Robust Multi-Agent Aquatic Localization

Vivek Sridhar (ECE ‘17), Vasu Agrawal (ECE ‘17), Kim Kleiven (SCS ’17), Tommy Lau (SCS ’17)

Abstract

Multi-agent localization is an increasingly popular research problem in robotics and embedded systems. There are challenges both in localization algorithms and in communication between multiple agents in a system. In this research project, we aim to improve the robustness of multi-agent localization systems by investigating wireless localization through different mediums, particularly water. Water poses a number of unique communications challenges, particularly in the disruption of communications frequencies, a drawback that does not exist to nearly the same extent in communication over air. By analyzing and developing a wireless localization system that is robust to communication disruptions and to communicating through an obtrusive medium, we hope to dramatically improve indoor localization systems both with respect to air to air systems and localization in other mediums.

Challenges

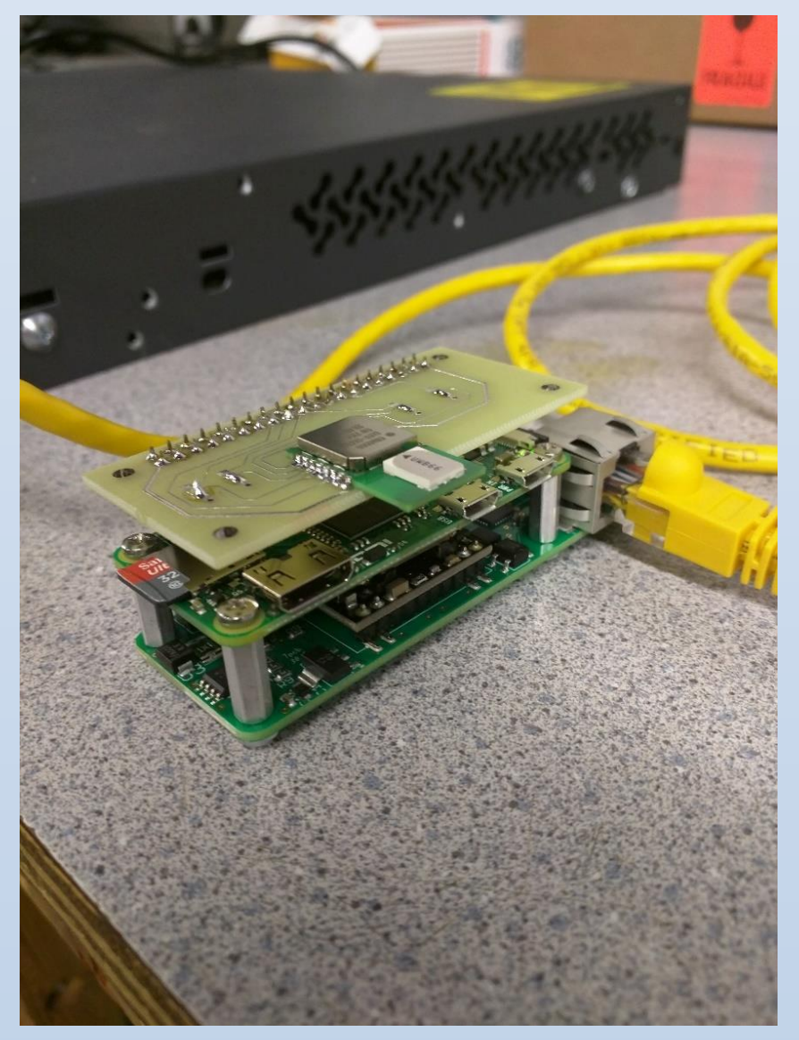
One of the most significant challenges of indoor localization in an aquatic environment is the presence of water acting as interference to radio frequency communication. Particularly, we had concerns about the water reflecting communication between our devices, resulting in packet loss and missing measurements.

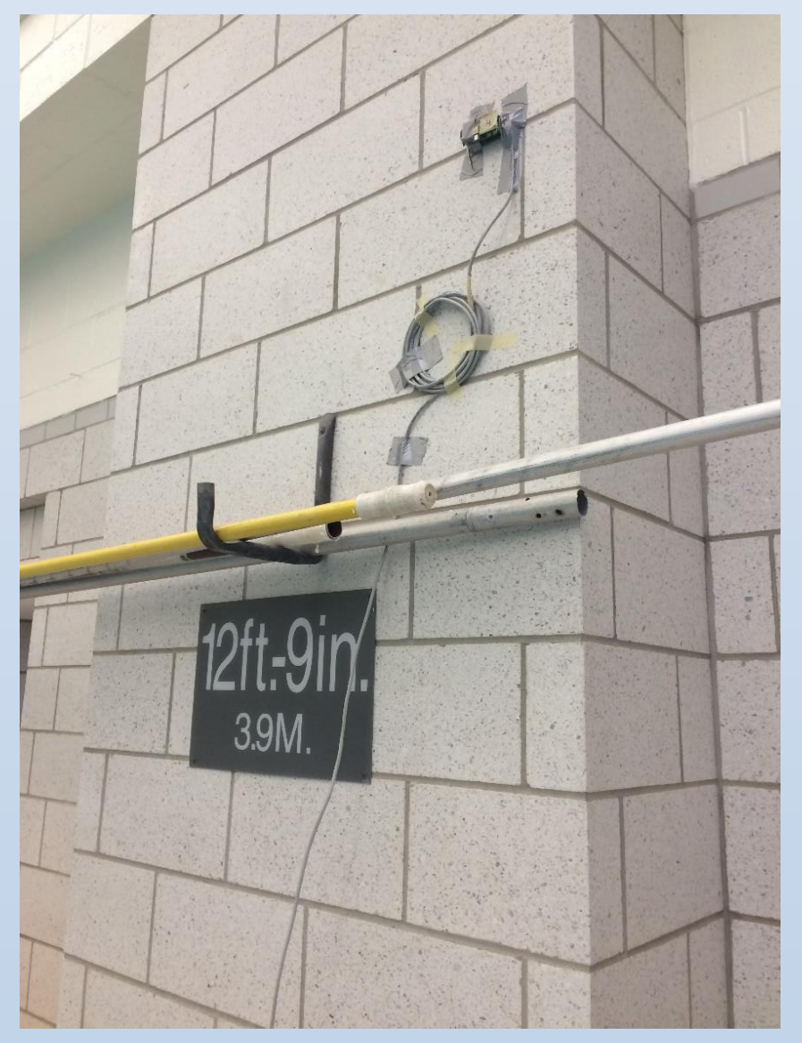
Another significant challenge was creating a robust localization system that was able to localize in any environment without prior calibration required in the setup procedure. We wanted our algorithm to make no assumptions about the state of the system, and compute relative positions simply from the measurements received from each device. In particular, we did not want to encode any global positions of beacons, as this would involve measurement and setup calibration, which we tried to avoid. We also wanted the system to be robust to handling an arbitrary network size known only at runtime.

Technology

The localization chip we used was the Decawave DWM1000 module, a chip that performs precisely timestamped messaging on the Ultra-Wideband spectrum. The Decawave chips have a range of up to 300 meters, which lets us place a large number of beacons in any given active work environment to provide redundant measurements for our trilateration algorithm.

To construct the beacons, we used a power-over-Ethernet board for a Raspberry Pi Zero, and spun up a custom breakout for the Decawave chip. This resulted in a small form-factor beacon that only required one line for power and data, as well as providing a robust communication protocol (Ethernet) to our localization server.

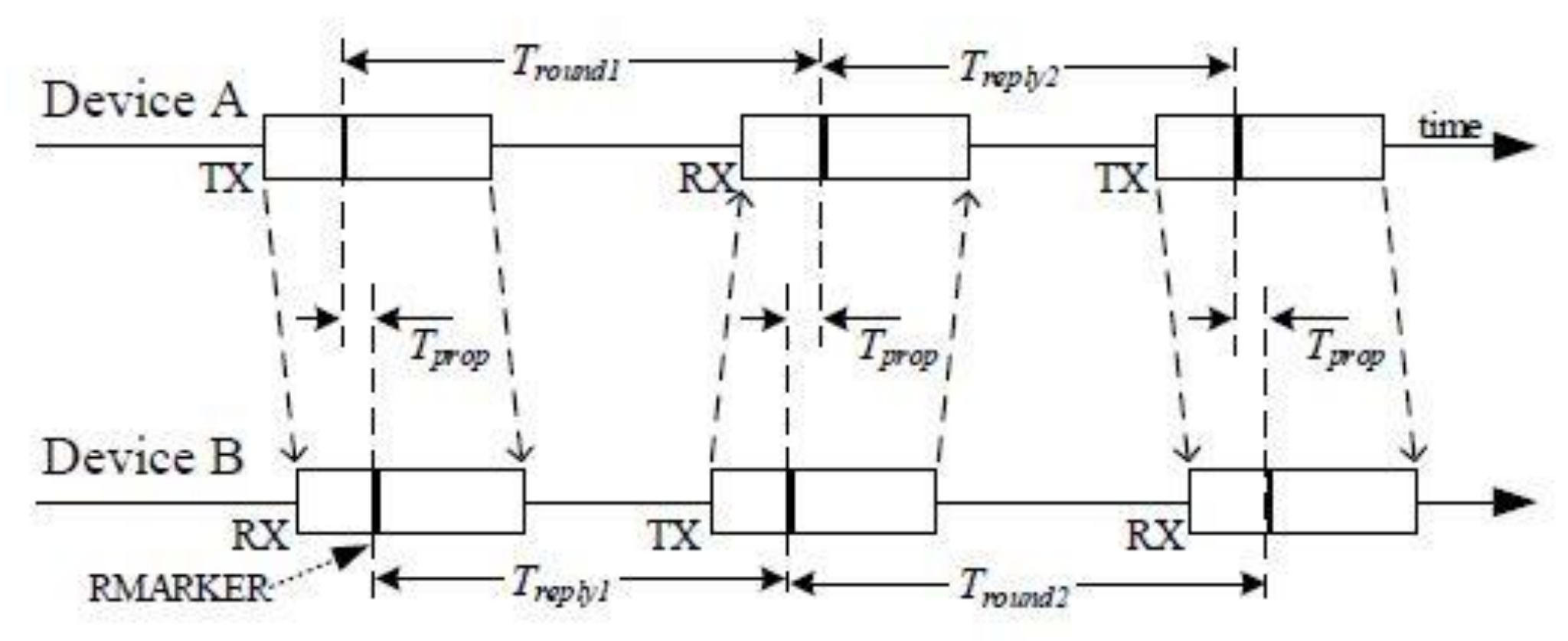




Software and Algorithms

Time-of-Flight Ranging Algorithm

We use a 3-message ranging exchange with timestamped messages to compute the distance between any two beacons. This double-sided two-way ranging algorithm eliminates the error due to clock skew between the two devices to under one percent. A diagram of the algorithm is presented below, with Device A initiating the exchange and Device B computing the distance.



Beacon Communication and Tracking (TDMA)

We used a Time-Division Multiple Access (TDMA) scheme to synchronize communication between our beacons. When computing the relative positions of the beacons, each beacon is assigned a slot during which it communicates with a specific target beacon. After the relative positions have been computed, the beacons enter into receive mode and begin listening for tags to localize. The tags also perform TDMA, and each is assigned a slot during which it initiates ranging exchanges with all the beacons in the system.

Bootstrapping

In order to compute the relative positions of the beacons, so as to avoid requiring a lengthy setup procedure involving precisely measuring the locations of the beacons, we developed a bootstrapping procedure. During bootstrapping, the localization system receives distances between pairs of beacons. From this, a pairwise distance matrix is constructed, with potentially missing entries. In order to ensure a sensible solution, constraints are placed on the sparsity of the matrix, namely that each beacon must have at least 5 valid measurements. Given a valid matrix, a gradient descent approach is taken in order to determine potential locations for points which would generate the observed pairwise distance matrix. Points are seeded randomly inside a cube with dimension equal to the average distance measured. From the current guess for points, a second pairwise distance matrix is created. The Frobenius norm of the difference of the two matrices at measured points (that is, measurement error) is optimized to be zero. Bootstrapping is considered complete upon reaching a certain error threshold. At this point, locations of beacons are considered relatively stable. Note that these positions have been determined in some arbitrary space; they are within an unknown homogenous transformation of any chosen world frame.

Tracking and Localization

Tracking tags poses a much simpler problem to solve. With the beacon positions considered stable, the tags simply require measurements to at least 3 different beacons to have their positions determined completely in the aforementioned arbitrary space. Often, we will observe more than 3 measurements, which overconstrains the problem and helps ensure a more accurate solution in the face of measurement error. In a manner similar to the above, a loss value is determined from a given position by summing the errors between the valid observed distances and the distances from the estimated point to the predetermined beacon locations. This loss value is then minimized using a least squares black box optimization routine provided by SciPy. This is necessary due to the overconstrained nature of the problem, which is impossible to handle simply with a system of equations.

Loss Function

Below is presented the loss function used in both the gradient descent bootstrapping algorithm and the trilateration algorithm optimization.

$$L(\bar{x}) = \sum_{i=1}^{N} \left|\left|\hat{r}_i - \bar{r}_i\right|\right|^2$$

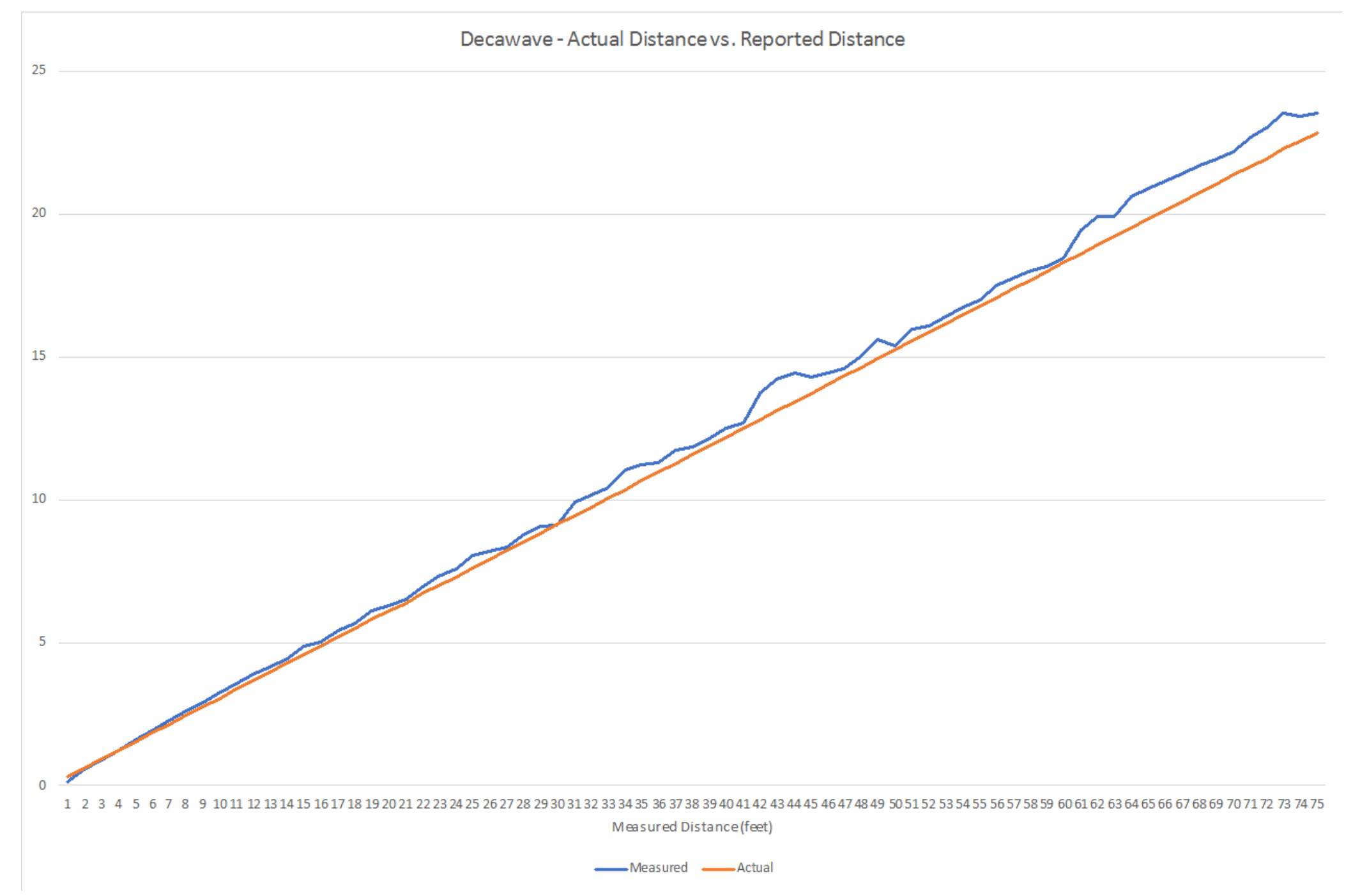
N is number of valid measurements

R_i is the calculated distance between estimated point x and point x_i

r is the measured distance to point x_i

Benchmarking

We performed a series of benchmarking tests on the Decawave chips, measuring the reported distances at various frequencies and data rates over a large number of distances. After performing these benchmarks, we determined that the data rate of 850kbps and channel of 499.2 MHz was most suited for our application. Below is a chart of measured Decawave distance readings, from 1 to 75 feet, plotted against the actual distance, at these settings.

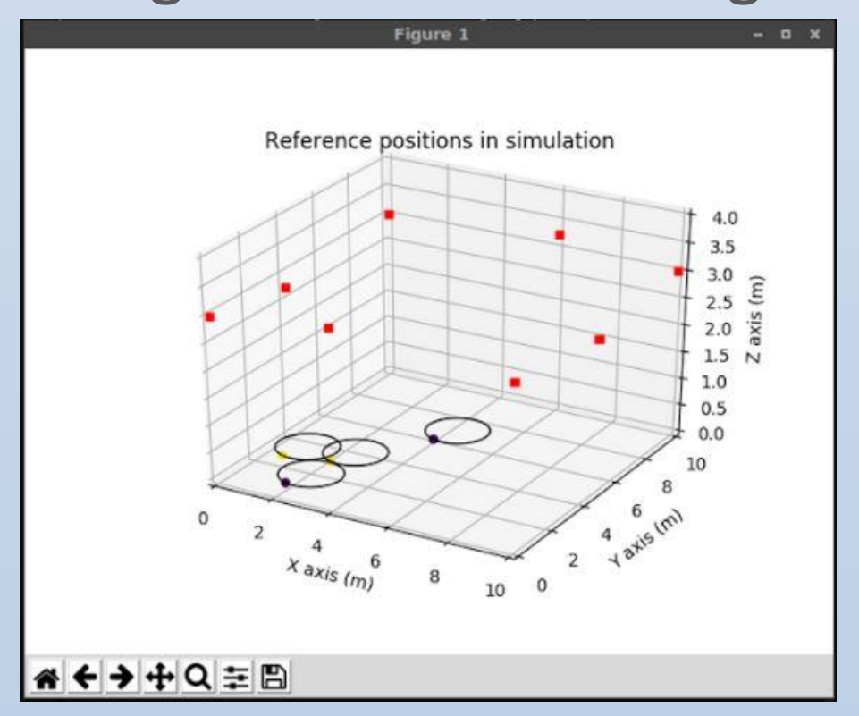


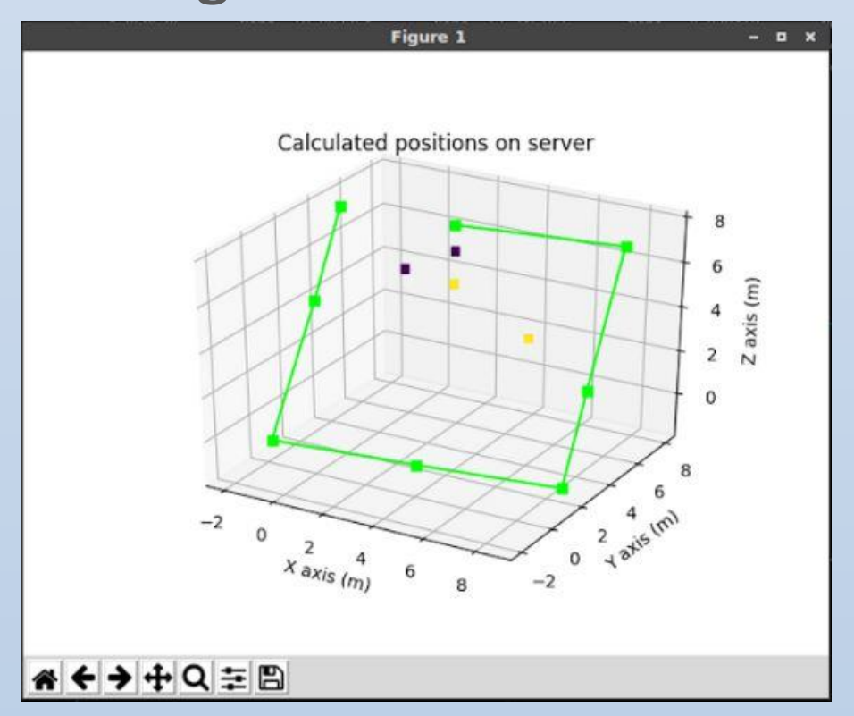
Overall, the Decawave chips performed admirably well when line of sight was maintained between the beacons. We noticed that there was some error as the distance increased; however, performing a simple linear regression on the benchmarked data reduced this error significantly.

Testing and Results

To test our localization algorithm, we constructed a beacon network of 7 beacons. Each beacon was of the assembly described in the Technology section – a Raspberry Pi Zero with a Decawave DWM1000 chip and a power-over-Ethernet attachment. All of the beacons connected to a PoE network switch to provide power and transfer distance measurements to a server running on the same network. We also constructed “wristbands” that would be tracked by this beacon network, which were simply Raspberry Pis with Decawave DWM1000 chips on them. We set this network up around a diving pool, where the maximum distance between the beacons was approximately 25 meters.

The gradient descent algorithm for computing the relative positions of the beacons proved highly effective. With randomly initialized positions, the algorithm typically needed only one set of 10 pairwise distance measurements between each beacon to converge to a solution. Below we show simulated beacon positions and the results after our gradient descent algorithm converged.





In addition, the trilateration algorithm for computing the positions of tags in the pool was effective, and had particularly accurate results due to the number of redundant distance measurements we had in our system. We were able to compute ten distance measurements for each tag in the pool, with each tag taking up a TDMA slot of 100ms. The frequency of position updates was therefore 100ms * N, where N is the number of tags in the pool.

To test the robustness of the system, we left it running while localizing multiple tags, and noticed that measurements continued to transmit for over half an hour after initially setting up the system. In addition, the system was robust to tags dropping in and out – as long as they did not change ID, as that would disrupt our slotted TDMA scheme.

Conclusion

Overall, we believe that the Decawave DWM1000 module is a highly promising piece of hardware for the challenge of indoor localization, and one that is robust to the challenges of an aquatic environment.

One interesting challenge we encountered in the process of testing this localization system was dealing with the positions provided by our algorithm being computed in an arbitrary coordinate frame. We believe this is actually helpful, as it allows us to impose no constraints on the setup of the localization system, and leave the processing of the positions to the next step in the pipeline. We did, however, discover that this problem is rather simple to solve. By taking 3 coplanar position measurements in the coordinate frame we wanted to work in, we were able to construct a coordinate basis and compute the transformation from the arbitrary localization frame to our physical Cartesian frame with a simple Singular Value Decomposition operation.

We hope that the gradient descent bootstrapping and triangulation algorithms presented in this work can be used in indoor localization systems in the future, particularly to avoid extensive and tedious setup and calibration procedures in such systems.

Future Work

One significant area of future work is improving the beacon communication algorithm. Rather than using TDMA, which has linearly increasing response time with respect to number of beacons, we could use a scheme that does not require a discrete number of slots. In addition, we could try to dynamically vary the delay between transmitting messages using an exponential backoff scheme, allowing us to more fairly distribute the network communication.

Also, we would like to make this system more robust to breaking line-of-sight. Currently, both the bootstrapping and trilateration algorithms are robust to measurement error; however, we can implement some form of normalization or outlier rejection to deal with situations where line of sight is broken between two beacons, resulting in a much larger reported distance.

Acknowledgements

We would like to thank our advisor, Anthony Rowe.
We would also like to thank the CMU Undergraduate Research Office for providing the SURG grant infrastructure, and the generous donors who made this research possible.