Cyber-Physical Systems (IECE-553) Final Project Report

Tagwa Abdelrahman, Andrew Boggio-Dandry, *IEEE Student Member*

◆

1 INTRODUCTION

The aim of this project was to develop a cyber-physical system to manage the spacing between antennae in an array. Our planned software-controlled system called for 8 antennae to be arranged on linear slide actuators and mounted on plexiglass or a similar surface, with each antenna capable of independent movement. Due to time, cost, and parts limitations, we proceeded to develop a proof-of-concept involving the movement of a single antenna. This allowed for the simulation of a 2-antenna array, by considering a second, virtual linear slide actuator placed back-to-back against the one used in our implementation.

A C-program, running on the Raspberry Pi acting as the system controller, was developed to accept a user-entered transmission frequency. The program computes the optimum antenna spacing for the given frequency ($\lambda/2$) using Equation 1, where c denotes the speed of light and f is the user-entered frequency [1, 2]. The code then directs the stepper motor (connected through a driver circuit to the GPIO pins of the Raspberry Pi controller) to move the antennae to the computed distance. As we are considering a second, virtual linear slide actuator at an angle of 180°, the motor only needs to move the antenna half of the computed distance.

$$\lambda = \frac{c}{f} \tag{1}$$

The distance $\frac{\lambda}{2}$ is desirable as it helps maximize the gain of the antennae working together, while minimizing the interference. By placing the antennae $\frac{\lambda}{2}$ apart from each other, at the voltage nulls, it also reduces the *standing wave* phenomenon [3].

We had planned to verify the distance using ultrasonic sensors, fed back to the Raspberry Pi controller reporting whether the desired distance had been achieved or if small corrections needed to be made. These feedback measurements were planned to continue until the desired distance had been achieved. This sensor was also planned to provide the current antenna distance at program start in order to return the linear slide actuator's carriage to its starting position. Unfortunately, the ultrasonic sensor we worked with was particularly imprecise. It's output readings varied wildly, both while measuring the antenna in motion and at rest.

Tagwa's contributions to the project included modeling the motor and driver circuit, while Andrew modeled the physical system. We then worked together as a team on all other aspects: implementing the hardware setup and control code, and analyzing the test results/collecting data.

2 SYSTEM MODELING

This section will discuss the modeling of the physical system (antenna board and connections), as well as modelling the motor movement through theoretical equations and evaluating based on the motor's datasheet. While our implementation varied from our initial model, the details of our original 8-antenna array modelling are included in case anyone is interested in continuing the project in the future.

2.1 Physical System Model

We had originally planned on using a maximum physical area of $1m \times 1m$. Since lower frequencies require longer spacing, only 2 of the antennae (numbers 1 and 8 – the antennae at each edge) would have been capable of achieving the lower frequency ranges. The challenge we faced was creating a continuous frequency band that the linear array of antennae could operate through. Due to the dimensions of the linear slide actuator we purchased for the project, our frequency range was severely limited.

The proposed physical system design can be seen in Figure 1. As further research and investigation proceeds, the length of antenna tracks number 1 and 8 may need to change to accommodate the desired range of frequencies. We believe it is possible to make the range's floor (antennae 1 and 8) around 400 Mhz ($\frac{\lambda}{2}$ = 37.5cm), and have continuous coverage of the frequency spectrum up to 5 GHz.

2.1.1 Calculating the X-Axis Distance

As illustrated in Figure 1, the x-axis coordinates are known: they correspond to the desired spacing between the antennae based on the user-inputted frequency. We also know the value of θ, B or T, and C (from the ultrasonic sensor) as seen in Figure 2. The desired linear spacing between two antennae can be calculated using the following trapezoid equations, as illustrated in Figure 2.

$$\sin\theta = \frac{opposite}{hypotenuse}$$

$$\sin\theta = \frac{a}{c} \tag{2}$$

• *T. Abdelrahman and A. Boggio-Dandry are with the Department of Electrical and Computer Engineering, SUNY Albany, Albany NY, 12222. E-mail: {tabdelrahman, aboggio-dandry}@albany.edu*

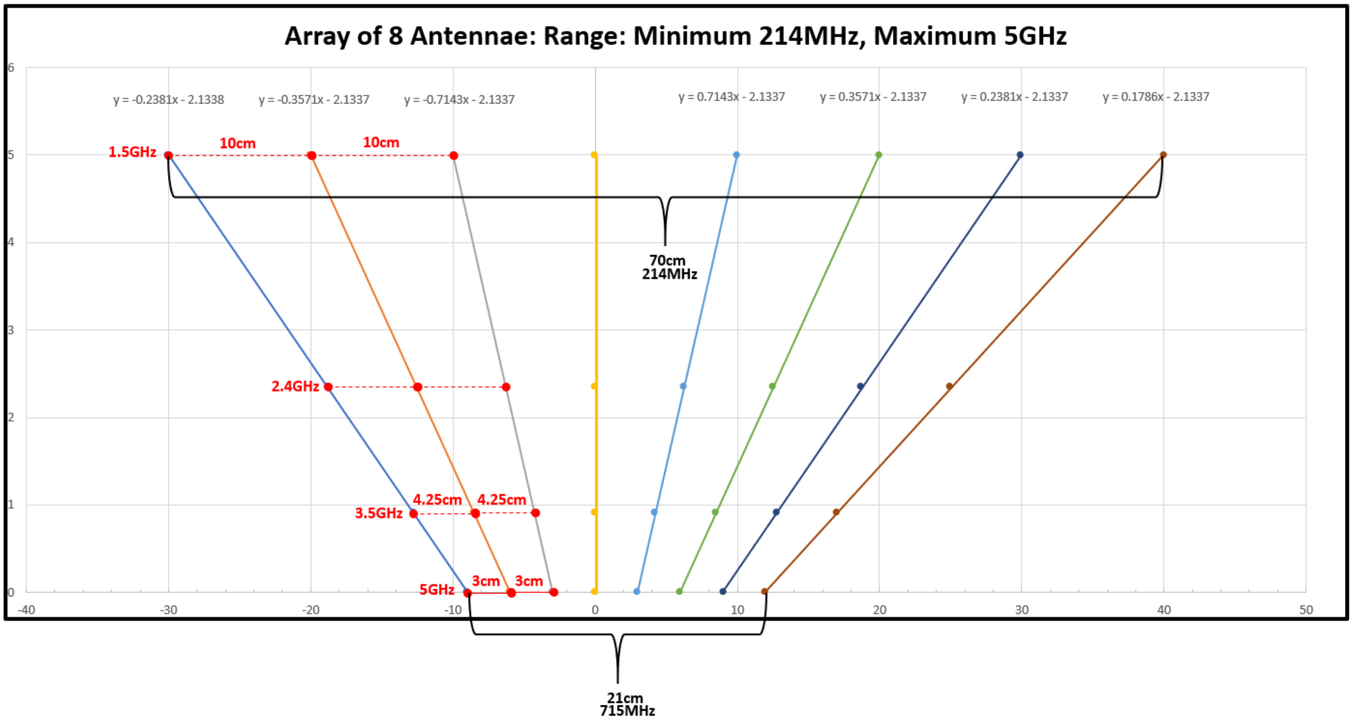


Fig. 1. Depicting the anticipated arrangement of antennae 1 – 8. As seen in the figure, every antenna will have the ability to accommodate any frequency in the 1.5 GHz – 5 GHz range. More investigation is needed on antenna 1 and antenna 8 to expand the range to lower frequencies. The y-axis has been scaled down in this figure, but each antenna track requires its own linear equation.

From the Pythagorean Theorem:

$$c^2 = a^2 + b^2$$

$$b = \sqrt{c^2 - (c \cdot \sin\theta)^2} \tag{3}$$

From the trapezoid:

$$T = B – 2b$$

$$T = B - 2 \cdot \sqrt{c^2 - (c \cdot \sin\theta)^2}$$

$$B = T + 2 \cdot \sqrt{c^2 - (c \cdot \sin\theta)^2} \tag{4}$$

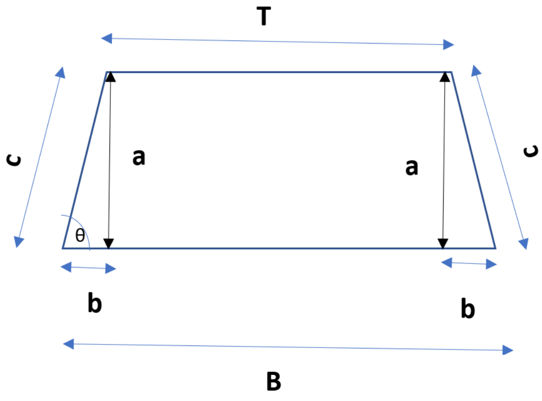


Fig. 2. Illustration of the trapezoid variables used in the preceding equations.

2.2 Motor Modeling

A motor applies torque, or angular force, to a load proportional to the current through the motor winding [4]. If we apply a voltage V to the motor, we expect the voltage and current through the motor to satisfy equation 5, where $w(t)$ represents the angular velocity of the motor, R represents resistance, L represents the inductance of the motor coils, $v(t)$ represents the voltage applied to the motor, and $i(t)$ represents the current. k_b is an empirically determined back-electromagnetic force constant, typically expressed in units of V/RPM (Volts per revolutions per minute) [4].

$$v(t) = R \cdot i(t) + L \cdot \frac{di(t)}{dt} + k_b \cdot w(t) \tag{5}$$

The torque T is proportional to the current flowing through the motor, adjusted by a friction value, and any torque that might be applied by a mechanical load (as seen in Equation 6) where k_t is an empirically determined motor torque constant, η is the kinetic friction of the motor, and τ is the torque applied by the load [4].

$$T(t) = k_T \cdot i(t) - \eta \cdot \omega(t) - \tau(t) \tag{6}$$

As seen in Equation 7, by using the rotational version of Newton's second law $F = ma$ [4], which replaces the force F with torque T, the mass m with moment of inertia I and the acceleration a with angular acceleration $d\omega(t)/dt$.

$$T(t) = I \cdot \frac{d\omega(t)}{dt} \tag{7}$$

Equation 8 is produced by the equality between Equations 6 and 7.

$$I \cdot \frac{d\omega(t)}{dt} = k_T \cdot i(t) - \eta \cdot \omega(t) - \tau(t) \tag{8}$$

From the data sheet of the 42BYGH48 motor into Equation 7:

- $i = 1.2amp$
- $v = 24v$
- $R = 3.2\Omega$
- $L = 2.8$
- $T = 0.45Nm(3.98lb.in)(0.045Kg.m)$
- $I = 68g.cm2(680Kg.m2)$
- $W = 360/1.8 = 200RPM$

$$\frac{d\omega(t)}{dt} = \frac{0.045}{680} = 6.6 \times 10^{-5} \tag{9}$$

To calculate the torque of the load, assume we have:

- maximum weight = 4oz = 0.1133kg
- Distance = 500mm = 0.5m
- $a = \frac{d\omega(t)}{dt} = 6.6 \times 10^{-5}$
- $Torque = force \times distance$
- $Force = ma$

$$\therefore \tau = 0.1133 \cdot 0.5 \cdot 6.6 \times 10^{-5} = (0.0000353N.m) \tag{10}$$

From Equations 6 and 10:

- Assume η=0
- $T(t) = K_T \cdot i(t) - \tau(t)$
- $0.45 = k_T \cdot 1.68 - 0.0000353$
- $k_T = 0.2678N.m/amp$

3 SYSTEM DESIGN

The system we built consists of three major components: sensors, actuators, and a controller (as shown in Figure 3). This section will discusses the linear slide actuator, the use of an ultrasonic sensor to return the plate of the linear actuator to its start point, the DRV8825 driver circuit that was used to drive the motor, and how we controlled them using the Raspberry Pi controller. The simplified block diagram (Figure 4) demonstrates the intended operation of the entire system.

3.1 Linear Slide Actuator

A stepper motor was used to control the movement of the antenna, based on the user-entered frequency. To convert from circular motion to linear motion we are using a leadscrew. The gear revolution per inch will substitute in Equation 11 [5]. We did not need to use the motor's microstepping feature as we observed outstanding accuracy in full-step mode. This gives the constant value of 1.

$$Leadscrew \cdot \Big(\frac{Revolution}{mm}\Big)\Big(\frac{1}{Microstep}\Big) \times Motor \cdot \Big(\frac{Step}{Revolution}\Big) = \frac{Step}{mm} \tag{11}$$

Calculating the revolution per mm for the lead screw (pitch = 4mm):

$$\frac{Revolution}{mm} = \frac{1}{pitch} \tag{12}$$

From the motor's datasheet, the steps per revolution = 200 steps. Inserting into equation 11:

$$\frac{1}{4mm} \cdot 200step = 50\frac{step}{mm} \tag{13}$$

This gives our result of 1 step equal to 0.2mm. The precision, therefore, is 0.02mm.

3.2 DRV8825

The DRV8825 driver circuit has a total of 16 pins available for use. The connections are as follows [6]:

- VMOT and GND MOT: supply motor power in the range of 8.2V to 45V
- M0, M1 and M2: allow 6-step resolution control
- A1, A2, B1, and B2: these pins connect to the motor coils. The A1/B1 and A2/B2 pins do not need a specific positive or negative coil wire
- STEP: controls the motor's steps
- DIR: controls the motor direction (clockwise vs counter-clockwise)
- EN: the enable pin, which is active low. When pulled LOW (logic 0) the DRV8825 driver is enabled
- SLP: the sleep pin, which is active low. When pulled LOW the driver enters sleep mode
- RST: the reset pin, which is active low. When pulled LOW, all STEP inputs are ignored until RST is again pulled HIGH.
- FAULT: drives LOW whenever the H-bridge FETs are disabled as the result of over-current protection or thermal shutdown
- GND: reference ground

In our application, we purchased a 24V power supply to connect to the VMOT pin as our motor calls for 24V in. Because of the high accuracy of the full-step mode of the motor, we are not using microstepping. Therefore, M0, M1, and M2 are unused. Both the STEP and DIR pins are connected to GPIO pins on the Raspberry Pi. Both the SLP and RST pins are connected to the Raspberry Pi's 3.3V out. The GND pin was connected to the Raspberry Pi's GND.

Before using the motor, the maximum current supplied to the stepper motor's coils needed to be limited to prevent it from exceeding the motor's rating. There is a small trimmer potentiometer on the DRV8825 driver circuit that can be used to adjust the supplied current by setting its V_{ref} using the following equation:

$$I_{limit} = \frac{V_{ref}}{2} \tag{14}$$

According to the datasheet, our motor calls for $I = 1.2A$, but the datasheet also cautions to reduce this current to 71% when using the full step mode. As such:

$$V_{ref} = \frac{0.71 \cdot 1.2}{2} = 0.426V \tag{15}$$

The potentiometer is very small, making fine-tuned adjustments difficult, but we adjusted it as close to 0.426V as possible. As the system is now designed back-to-back linear actuators (180°between them), the minimum distance possible is 200mm and the maximum distance possible is 852mm. This allows us to calculate the total frequency range for our system, from 145.8 – 749.5MHz (as shown in Figure 5).

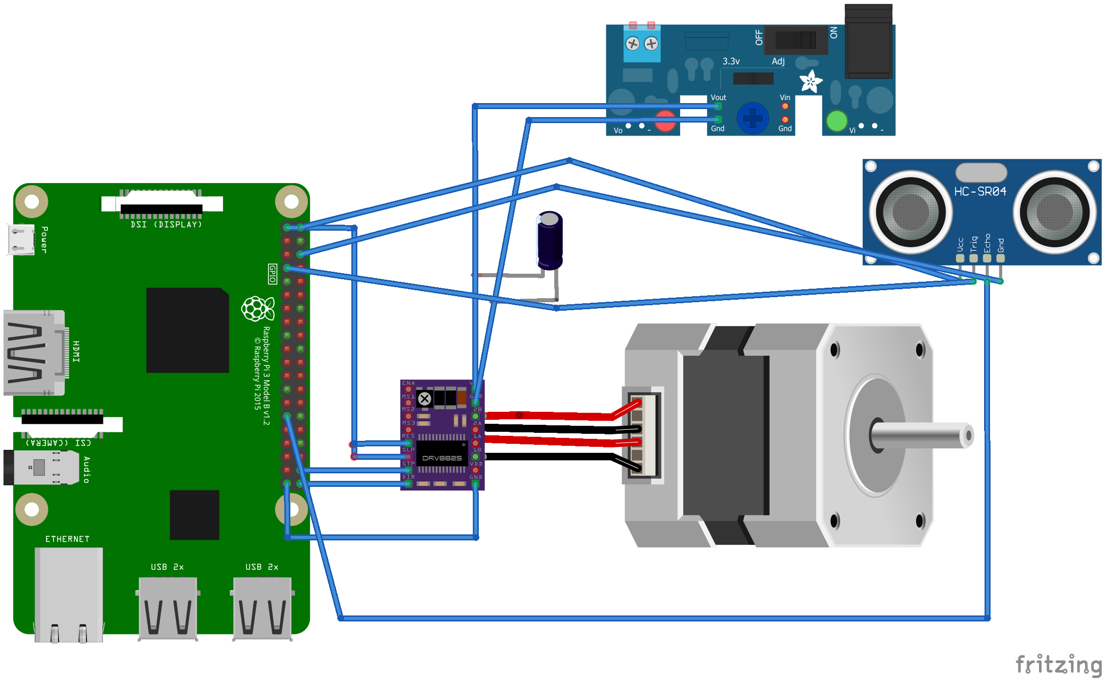


Fig. 3. Wiring (Fritzing) diagram, showing the connections between the Raspberry Pi, breadboard, DRV8825 driver circuit, ultrasonic sensor, the stepper motor, and our 24V power supply. Notice the use of a 100μF capacitor on the 24V in before connecting to the motor.

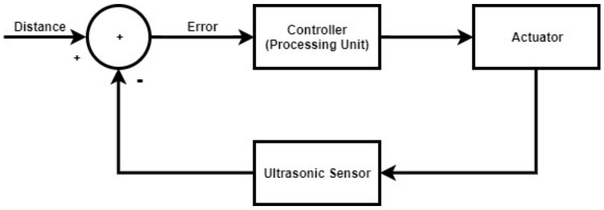


Fig. 4. The simplified block diagram of our system.

3.3 Ultrasonic Sensor

Once motors move the antenna, an ultrasonic sensor is used as a feedback device to the controller to verify that the motors have moved the antenna to the desired position. If not, small corrections will be made until the desired distance has been achieved. Also, after each entry of the frequency from the user, the antenna is driven back to the initial start position using the readings from the ultrasonic sensor.

3.4 Raspberry Pi

A Raspberry Pi Model 3 was used as the controller for the system, to which all sensors and actuators were connected, and from which the control program operates as shown in Figure 3.

3.5 Implementation

Now that each system component has been described, the "big picture" of how the entire system is controlled will be illustrated. As shown in Figure 3, all components were connected using a breadboard to the Raspberry Pi controller. The software created for this project running on the controller (written in C, and included with the project submission) first asks the user for the desired frequency. The program checks if our system is capable of providing the frequency or not. If the desired frequency is indeed in the system's supported range, the corresponding distance is calculated using Equation 1. The distance between the two back-to-back linear actuators (200mm) is then subtracted, allowing for the calculation of the number of steps required. The number of steps and the direction (counter-clockwise to move the antenna forward, clockwise to move the antenna back toward the starting position) is then sent to the DRV8825 driver circuit to move the stepper motor. Once complete, the achieved distance is verified using the ultrasonic sensor, and any necessary corrections needed are implemented.

4 DATA ANALYSIS AND CONCLUSIONS

Once the system was built, tested, and operational, we were able to perform a series of trials. We selected 10 frequencies from 250MHz to 700MHz (within the range our system is capable of handling), separated by 50MHz. We performed 3 trials for each frequency to better estimate the system's accuracy. Our results, shown in Figure 6, show that the accuracy To do the analysis we took multiple values of the moved distances and compare it to the actual distances as shown in figure 6 to determine the accuracy of the system. We then graphed the distances measured in the first trial compared to the expected distances, seen in Figure 7. Finally, we calculated the error from each trial and plotted them in Figure 8.

It was unfortunate that we were not able to use the ultrasonic sensor to assist in verifying the veracity of the data. Since its measurements were so unreliable, we took the measurements physically (using a ruler). As such, due to the limited precision of the ruler, it's possible the system was even more accurate than described in the graphs. This project was a fun and educational experience, and helped drive home many of the concepts discussed in the CPS course this semester.

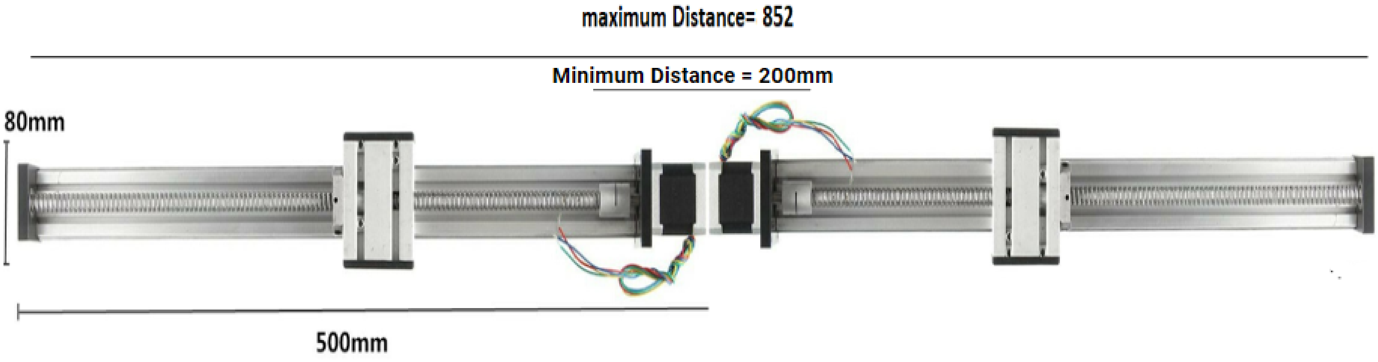
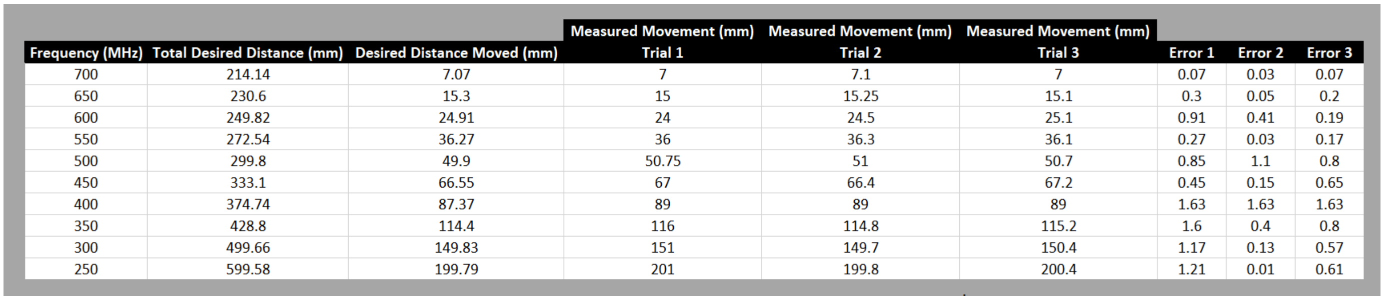


Fig. 5. This figure is showing the system with the two linear slide actuators back to back , minimum and maximum distance.



Frequency (MHz)	Total Desired Distance (mm)	Desired Distance Moved (mm)	Measured Movement (mm) Trial 1	Measured Movement (mm) Trial 2	Measured Movement (mm) Trial 3	Error 1	Error 2	Error 3
700	214.14	7.07	7	7.1	7	0.07	0.03	0.07
650	230.6	15.3	15	15.25	15.1	0.3	0.05	0.2
600	249.82	24.91	24	24.5	25.1	0.91	0.41	0.19
550	272.54	36.27	36	36.3	36.1	0.27	0.03	0.17
500	299.8	49.9	50.75	51	50.7	0.85	1.1	0.8
450	333.1	66.55	67	66.4	67.2	0.45	0.15	0.65
400	374.74	87.37	89	89	89	1.63	1.63	1.63
350	428.8	114.4	116	114.8	115.2	1.6	0.4	0.8
300	499.66	149.83	151	149.7	150.4	1.17	0.13	0.57
250	599.58	199.79	201	199.8	200.4	1.21	0.01	0.61

Fig. 6. Data collected from frequency range 250MHz – 700MHz in 50MHz steps. The trials were completed 3 times to get a better sense of accuracy.

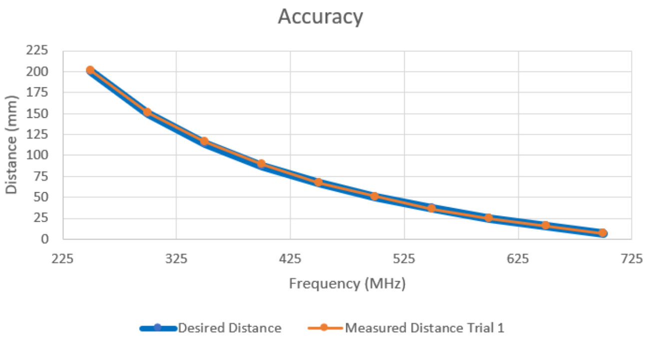


Fig. 7. The relation between the desired distance and the measured distance.

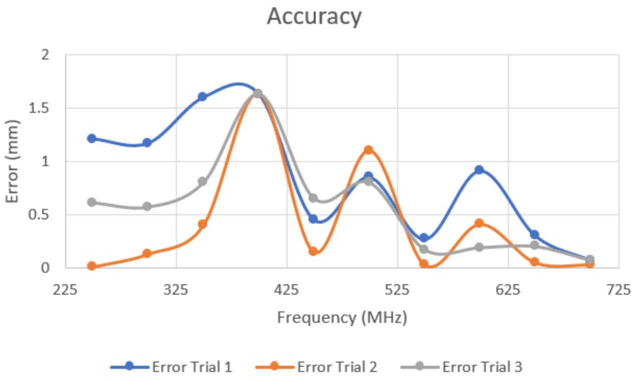


Fig. 8. The difference (error) between the desired distance and the measured distance.

5 RECOMMENDATIONS FOR FUTURE WORK

- Implement the system in its planned entirety, including all 8 antennae
- Update the software in order to control 8 linear actuators, DRV8825 driver circuits, and ultrasonic sensors
- Investigate other driver circuits capable of driving multiple antennae
- Research a more accurate distance measurement sensor for antenna adjustment
- Find a better linear actuator for this project with a smaller plate, or build one, to be able to provide a larger frequency range

REFERENCES

[1] D. Doody and G. Stephan, *Basics of Space Flight Learners' Workbook*. The Planetary Society, 1995.
[2] W.-Y. Lee, "Antenna spacing requirement for a mobile radio base-station diversity," *Bell System Technical Journal*, vol. 50, no. 6, pp. 1859–1876, 1971.
[3] A. W. Rudge, K. Milne, A. D. Olver, and P. Knight, *The handbook of antenna design*. Iet, 1982, no. 15.
[4] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*, 2nd ed. MIT Press, 2017.
[5] "Stepper Motor Calculations," https://wiki.cnc.xyz/Stepper_Motor_Calculations, accessed: 2019-10-22.
[6] "Interface DRV8825 Stepper Motor Driver Module," https://lastminuteengineers.com/drv8825-stepper-motor-driver-arduino-tutorial/, accessed: 2019-12-2.