

High Resolution Object Triangulation Using Ultrasound Sensors

Nada Alqaderi¹, Anthony Bastidas², Madison Desormeau³, Aaron Friedland⁴,
Kelsey Johnson⁵, Ethan Laba⁶, Dean M. Aslam⁷

¹⁻⁷Electrical and Computer Engineering, Michigan State University, E. Lansing, MI 48824

Abstract: Object triangulation has become a vital part of many systems and applications. As different application areas develop, the need for more refined object triangulation methods has increased. This paper aims to utilize on-chip transducers that transmit and receive a signal of a small object in a precise way. These transducers are controlled by a python code and are mounted on a working surface area which can be moved to wherever it may be needed. The sensors will be able to triangulate an object's position within a three-dimensional plane, as well as communicate with each other to calculate the object's position more accurately. These transducers are wired to a computer system which displays the recorded data. A similar system could then be implemented in various industries to increase safety or performance.

INTRODUCTION:

Object triangulation, defined in Fig. 1 [1], has become a vital part of many systems and applications. As different application areas develop, the need for more refined object triangulation methods has increased [2] [3]. An example of prior work related this paper is shown in Fig. 2 [4]. The earlier 3D printed model resulted in inaccurate measurements, as the sensors were not able to communicate properly with another sensor. This resulted in calibration issues, as the sensors were inaccurately measuring distances.

This paper uses on-chip transducers that transmit and receive a signal of a small object in a precise way. These transducers are controlled by a python code and are mounted on a working surface area which can be moved to wherever it may be needed. The sensors will be able to triangulate an object's position within a three-dimensional plane, as well as communicate with each other to calculate the object's position more accurately. These transducers are wired to a computer system which displays the recorded data. A similar system could then be implemented in various industries to increase safety or performance.

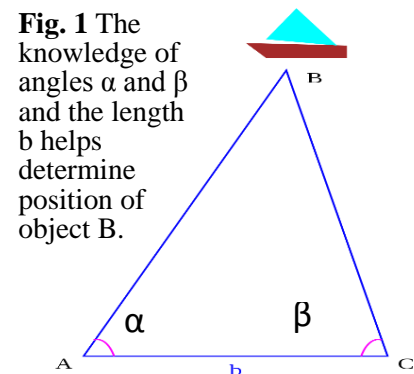


Fig. 1 The knowledge of angles α and β and the length b helps determine position of object B.

DESIGN CONSTRAINTS:

Upon the creation of the new module design, the main constraints for the design depend on design goals. The sensors were required to be able to accurately track an object between the distances of 6 in and 1 m. The object size needed to track is 2.5"x2.5"x2.5". The working surface area of the module could be no greater than 1 sq. ft. to ensure it was small enough to be able to fit into most applicable systems. The design needed to be powered by a 12 V battery, which would allow the system to be placed in vehicles containing a typical car battery. Additionally, the collected location data needed to be processed and displayed on an x-y graph in real time.

DESIGN RANKINGS AND TRADE-OFFS:

The authors considered the previous design of the 3D printed module and two newly 3D printed design models as options to hold the sensors that would be able to track an object's distance. The first model considered is shown in Figure 2. The second and third models are shown in Figure 3. The biggest difference between the three models is the size and how the sensor is being mounted onto the model. The first design consists of a set center distance from the center to each sensor, but the way the model was designed caused vibrations to have the sensors give off inaccurate measurements. The second design allows the center distance to be changed to what the user may want and consists of a smaller module compared to the first one. The third design contains a similar design to the second, but without the sliding feature, and it is also smallest of the three.

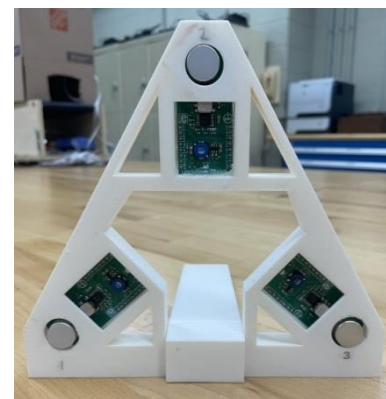


Fig. 2 Prior work [4].

Comparing the three models, the third design allowed the sensors to be held in place within the module from a non-changeable center distance. It also resulted in the sensors being placed within the module without interference from any vibrations, causing the results to be read to be accurate compared to the third.

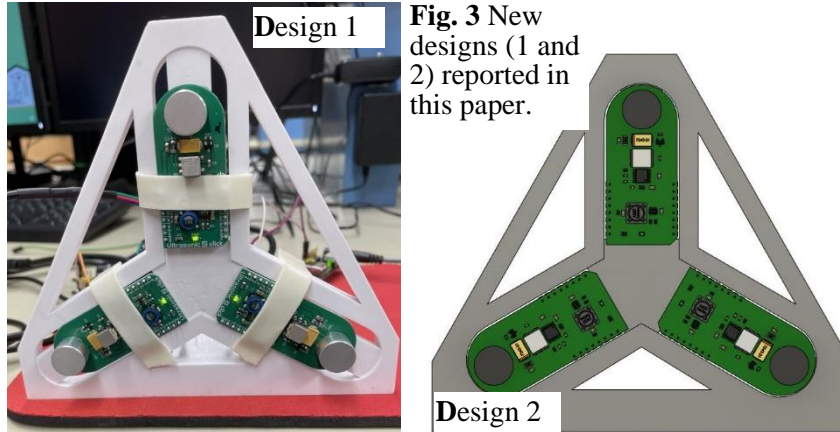


Fig. 3 New designs (1 and 2) reported in this paper.

PROPOSED DESIGN CONCEPT:

The authors used ultrasonic sensors, an Arduino board (seen in Fig. 3), and a 3D printed model to design a specific solution. The design revolves around using three sensors connected to an Arduino, which will then allow for data to be transmitted to a computer to allow to be processed and visualize a plot to display. This will then allow for orientation measurements and 3D positioning to be shown between the sensor and the object. This will be compatible with an autonomous vehicle, which will allow object locations surrounding the car to be transmitted and create a virtual mapping system using a triangulation system.



Fig. 4 Pin layout for PGA460-Q1.

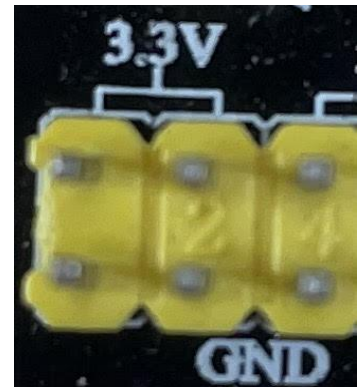


Fig. 5 Power supply pins on transformer.

RESEARCH:

The purpose of this project was to design a triangulation navigation system that detects objects and reports the distance from the center of the module to the object. To report this distance, the sensors should read in the correct distance from themselves to the object. Then, triangulation mathematics should use the distance from each sensor and the sensor placement to locate the object with respect to the center of the module. Knowing all this, research was conducted to learn more about sensor operation and triangulation mathematics.

		MIN	MAX	UNIT	
V _{PWR_XF}	Supply input for transformer topology	6	15	V	
V _{PWR_DD}	Supply input for direct drive topology	6	28	V	
V _{ID}	IO pin	-0.1	V _{PWR}	V	
V _{INX}	Transducer receive input	-0.1	0.9	V	
V _{DIG_IO}	Digital I/O pins	-0.1	V _{IOREG}	V	
V _{GND}	Ground pins	-0.1	0.1	V	
I _{LPM}	V _{PWR} Input current	Low Power Mode Enabled		500	μA
I _{BURST}		During Ultrasonic Burst		500	mA
T _A	Operating free-air temperature	-40	105	°C	
T _J	Operating junction temperature	-40	125	°C	

Fig. 6 Sensor limits for voltages, currents, and temperature [5] .

With provided sensors to work with (TI-PGA460-Q1), the data sheet was studied [5] to provide insight into the sensor schematic, layout, and code. To run the module, the sensors must connect to a computer so that they can communicate with one another. Using the pin layout shown in Figure 4, the transmission and receiver pins were found and connected to the computer using USB serial cables. Next, the sensors must be connected to a power supply to start up. Again, using the pin layout in Figure 4, the 3V3 and ground pins were found and used to connect power to the sensors. Transformers were needed to transform the 12V from the battery into a sensor acceptable power source of 3.3V. Therefore, the sensor

power and ground pins must be connected to the 3.3V and ground pins of the transformer (shown in Figure 5), and the transformer must be connected to the power supply using a power supply cable.

Now that all the connections were researched, the sensor data sheet was studied to find the limits, functionality, calculations, and pseudocode for the sensors being used. The limits for different voltages, currents, and temperatures the sensor used are shown in Figure 6.

These limits were considered when the code was written so it was known how high or low a certain parameter could be without breaking a sensor.

Figure 7 displays how the sensor calculates the distance of an object using the speed of sound. A transducer acts as a microphone and can send and receive ultrasonic sounds. As sound is generated, a temperature sensor is used to equip correct measurements. Since temperature fluctuation affects the speed of a sound wave, if a temperature sensor was not included, as temperature increases, sound waves travel faster to and from the object. This makes the object seem closer than it is. Once sound waves are generated and sent out, if there is an object within a certain distance, the sound waves will reflect off the object and bounce back toward the sensor.

The transducer will then be able to receive this ultrasonic sound back. Shown in Figure 8 is a more in-depth visualization of how the ultrasonic burst is sent out. The sensor transmits the ultrasonic burst and provides a pulse that corresponds with the time that an echo is returned to the sensor. By measuring the echo pulse received, a distance to an object can be calculated.

One common issue with these sensors is accepting random noise as the distance signal. To tackle this problem, the signal magnitude thresholds of the sensor should be changed based on the amount of noise a specific sensor picks up. Since random noise signals are normally signals that bounce off farther objects before being picked up, they tend to come back with a larger sound wavelength, and therefore, a larger distance. This sensor reads in a larger sound wavelength as a smaller amplitude on its plots, so raising the thresholds on the signal magnitude graph will eliminate much unwanted noise. An example of threshold setting is shown in Figure 9.

To write to the sensor to turn it on and tell it to send ultrasonic bursts, example 3 of Figure 10 was used. These preset hexadecimal values tell exactly what to send to the sensor to turn it on and what to send to make it burst. These hexadecimal values are used in a create burst function of the code created for this project.

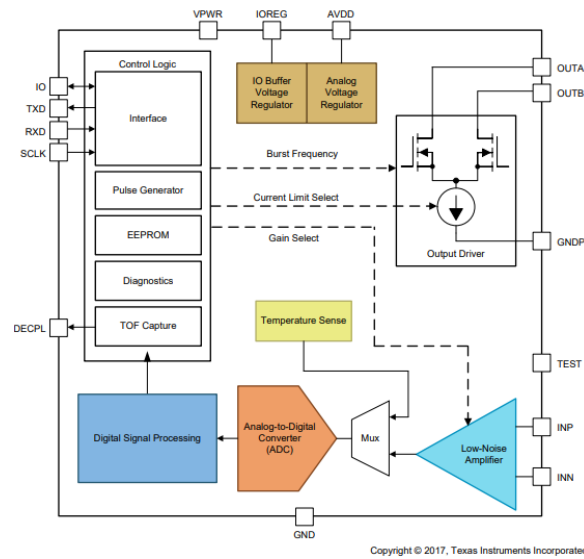


Fig. 7 Functional block diagram for sensor [5].

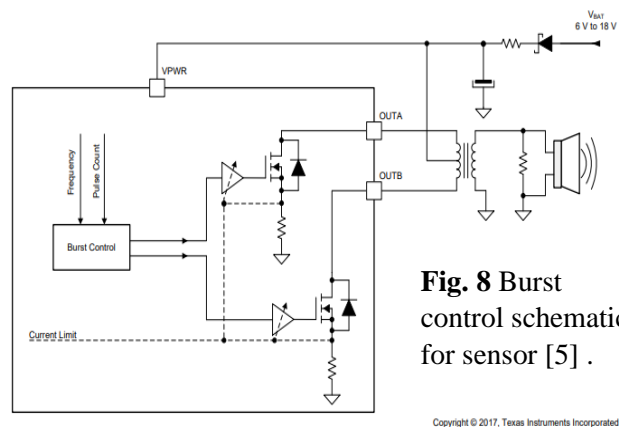


Fig. 8 Burst control schematic for sensor [5].

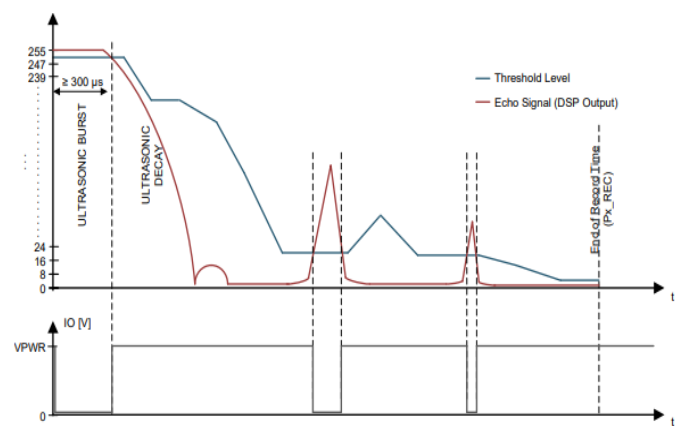


Fig. 9 Threshold examples and limits [5].

Lastly, in the data sheet, it provided calculations to convert the sound byte list provided by the sensor to a physical distance in meters, as shown in Figure 11.

After studying the data sheet, triangulation mathematics was studied to determine the most efficient way to triangulate on a point using three sensors. Triangulation, which is also called trilateration, is a strategy used to locate an object based on three other measurements to that object. After calculating the distance from each sensor to the object, three equations like the one shown in Figure 12 should be created

```

Example1 – Read register 0x1B, where PGA460-Q1 address is 0x0:
Master to PGA460-Q1: 0x55, 0x09, 0x1B, 0xDB ...
PGA460-Q1 to master: ... 0xdiag, 0xdata, 0xchecksum

Example2 – Write register 0x40, data 0x80, where PGA460-Q1 address is 0x0:
Master to PGA460-Q1: 0x55, 0x0A, 0x40, 0x80, 0x35
PGA460-Q1 to master: No response, idle (0xFF)

Example3 – Execute command 0 (Burst/Listen Preset1) to detect 1 object, where PGA460-Q1 address is 0x0:
Master to PGA460-Q1: 0x55, 0x00, 0x01, 0xFE
PGA460-Q1 to master: No response, idle (0xFF)

Example4 – Execute command 5 (ultrasonic measurement result), where PGA460-Q1 address is 0x0, assuming previous execution of Example3 where the master has commanded PGA460-Q1 to search for one object:
Master to PGA460-Q1: 0x55, 0x05, 0xFA ...
PGA460-Q1 to master: ... 0xdiag, 0xtime_of_flight_in_us_[MSB],
0xtime_of_flight_in_us_[LSB], 0xtime_object_width_in_us, 0xpeak_amplitude_in_LSB, 0xchecksum
    
```

Fig. 10 Example read/write commands that are sent/received from the sensor [5] .

(2) To convert the object in time-of-flight in microseconds to distance in meters: $distance (m) = [343m/s \times (MSB \ll 8 + LSB) + 2 \times 1\mu s] - (50\mu s \times 343m/s)$.

Fig. 11 How to calculate distance from sound bites received from sensor [5] .

using the x and y components of each sensor as X_a and Y_a and the distance as the radius r .

$$r^2 = (x - x_a)^2 + (y - y_a)^2 + (z - z_a)^2$$

After creating three equations (one for each sensor), a simple system of equations should be solved to solve for the x, y, and z components of the object with respect to the center of the module. This process is explained similarly in the flow chart shown in Figure 13.

Fig. 12 Triangulation equation for one sensor [6] .

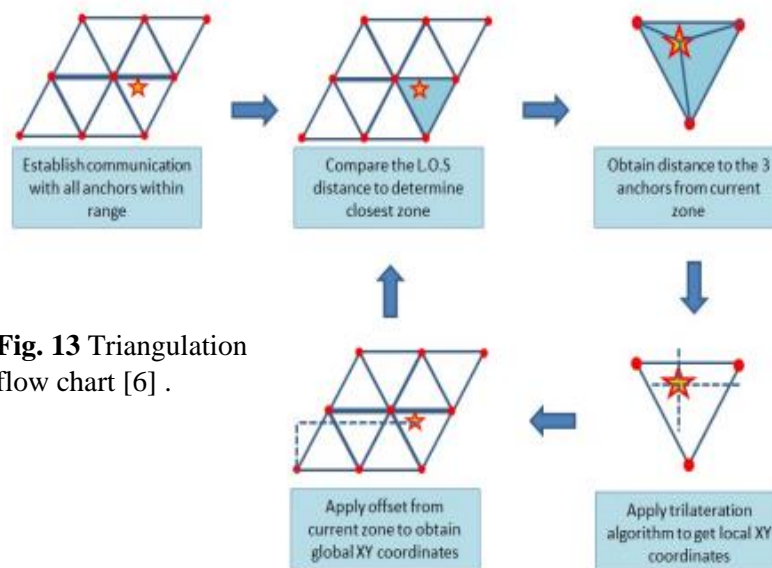


Fig. 13 Triangulation flow chart [6] .

HARDWARE DESIGN:

This project is split into hardware and software but will split again between electrical and non-electrical hardware. The electrical part consists of adding a step-down voltage convert, allowing the device to receive a 12V DC source and operate on a voltage of 3.3V. The sensors then connect to a laptop that controls the sensors. The non-electrical hardware consists of the 3D printed triangular holder in which the three ultrasound sensors are mounted on.

The previous had a grave issue with the plastic touching the sensors, creating vibrations which ruined the sensor's readings. The second holder, shown in Figure 14 implemented a "slide" function, where the investigators could place the sensors at varying distances from the center to find where they function best at while abiding by the sponsor's range requirements. After the investigators found this distance, approximately 6.49cm from center to (each) sensor, the third holder, shown in Figure 15, was designed to lock the sensor chips in those spots where they were found to function the best at. The slide function was also removed, reducing the size



Fig. 14 Slide Design (shown on Left), **Fig. 15** Final Slot Design with Chip (shown on Right)

and material required to print. An overall improvement from the previous design was fixing the rubber cement placement which is an adhesive that holds the sensor chip onto the 3D printed holder. The PCB chip has internal ultrasound reflections while the chip is operating, but if the edges of the chip are touching the plastic, then it will vibrate the chip and ruin the sensor readings. The new innovative design gives optimal tolerances to fit the chip, using rubber cement, without any interference to the sensors. The second and the final designs use Polylactic Acid (PLA) plastic. Although it is brittle compared to other plastics, it is the most inexpensive option for current research [7].

SOFTWARE DESIGN:

The researchers were provided with code from a previous project. In this code, the correct sensor startup code, set limits code, burst code, and checksum code were provided. However, insufficient error checking readability, and power/threshold levels were provided. Also, incorrect distance calculations, triangulation mathematics, and plot function were available. Lastly, no calibration function was provided.

The researchers first implemented a calibration function (Figure 16) to help reduce error in the sensors. This calibration function needs an object to be placed one meter away, calculates the distance, and subtracts one from it to calculate the error. Then, in the main function, the error is used to adjust all subsequent distances. implemented A triangulation math function (Figure 17) was provided. This function used the equations from the trilateration algorithm discussed earlier to calculate the x, y, and z components of the object with respect to the center of the module. A plot function was then implemented to plot the x vs. y component continuously. Lastly, the previously provided distance function was adjusted to calculate the correct distance from each sensor to the object.

After the implementation all these functions, the main function was rebuilt based on the new functions and rigorously tested. Then, the thresholds of each sensor, the current limit, and the number of pulses were adjusted to minimize error of the distance measurements. Once the measurements were reading correctly, error checking was developed for edge cases such as sensors not detecting the object, triangulation failing due to sensor error, and sensors that stop working entirely. Once error checking was implemented, the code was converted from test mode to user mode, with a GUI being created to assist with user accessibility. Lastly, comments and docstrings were added to the main and all functions to provide code readability.

```
def calibration(sensor):
    """
    Calibrates each sensor by finding the error in
    the distance reading when an object is placed
    exactly one meter away; returns error if object
    detected and returns "no object detected" otherwise.
    """
    data = create_burst(sensor)
    c = calc_c(data, True)
    if not c:
        return "no object detected"
    return (c - 1)
```

Fig. 16 Code to calibrate sensors.

```
def x_y_components(x_list, y_list, d_list):
    """
    Uses triangulation math to solve for x,y,z components
    of the point that the three sensors triangulated on. It
    returns the three components. The x component returns as
    "failed" if triangulation failed.
    """
    a2 = d_list[1]**2-d_list[0]**2+x_list[0]**2-x_list[1]**2+y_list[0]**2-y_list[1]**2
    a3 = d_list[2]**2-d_list[0]**2+x_list[0]**2-x_list[2]**2+y_list[0]**2-y_list[2]**2
    x_diff2 = x_list[0]-x_list[1]
    x_diff3 = x_list[0]-x_list[2]
    y_diff2 = y_list[0]-y_list[1]
    y_diff3 = y_list[0]-y_list[2]

    x = (a2/(2*x_diff2)) - ((y_diff2*(a3-(x_diff3/x_diff2)*a2))/(2*x_diff2*(y_diff3-(y_diff2*(x_diff3/x_diff2))))))
    y = ((a3-(x_diff3/x_diff2)*a2)/(2*(y_diff3-(y_diff2*(x_diff3/x_diff2))))))
    try:
        z = math.sqrt(d_list[0]**2-x_list[0]**2-y_list[0]**2-x**2-y**2+2*x_list[0]*x+2*y_list[0]*y)
        x = round(x,3)
        y = round(y,3)
        z = round(z,3)
        return x,y,z
    except ValueError:
        print("Triangulation Failed")
        return "failed", "", ""
```

Fig. 17 Code to triangulate.

The only issue that is still arising in the project is the sensors not reading in the correct signal consistently. While lowering current limit levels, lowering the number of pulses sent, and raising threshold levels all helped the sensors read more consistently, they are still not reading correctly 100% of the time.

GRAPHICAL USER INTERFACE DESIGN:

A Graphical User Interface (GUI) is used to create a more user-friendly interface to run the object triangulation code. After starting the GUI, a window appears that allows a user to start the code by pressing a button, rather than running the code in an interpreter. This is shown in Figure 18. Currently, the graph showing the object location does not appear within the GUI, but through further work this addition can be made to create a fully complete interface.

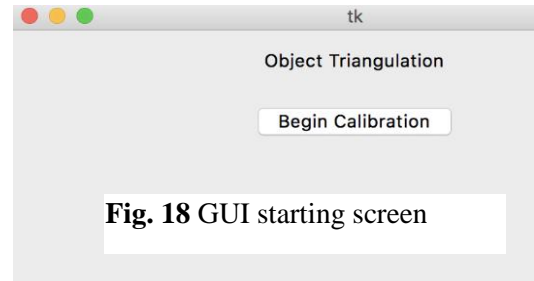


Fig. 18 GUI starting screen

RESULTS AND CONCLUSIONS:

The results of the three designs in order went from non-functional, to varied accuracy, to somewhat accurate. The first did not allow two of the sensors to read properly due to plastic contact interference. The PCB material of the sensor chips has internal ultrasonic reflections from the ultrasonic node in which solid contact with the plastic will interfere with those reflections, and those interfered reflections will interfere with the sensor chip's distance readings. The second design functions perfectly, with acceptable accuracy, when the sensors are placed in the optimized placements. It will not be as accurate for smaller objects. The point of the second design was to find the best distance from sensor node to the center of the holder while being able to physically change that distance using the slide function of the design. That objective was accomplished at ~6.5 centimeters from the center of the holder to the center of the ultrasonic sensor node. The third design's main difference from the second is switching from a slide mount to a slot mount. The slide mount helped find the optimized placement for the sensors, so the third design will lock the sensor chips in place in a slot. This reduces the size and material of the holder while also looking more presentable with exact fits rather than the slide mount which has already served its purpose and simply takes up space. The 3D printed holder's design process was one of the most important parts of this project as it sets the environment to allow the sensors to fulfill the customer requirements.

FURTHER RESEARCH:

The sensors currently used are overly sensitive and are prone to picking up reflections from the module they are mounted on. High sensitivity can assist in better accuracy for detecting objects, but in this case, it creates a hinderance as the sensors are detecting themselves and the object, they are mounted to rather than an object farther away. Utilizing different sensors and an on-chip system could potentially improve future design. There are very diverse applications for triangulation [8] [9] [10] [11] [12] to consider.

ACKNOWLEDGEMENT:

Dr. Prem Chahal provided the idea and insight into sensor capability and performance. **Aditya Purandare** and **Ethan Kepros** demonstrated earlier work. **Brian Wright** provided technical help.

REFERENCES

- [1] <https://en.wikipedia.org/wiki/Triangulation>
- [2] James A. Rodger, "An expert system gap analysis and empirical triangulation of individual differences, interventions, and information technology applications in alertness of railroad workers", *Expert Systems with Applications*, vol. 144, 113081, 2020; <https://doi.org/10.1016/j.eswa.2019.113081>
- [3] V. I. Moreno-Oliva, O. Flores-Diaz, E. Román-Hernández, M. Campos-García, E. Campos-Mercado, J. R. Dorrego-Portela, Q. Hernandez-Escobedo, J. A. Franco, A-J. Perea-Moreno, and A. A. García, "Vibration Measurement Using Laser Triangulation for Applications in Wind Turbine Blades", *Symmetry*, 13(6), 1017, 2021; <https://doi.org/10.3390/sym13061017>
- [4] Team members, ECE480 project in Spring 2021, Michigan State University.
- [5] PGA460-Q1," PGA460-Q1 data sheet, product information and support; <https://www.ti.com/product/PGA460-Q1>.
- [6] M. V. Pablo Cotera, "Indoor robot positioning using an enhanced trilateration algorithm - Pablo Cotera, Miguel Velazquez, David Cruz, Luis Medina, Manuel Bandala, 2016," SAGE



- [7] <https://www.creativemechanisms.com/blog/learn-about-polylactic-acid-pla-prototypes>.
- [8] Viktor Boström, "Positioning and tracking using image recognition and triangulation"; <https://www.diva-portal.org/smash/get/diva2:1576987/FULLTEXT01.pdf>
- [9] J. R. Bartels, J. Wang, W. "Red" Whittake, and S. G. Narasimhan, "Agile Depth Sensing Using Triangulation Light Curtains", 2019; https://openaccess.thecvf.com/content_ICCV_2019/papers/Bartels_Agile_Depth_Sensing_Using_Triangulation_Light_Curtains_ICCV_2019_paper.pdf
- [10] Z. Pan, T. Lang, C. Li1, M. Di1, G. Chen, Y. Kalay, R. Pai, and A. Wang "Visible Light Communication Cyber-Physical Systems-on Chip for Smart Cities", J. Communications, Vol (14), 12, 1141-1146, 2019.
- [11] Brian G. Kilberg; Felipe Mulinari Rocha Campos; Filip Maksimovic; Thomas Watteyne; Kristofer S. J., "Accurate 3D Lighthouse Localization of a Low-Power Crystal-Free Single-Chip Mote", IEEE J. Microelectromech. Systems, vol. 29(5), 818 – 824, 2020; doi: 10.1109/JMEMS.2020.3011460
- [12] P. Motreuil-Ragot et al., "Enabling actuation and sensing in organs-on-chip using electroactive polymers," 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft), 2020, pp. 530-535, doi: 10.1109/RoboSoft48309.2020.9115977