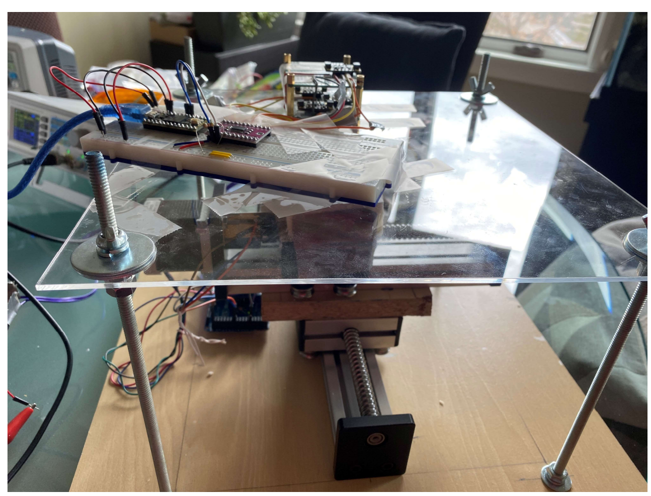

Figure 7: Experimental setup for getting data using an XY table

Next we then used scipy non-linear optimization to solve the equations described in Section 4, giving us an estimated position.

Fig. 8 shows the comparison of the estimated positions compared to the actual ones specified in the XY table scan.

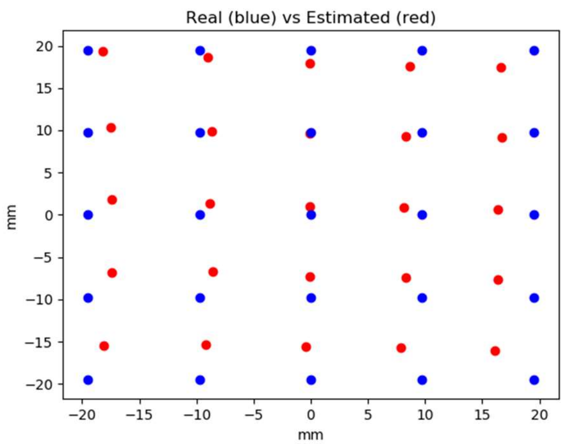

Figure 8: The predicted vs actual position of the electromagnet

The accuracy is reasonably good for a non-optimized system, with mean distance error of 2.6 mm and standard deviation of 1.25 mm. We also note that the error is larger near the edges of the scan. We attribute that to some errors in the determination of the K factor for the electromagnet which should also be an additional fitting parameter as we have incomplete data on the electromagnet characteristics. With time the electromagnet characteristics can be more accurately measured to obtain a more precise position estimation. That is part of our planned future work.

6 Discussion of Benefits and Limitations

The biggest limitation of this approach is that it is susceptible to magnet interference. Even putting a non-magnetized piece of iron too close to the tracker negatively affects the accuracy of the measurements. [4]

Another limitation is the calibration of the magnetometers. This calibration can change based on temperature or if a strong enough magnetic field is applied to the sensor. Therefore, the tracker must be recalibrated often, otherwise the sensors will give bad data.

The biggest benefits of magnetic tracking are that it does not suffer from occlusion, which is especially beneficial when it comes to tracking fingers. Another benefit is that it can be made extremely accurate, with sub milli-meter accuracy if done well.

7 Future Work

The next step for this work would be to combine it with a solution to track the 6-DOF of the hand; as well as to mount the magnetometer tracker and electromagnets to a glove that can be worn. Two viable solutions for the 6-DOF are marker-based tracking, and lighthouse-based tracking.

Another next step would be to acquire faster magnetometers. The biggest weakness of this approach is that the cheap commercial magnetometers IC's on the market run at only 1kHz. Faster magnetometers would enable us to tracking electromagnets running at higher frequencies. This is important because as we increase the frequency of the electromagnet, we decrease the uncertainty of the measurement, greatly improving accuracy.

Lastly we would like to utilize wavelets instead of sinusoid excitation in the future. Wavelets are interesting signals because they allow for rapid identification of specific motions. Hence in the future it will be possible to have direct gesture interpretation built into the hardware.

8 Conclusion

We designed, assembled and tested a system with 4 magnetometers to track the position of an electromagnet beacon driven with alternating current representing the position of a fingertip moving in space. Using this system, we measured the displacement of the electromagnet along a flat surface ~50 mm away from the magnetometers and we were able to track the position inside a 40x40 mm^2 region with about 2.6 mm mean error.

ACKNOWLEDGMENTS

Special thanks to my dad for buying all the parts we needed.

REFERENCES

[1] N. I. Durlach and Anne S. Mavor, *Virtual reality; Scientific and Technological Challenges*, NRC, 1995.

[2] Chen, Ke-Yu & Patel, Shwetak & Keller, Sean. (2016). "Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing," 1504-1514. 10.1145/2858036.2858125.

[3] https://makersportal.com/blog/2018/2/25/python-datalogger-reading-the-serial-output-from-arduino-to-analyze-data-using-pyserial

[4] https://learn.adafruit.com/adafruit-sensorlab-magnetometer-calibration

[5] https://diydrones.com/profiles/blogs/advanced-hard-and-soft-iron-magnetometer-calibration-for-dummies

[6] Juan Nunez-Iglesias, Stéfan van der Walt, and Harriet Dashnow. *Elegant SciPy: Elegant SciPy: The Art of Scientific Python*, O'Reilly Media; 1 edition (August 31, 2017)

[7] https://www.ritchievink.com/blog/2017/04/23/understanding-the-fourier-transform-by-example/