Localization for FRC

A Term Report

Jinan (Dorothy) Hu, Peter Mitrano, Kacper Puczydlowski, Nicolette Vere

October 12, 2017

1 Introduction

Knowing the position and orientation of a mobile robot is critical to many tasks. For robots designed for high-speed gameplay, knowing the position and orientation allows the robot to perform complex autonomous behaviors such as shooting and retrieving game objects. In this report, we describe a system for determining the pose (x, y, θ) of a mobile robot in a cluttered environment. The environment we are interested in is the FIRST Robotics Competition (FRC). FRC is a challenging environment because the robots make rapid and aggressive maneuvers by human drivers for part of the time, and at other times are under complete autonomy. Another challenge is that FRC fields are cluttered with other robots and game pieces such as soccer balls, pool noddles, or inflatable shapes, and these objects change from year to year. A successful localization system for FRC must support up to six robots, as well as occlusion from the playing field elements, unpredictable lighting, and frequent impacts. Our research suggests that there are at least five appropriate methods for localization: cameras and tags, radio and ultrasonic beacons, optical flow, dead reckoning with encoders, and dead reckoning with an IMU. All of these methods have seen success in robot localization.

This report begins with a review of existing literature on indoor localization for mobile robots in the Related Work section. The strengths and weaknesses of these existing methods are described in the Evaluation of Localization Techniques section. The Experimental Results section contains information on experiments we conducted. The System Specification section provides a complete description of our design criteria and system specifications. The Conclusion section details our plan moving forward.

2 Related Work

Robot localization has been studied for decades. Overall, the problem of localizing a mobile robot can be viewed as accurately measure the absolute distance to known landmarks, or by measuring the changes in position over time. All localization methods lie somewhere on a spectrum between these two points of view, and we will henceforth refer to these two ideas as global and local pose estimation. Some of the high level techniques for robot localization are: measuring range at various points around the robot and matching these readings to a map, measuring time of flight or difference of time of arrival to calculate distance to known locations, recognizing landmarks and computing pose relative to those landmarks, and measuring changes in pose and accumulating these changes over time. There

are different sensors that can be used for each of these techniques, such as laser range finders, ultrasonic, cameras, inertial measurement units (IMU), encoders, radio, infrared light, visible light, and human-audible sound. Although there are a tremendous number of possible methods for indoor mobile robot localization, there are a few which have received the most attention and shown the most success. These include:

- Lidar mapping
- Ultrasonic mapping
- IMU and Encoders fusion
- Infrared or Radio and Ultrasonic beacons
- Wireless network methods based on signal strength
- cameras with visually identifiable tags
- Optical flow mice and cameras

In our research, we learned how these techniques work and found descriptions and implementations in order to evaluate them. These descriptions and implementations are presented in this section with the purpose of demonstrating a thorough literature review and of providing background information to the reader.

2.1 Lidar Mapping

Lidar is a sensor that works by measuring the amount of time it takes light to return to the Lidar after hitting a desired object [9]. Light travels at about $0.3\,\mathrm{m\,ns^{-1}}$ which is a constant speed[9]. The sensor sends a certain amount of beams at certain intervals towards an object. Since light moves at a constant speed the Lidar can calculate the distance between itself and the object that light was hitting. It can do

$$\frac{d * c}{2}$$

where d is distance and c is speed of light and then divided by two to account for traveling to the object and back. By repeating this process the Lidar can produce a map of its surroundings by finding the distance between it and surrounding objects within its detecting range [9]. There is three types of information Lidar can collect depending on the type of Lidar. The first is the range to the target which is found topographical Lidar[9]. A differential Absorption Lidar can find the chemical properties of the targets it is measuring. A doppler Lidar can measure the velocity of a target[9]. This project would use a topographical LiDAR to measure the distance from itself to other objects to find position. Most Lidar have two main pulse systems for measuring distance. The first system micropulse have have lower powered lasers that are usually considered safer[9]. The wavelength for these is typically 1.0-1.5 m[21].The second system is high energy which have stronger lasers and are typically only used for atmospheric measurements [9]. The wavelength of these is typically 0.5-0.55 m [21].

2.2 Ultrasonic Mapping

Ultrasonic mapping (often referred to as sonar) was one of the first techniques used for indoor robot localization, and has been explored deeply since the 1980's. The most common approach is to use multiple emitter-receiver pairs placed around the perimeter of the robot, measure the range at each of those points, then localize to a given map of the environment [6]. Alternately, some systems use one sensor and rotate it around to achieve the same effect [12, 6]. The algorithms for interpreting the measured distances work by first extracting lines, then matching these lines to an existing map. Reported accuracies of the system in [6] was 1 ft for position, and 10° for angle. In [6] and [12], the reported rate of position updates is 1 Hz. Additionally, some methods will explicitly model the uncertainty of the position estimate, or explicitly model the behavior of ultrasonic sensors to ignore unreliable data. A more recent and sophisticated approach to localizing with sonar can be found in [20], in which 24 sonar sensors in conjunction with encoders were used to perform simultaneous localization and mapping. Their experimental results report drifts of 3.9 m and 21° over the course of 35 m of travel.

2.3 IMUs and Encoders

An inertial measurement unit (IMU) is a sensor reporting the forces acting upon an object, the angular rates of change, and surrounding magnetic field. The device typically comprises an accelerometer, gyroscope, and magnetometer which sense the above data respectively. These devices function by detecting Coriolis forces, inertial forces acting in the direction of motion relative to a rotating reference frame. These forces are proportional to the magnitude of the acceleration. These forces may be detected by a simple strain gauge mechanism or object vibrating at a know frequency (the rate of change of vibration frequency is detected) [**?**]. The premise behind position sensing using this device involves integrating the data with respect to time to calculate position and orientation. This approach was first used in aeronautics to estimate projectile attitude, orientation, and position [**?**]. High cost IMU's have been used historically for defense and transportation systems; the quality of the sensor is high and the data is reliable in these applications. An inertial navigation system (INS) often comprises multiple accelerometers, gyroscopes, and magnetometers to estimate orientation and position. Their performance is increased by referencing, or filtering, one sensor to estimate the error from another. Simple double integration of a filtered system using expensive sensors is often sufficient for position tracking applications like ballistics or missile tracking [**?**].

In cost-sensitive systems, this methodology is subject to high error rates from accumulation. Because of integration of accelerometer data, the velocity error term grows linearly and position error quadratically. This introduces a need for filtering, sensor fusion, and optimization based approaches.

Rarely in mobile applications are IMUs the primary position sensor. Odometers and encoders offer relative position sensing because they measure change between distances, but must be provided with a frame of reference. Most odometry-based sensors are updated at a high-frequency, but are subject to high error rates from gear inefficiencies, wheel slippage during normal rotation, and irregularities in data processing [**?**]. Complementary filtering, or using multiple, weighted sources of data, is used to obtain an position estimate. Linear quadratic estimation can be used to estimate the position of a object; however, this technique is relatively complex. The algorithm makes a prediction about the current position, taking into account error from the previous measurements. Once the current data is available, the

algorithm uses it to correct the parameters it used to make the estimation, provided the data is of high certainty. Known as a Kalman filter, this process uses a known model of the system and assumes errors in the sensors to smoothen the data. Systems can leverage data from a range sensor [**?**] or indoor positioning systems that use radio frequency signals [**?**].

Cameras, radio beacons, GPS, and similar landmark-based technologies offer global position sensing and have lower accumulated error than local systems discussed above. However, update frequencies are comparatively low, leading to false predictions or periods of unknown position. To compensate for this phenomenon, systems can leverage data streams with higher update frequencies to interpolate between frames. Forster *et al.* describe a visual-inertial navigation system that uses accelerometer and gyroscope data streamed at a high frequency to estimate a pose trajectory between select frames from the camera. Known as keyframes, these camera data are selected to minimize computational requirements while minimizing feature loss. Often, a parallel thread selects appropriate frames. Then, the IMU data are processed into a relative motion constraint. Effectively, this yields a trajectory of position and orientation between camera updates [**?**].

If the rate at which the position must be updated is lower than the update rate of the data, many values can be processed and used to calculate an approximation within a given time window. Known as preintegration, this technique, instead of filtering the data, combines many data points into a single trajectory estimate. Then, it transforms the data into the navigation frame, allowing for a smoother approximation of system position. This was beneficial in cases where global position data was unavailable for extended periods of time and decreased the computational load of the localization thread [**?**]. Systems like this are effective because camera data is used to correct IMU data on-line when data is available and rarely fails to update the pose estimate between frames. The above work describes an overall CPU time of about 10ms for data processing and real-time execution, although the system update frequency is unknown.

Leveraging preintegration, systems expanded sensor fusion frameworks with probabilistic terms and models of relationships between sensors. One such approach used a factor-graph to represent system state variables as nodes on a tree and the functional relationship between them as factors (edges). Instead of updating the state each time new data is available, the factor tree updates relevant state variables add into account an error term [**?**]. A key aspect of this system relies on frame transformations between sensor data happening at a low level. This takes workload off of the main CPU by utilizing FPGAs or other processors to monitor and process odometry or IMU data [**?**]. Implementations of such systems claim reduced computational load and similar performance to ORB-SLAM and other modern navigation systems.

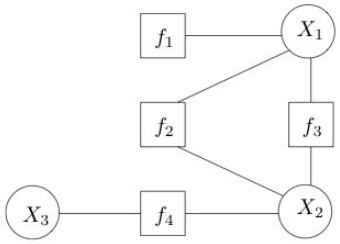


Figure 1: A simple factor graph [**?**].

Designed for the FIRST Robotics Challenge, frameworks such as Sensor Fusion 2 provide users with an algorithm for latency correction between IMU and camera data. Again, this algorithm uses known system parameters, such as update frequencies of sensors, frame transformations between sensors, and data from landmarks for filtering and position estimation. Usually, the IMU data is available at a rate an order of magnitude larger than that of the camera. Additionally, the data is accurately timestamped and easily accessible to the vision processing thread. This way, the user receives an updated pose estimate without lag and has an history of the orientation. This is useful for trajectory planning or on-line path correction (like recovering from a collision). Within the next several years, systems like this will mature, offering localization or possibly SLAM.

2.4 Beacon systems and Wireless Networks

Beacon systems have been used many times with success in the literature. Generally, these systems use ultrasound and or radio as a medium and either signal strength, phase shift, or time to measure distance to the beacons. Among radio systems, the system in [3] identified the location of people moving around buildings using signal strength in the 2.4gHz band received at three or more beacons, and they report accuracy of a few meters with an update rate of at most four times per second. The systems described in [5] uses passive RFID tags on the ceiling and an RFID transmitter on the robot, and report an accuracy of 4cm within a $5m^2$. Another RFID system [16] also uses signal strength to RFID, and reports accuracies for various configurations ranging from 1cm to 3m. These RFID systems use readers that cost over $500.

There are also countless localization systems that use standard wireless networks. A comprehensive survey of these systems can be found in [14]. Systems that use signal strength in standard wireless LAN networks have achieved up to 10cm accuracy and hundreds of updates per second. Another radio beacon solution is to substitute single-frequency radio with Ultra-wideband radio. These systems can achieve centimeter level accuracy, but they use obscure or custom made transmitters and receivers that cost in the hundreds of dollars [1] [2].

Among ultrasonic beacon systems, [11] uses the raw arrival times of ultrasonic pulses over time plus odometry together in a Kalman filter. Many beacon systems use the speed difference between sound and electromagnetic waves to measure system. Systems like [18], [22], and [10] send radio pulses followed by ultrasonic pulses. Nodes in the network us the difference in arrival time of these two signals to measure distance. Alternately, some systems use infrared pulses in place of radio [7] [24]. These systems are inexpensive, and report accuracy of between 2 cm and 14 cm.

2.5 Cameras with Visual Tags

Vision based localization is widely used in robotics. Different visual approaches have applied on mobile robot systems. Most of the vision based system have cameras on the robots and had either natural landmarks or artificial landmarks in the environments as references to absolute positions. If the natural landmark approach is used, knowing the absolute positions of objects in the circumstance, a robot could calculate its absolute position by calculating orientation and distance toward the object detected and recognized with its absolute position. Another similar approach is using 3D models and 2D to 3D matching techniques. The system described in [17] had accurately localized the camera's position

using this 2D to 3D mapping technique. Among all of the vision based approaches, the usage of a combination of cameras and artificial landmarks to localize robots is one of the most popular. Common artificial landmarks include 1D binary barcode, 2D binary barcode and 2D colored barcode. The system in [13] applied the approach of using cameras and ID tags. The system had ID tags on the ceiling, which were 2 m away from the floor. A web camera,facing to the ceiling, was mounted on a moving robot with a speed of $20\,\mathrm{cm\,s^{-1}}$. This system measured the position and orientation of the robot relative to the ID tags through image processing. The result of the experiment in [13] showed that this method was accurate even though there was an unevenness between the ceiling and the floor. Another system [8] also used camera and tags. However, instead of sticking ID tags on the ceiling, it put invisible tags on the floor by every 3.75 cm. The camera it used was surrounded by a UV light, which allowed the camera to capture those invisible tags. This system performed really well in homelike environments, it only had few centimeters of error. Another barcode based localization system of low capacity mobile robot (8 KB memory, 16 MIPs) [4] used 1d barcodes as references. Using a camera with 80 vertical pixels and 640 horizontal pixels, the system achieved localization within $3.5\,\mathrm{m\,s^{-1}}$ of error on average. An experiment on 2D barcode based localization was performed on a Pioneer 3 - AT robot. The robot was placed in a random location in the environment and had a Canon VC-C50i analog camera connected with a 1.8 GHz processor and worked under Linux operating System. The robot moved with a speed of $50\,\mathrm{mm\,s^{-1}}$, trying to locate itself by finding a code vertically mounted on a wall. The code contained information about its xyz position, normal vectors and its length, therefore, the system wouldn't need to query through databases to find information, which made the process faster.

2.6 Optical Flow

Optical flow is the ability to track changes between camera frames and measure the differences within them to track position. Optical flow is where given a sequence of images it can find the movement of objects between the images. More exactly optical flow looks at the movement of pixels between images. There are many assumptions about the image that has to be made in order to calculate in order to calculate optical flow. The first is that the lighting in the image stays consistent in how the lighting is throughout the sequence of images [15]. In an images inconsistent lighting, transparent objects would all violate this assumption so the amount in the image should be made as small as possible [15]. There are many methods of calculating optical flow that deal with different constraints with lead to varying drawbacks and benefits. This first is the Horn and Schunk method which calculates optical flow of the whole image making it considered a global method [15]. Along with the lighting constraint it also adds that the optical flow field should be as smooth as possible and have few variations in the image. The way optical flow is calculated is it finds the error for every pixels optical flow vector. The error is based on how much of the two constraints there are. The more illumination between pixels surrounding it and the less smooth it is across pixels the higher the error will be [15]. It only checks the illumination and smoothness in a few pixels surrounding each pixel, it does not compare every pixel to every other pixel in an image. It then adds up all of the error values it found which is called temporal gradient. It does this for the x components and y. components [15]. It also finds the error in the difference between the an average previous optical flow vector and the current ones spatial gradients and temporal gradients. The spatial gradient is the x component and y component difference between the pixel and each neighboring pixel. It then uses these found

$$u^{n+1} = \bar{u}^n - \frac{I_x[I_x\bar{u}^n + I_y\bar{v}^n + I_t]}{\alpha^2 + I_x^2 + I_y^2}$$
$$v^{n+1} = \bar{v}^n - \frac{I_x[I_y\bar{u}^n + I_y\bar{v}^n + I_t]}{\alpha^2 + I_x^2 + I_y^2} \quad (1)$$

Figure 2: Horn and Schunk Optical Flow vector equation

$$E_v = \sum_{p\in\Omega} W^2(p)[\nabla I(p) \cdot v + I_t(p)] \quad (2)$$

Figure 3: Lucas Kanade Optical Flow vector equation

values in the an equation to find the optical flow vector of the current pixel. These values are used in the equation below [15]. I_x and I_y are the spatial gradient of the current pixel. It is the temporal gradient of the current pixel. α is a weighting term. $\bar{u}$ and $\bar{v}$ are the neighboring pixels average optical flow vector values for $\bar{u}$ and $\bar{v}$. This is expressed formally in equation 2 [15]. n represents which iteration it is on. Each current pixels optical flow is calculated based on previous iterations pixels. This is one of the most popular global optical flow methods.

Optical flow can also be done locally using the Lucas Kanade method. This methods is based off of the assumption that the optical flow vector of pixels are similar to their surrounding pixels. This method finds an optical flow vectors that is consistent with its neighboring pixels temporal gradients and spatial gradients [15]. Each neighbor is then given a weight based off of how close it is to the pixel. The farther away a pixel is the lower a weight it is assigned [15]. This is because spatial and temporal gradients are based on how far away a pixel is so the error will be larger. Having a lower weight will reduce this. The formula for the optical flow vector is a least squares equation shown below in equation 3 [15].

$\nabla I(p)$ and It(p) are the spatial gradient and the temporal gradient for each of the neighboring pixels p. V is the optical flow vector for pixel located at x,y on the image [15]. W(p) is the weight assigned for each pixel. Local methods tend to work better since they do not allow for information about vectors to spread to unrelated regions of the image [15]. The constraints such as needing consistent smoothness and illumination can lessen the amount this impacts global optical flow methods but does not completely solve the problem. There are a variety of other optical flow methods that focus on different ways of comparing pixels within images but local and global are the most popular methods [15].

Lucas Kanade looks at pixels which have a certain pixels that have a large gradient in intensity of color[19]. It changed images to be gray scaled in order to see changes in darkness instead of colors [19]. Lucas Kanade works best when it uses pixels that have three certain qualities [19]. The first is that the pixels in a small have the same brightness [19]. Even though the location of the pixel has changed the brightness is consistent, so using a pixel from that area to track is optimal [19]. They also should be temporally similar meaning the path the pixel takes should change gradually [19]. Pixels should also be spatially similar meaning that neighboring pixels should have similar motion to each other [19]. Picking

pixels that fit these requirements work best with Lucas Kanade.

3 Evaluation of Localization Techniques

Each of the techniques presented thus far have strengths and weaknesses. In cases where those strengths and weaknesses are orthogonal, combining multiple techniques improves the overall performance. This is the fundamental principle behind sensor fusion. For example, in [10] the authors use a compass to make up for the inability of beacons to measure orientation of the robot. In order to tackle all of the diverse challenges of localization in the FRC environment, we believe it is necessary to combine techniques. In this section we will outline the advantages and disadvantages of each technique, justify why none of the techniques discussed are sufficient on their own, and explain which techniques we will pursue and why.

Inertial sensing offers several promising results, but suffers from accumulated drift is not developed enough to function as a stand-alone localization system. The cost of the sensor is inversely proportional to error, and high quality devices are prohibitively expensive. A majority of end-users of the system proposed in this paper have experience with inertial sensing, presenting a need for support of basic features such as heading detection and filtering. This complements other subsystems because they suffer from lower update frequencies and physical sources of error. In this way, an IMU is used to complement an existing suite of sensors. Although research in this field is extensive, much is developed around outdoor navigation or aeronautics and requires some adaptation. Extensive incorporation of advanced techniques may be out of scope for this project.

Occlusion is one of the major challenges facing global localization systems designed for FRC. In many cases, there are other robots or field obstacles than can block vision or ultrasonic signals. However, there are no obstacles that would block radio or other high frequency signals. Lidar and ultrasonic systems both struggle in this case because the map of the field is interrupted. Occlusion is also problematic for cameras looking for tags and ultrasonic beacons. However, local pose estimation techniques like encoders and IMUs are unaffected by occlusion. Pairing IMUs and encoders with cameras or beacons has the advantage of not failing completely during short periods of occlusion. Furthermore, by using both beacons and cameras, we can increase likelihood that one of the two systems will be able to reach their landmarks. Wireless networks perform very under these scenarios, but they require knowledge and control over the 2.5 gHz spectrum in the area where they are used. At FRC events, there can be dozens of wireless networks running, as well as the wireless networks used on the field for communication between robots. For this reason, we feel that techniques using wireless frequency have too many unknown variables.

Another major issue with any methods relying on the reflection of ultrasonic is the interference between robots. If multiple robots range ultrasonic near one another, there could be cross talk or interference between the signals. For is reason enough to rule out any use of reflecting ultrasonic. Note however that ultrasonic beacons do not have this weakness, since the pulses being emitted are not expected to reflect.

Lidar mapping has been shown to be one of the highest performing localization methods in terms of accuracy, precision and update rate. The two reasons why we are not pursuing it further are cost and applicability to FRC. Lidars capable of ranging across an entire FRC field are over $400, which is the cost limit for any single part on an FRC robot. Lidar techniques also require either mapping on the fly, or an existing map. Mapping on the fly presents its own challenges, and usually suffers from very bad localization for some initial

period of time while the map is built. Existing maps would work very well on the competition FRC fields, but would not apply in the practice spaces teams us, and it is unrealistic to have a consistent practice space in an FRC shop.

Radio and ultrasonic beacons are very attractive because of their low cost, flexibility of setup. The cost of each beacon, according to the specifications laid out in section 5.2, are projected to cost about $30 (see 5.4.1). Beacons have more flexibility in their placement than tags because they are much smaller and do not need to be on flat surfaces, or in specific orientations.

Optical flow gives us accurate angle measurements and fast updates that are relative to our current position. Like all camera based solutions, the vibration of the robot will likely makes this technique difficult. However, cameras are the most widely used sensor according to our survey of FRC students and alumni. The camera also doubles as a sensor for matrix tags, which give us accurate position estimates in the global frame. This is complemented by beacons, which update more slowly, but are not effected by occlusion and is robust to vibration.

Among the vision based localization systems discussed in the Literature review, artificial landmark based systems were more preferable due to their low cost and ease of implementation. The combination of cameras and natural landmark could accurately and precisely localize. However, artificial landmark based systems could perform localization more accurately and precisely. Considering that the field of FRC changes over time and dynamic robots are moving all round the field during the competition as well as the complexity of implementing real time object recognizing algorithms, implementing an equally performed artificial landmarks based system seems to be a better choice. Not a lot of robot localization system use 1D barcodes as references since a 1D barcode can only contains up to 25 characters, which limits the length of information. Among 2D barcodes, fiducial tags and QR tags are two of most popular choices in mobile robot localization. The advantages and disadvantages of different types (QR, Data matrix, PDF417, fiducial tag) of 2D barcodes are discussed. QR code is designed to align with camera. It contains 268 pixels without payloads. Datamatrix is very similar to QR code, it has high fault tolerance and fast readability. Datamatrix can be recognized with up to 60% of damages. PDF417 is famous for a storage of huge amount of data. Complex information such as photographs, signatures can be inserted into PDF417 easily. Fiducial tags contain less information than QR codes. However, many of them can easily be detected in one shot and the process speed for fiducial tags is faster than of QR codes. One of the most commonly used 2D barcode in robotics is fiducial tag. A system in [8] measured the distance between AprilTags and the camera. A sheet of 16.7 cm AprilTags were tested from 0.5 m to 7 m away. The calculated distance was almost the same as real distance from 0.5 m to 6.5 m. The position errors were within 25 mfrom range 0 to 10 m. However, orientation errors were pretty high (1.5 degree off) when the off - axis angle was small, but were within 1 degree from 20° to 75° of off - axis angle. The detected rates for tags were 100% from 0 to 17 m away. This system showed that the combination of camera and fiducial tags can potentially localize robots accurately and precisely. The research, [7] developed an algorithm to enhance the quality of QR codes captured in order to improve the recognition rate. Its algorithm successfully recognized 96% of QR codes under a variety of qualities captured by a mobile phone camera. The average time for decoding a QR code is 593 ms. Another deblurring method in [23] can be applied to enhance the quality of motion-blurred ArUco code.

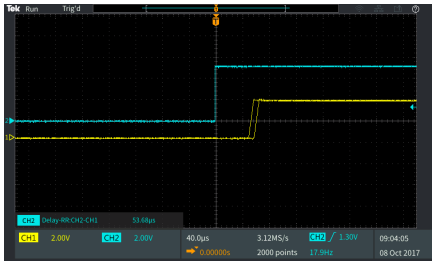


Figure 4: Example measurement total trip time for radio signal. The blue line is the input to the transmitter, and the yellow is the output of the receiver

4 Experimental Results

4.1 Measuring Beacon Delays

The beacon system relies on measuring the time it takes for a sound signal to travel from the beacons to the robot. To do this accurately, one must account for transmit and receive delays in addition to the actual time of flight. Figure 5.4.1 illustrates the various delays we need to account for. We conducted experiments to get initial estimates of these delays. First, to get an estimate of the radio transmit receive delay, a transmitter and receiver were set up on two microcontrollers. The transmitter sent 5 ms pulses of radio energy (no encoded data) every 55 ms, and oscilloscope probes were attached to the input pin on the transmitter and the output pin on the receiver. By comparing the time difference between the input and output signals on the oscilloscope, we can determine the total time. Furthermore, we can measure the distance between the transmitter and receiver and subtract from the total time the theoretical time of flight of the radio signal. The full data for these measurements are available in Appendix B, and an example measurement is shown in figure 4. The time of flight of radio over distances of a new centimeters or meters is on the order of nanoseconds. We measured an average delay of 45.175 µs, which we attribute to the internal circuitry of the transmitter and receiver. The variance of this delay was 16 µs. However, we also measured delays as low as 32 µs and as high as 79 µs. Since the theoretical time of flight over the distances used in this experiment were at most 1 ns, we can conclude that there is both delay and significant variance in the delay of the transmitters and receivers. This information will be used to better model the timing of our beacons to make the system as accurate as possible.

Next we performed a similar experiment with the ultrasonic transducers. For this experiment, we used two NTX-1004PZ piezo tweeters placed 25 cm apart. The NTX-1004PZ is meant to be a normal high-frequency for DJ rigs, and is designed to operate between 4 kHz and 20 kHz. However, because they are incredibly cheap we decided to evaluate them as ultrasonic speakers running just above that range. One was connected to a PSoC 5LP for transmitting, and the other was connected only to the oscilloscope. The other oscilloscope prove was connected to the transmitting piezo. The time difference between the transmitting signal on and the receiving signal was measured. The signal applied to the transmitter was short bursts of a 24kHz square wave. The measure time delay between the transmit and receive very closely matched the expected time for sound to travel between the speakers. This indicates that we will not need to account for constant delays in the transmit or receive circuits. However, there were several other noteworthy behaviors we discovered during these

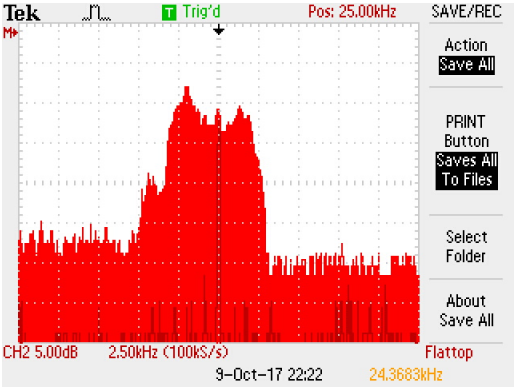


Figure 5: Frequency response of the NTX-1004PZ. The best response is achieved at 22kHz, and the highest detectable frequency is 27kHz.

tests. First, we noticed that any immediate changes in the frequency of the transmit signal will clicks in the audible range. these clicks do not effect the distance measurement, but they are mildly annoying and should be suppressed if possible by changing the transmit signal.

4.2 Measuring Frequency Response

After testing for delays in the ultrasonic circuitry, we also measured the frequency response of the NTX-1004PZ. We placed two tweeters in the same configuration as described above. Using a function generator, we transmitted a square wave at 8vPP and swept from 20 kHz to 30 kHz and back down over the course of 20 seconds. We attached an oscilloscope to the receiving speaker and captured the power at each frequency using the FFT mode, persisting the display over the course of the sweep to see how the frequency response changes across our frequency range. Figure 5 shows the results of this experiment. From this experiment, we learned that the best frequency response is achieved at 22 kHz, and the after 27 kHz the signal is indistinguishable from the noise.

5 System Specification

5.1 Design Criteria

Here we present our goals and the criteria our system must meet in order to be successful. Broadly, we consider the following factors to be those which are important to the success of our system.

1. **Accuracy**
   how close our position estimates are to ground truth.
2. **Precision**
   how close our position estimates are to each other given the same ground truth.
3. **Update Rate**
   how quickly does our system provide position estimates.

4. **Accessibility**
   how affordable is our system, how difficult is it to make, and how easy is it for teams to use.

To come up with hard numbers for these criteria, we first performed a few simple calculations based on our knowledge of FRC. First, we consider what teams would want to use position information for, and decided that the applications requiring the most accuracy are shooting and autonomous pick of game pieces at known locations. Both of these require the position estimates to be close to the true position of the robot. From there, we know that most FRC shooting and pickup mechanisms will work within $\pm 10\,\text{cm}$. Next, we decided the application requiring the most precision would be path following. If position estimates are imprecise and jump around rapidly, smooth path following will be difficult. From our experience with path following, we estimated that $\pm 5\,\text{cm}$ would be sufficient. For update rate, we considered what the maximum distance a robot could move within a period and used that to decide what our update rate should be. The very fastest FRC robots move $6\,\text{m}\,\text{s}^{-1}$, which at an update rate of every 20 ms is a distance of $0.02 * 6 = 0.12\,\text{m}$. The rate of 20 ms is a realistic cycle time in FRC, and we feel 12 cm is sufficient given the speed. For accessibility, we acknowledged that teams cannot spend more than $400 on any part, and that most times source parts from websites AndyMark, Cross-the-road Electronics, and National Instruments among other suppliers. We are also conscious that many FRC teams have limited or cluttered spaces for testing their robots, and may be working in a shared space that must be clean and usable after their work sessions.

Using all of these informal estimates as a starting point, we conducted a survey of FRC students, Alumni, and mentors. We received 65 responses in total, and used the results of this survey to solidify our design criteria. The full response of this survey are presented in Appendix A. In summary, the median for accuracy was 10 cm in x,y and 5° in yaw. Our survey did not include questions about precision and update rate, because they depend on what position is used for. Instead, we asked if students would try path planning if they had a localization system, which would back up our estimate of precision. Our survey indicated that 90% of students would try to make the robot autonomously follow paths. Therefore, our precision estimated based on path planning as an application is supported by our survey. Update rate was not addressed in the survey because we didn't think FRC students would have informed opinions on this metric. Finally, we asked several questions about the accessibility requirements. A cost of under $200 was deemed acceptable by 84.6% of responses, and so we have made $200 our goal for cost. Furthermore, we learned that the amount of space in teams shops varies from a 5 by 5 foot space up to several thousand square feet, but the median shop size is 775 ft^2, which one can imagine as a 25 by 30 ft space. In terms of access, about 76.5% of teams could leave up tags or beacons, with the others stating that they must clean up everything because they work in a shared space such as a classroom. Lastly, we asked students what sensors they were familiar with. The most familiar sensors were cameras (90%), followed by encoders (84.6%), then IMUs (60%). Therefore, it would be beneficial to incorporate cameras, encoders, and IMUs because teams are already familiar with them.

Ultimately, we formulated design criteria based on our own experience with FRC and with localization, as well as by conducting a survey of the needs, experience, and opinions of FRC participants. These design criteria will help us pick which localization techniques to pursue as well as define the goals for our system.

5.2 System Specification

Based on the extensive literature review, the few preliminary experiments we've conducted, and our design criteria, we eliminated our initial list of possible techniques down to the following five:

1. Radio and ultrasonic beacons
2. IMU
3. Drive wheel encoders
4. Optical flow
5. Camera with matrix tags

We have found examples of each of these techniques being used successfully, and in many cases have verified that they satisfy our criteria for accuracy, precision, update rate, and accessibility criteria. We are confident that any combination of these methods would work. These techniques are complementary in their sources of error, and together we believe they will make a robust localization system.

5.3 Timeline and Goals

To maximize the likelihood that our project is complete in timely manner and with rigorous methodology, we have drafted a set of goals and a timeline to help us manage our time. Below we present our set of *must* goals, which we have agreed are all required for our project to be a success (in no particular order).

- Support 4 wheel differential drive robots moving at continuous driving speeds of up to $3\,\mathrm{m\,s^{-1}}$
- Support the NavX and do complementary and basic Kalman filtering
- Differentiate between tags with the camera and measure pose relative to them with a standard 720p webcam.
- Beacons provide global position updates.
- Use the out of the box OpenCV API to get x and y translation between frames
- Provide a GUI for specifying the configuration of the sensors on the robot and the configuration of the field or practice space.
- Accurately describe the conditions under which camera localization will work, especially with respect to jitter and occlusion.
- Support getting the position from each subsystem individually
- Support getting one fused position update
- Specify what kinds of sensor mounting configurations are supported
- Specify which of the subsystems are necessary and which are optional (ex: encoders are necessary, beacons are optional)

- Provide the designs and the code for the beacons.
- Provide the optimal analysis or coverage of an empty FRC field and the 2018 field
- Provide the methodology for finding the optimal placement of tags and beacons for other shape fields
- Fail gracefully when some of the sensing techniques are unreliable or failing

These goals will hopefully constrain the scope of each localization technique such that all of them can be incorporated to some degree. Furthermore, we have drafted a timeline of which of these goals we will focus on over the course of the year. For example, by choosing to start with encoders, IMUs, and cameras, we will be able to have a working localization system even without the other techniques. Additionally, we hope that by giving an end date for some of the components we will be able to move on and not spend too long optimizing any one component.

- B Term:
  - Meet goals for encoders, IMUs, and cameras with tags
  - Specifying robot configuration and tag locations in the GUI
- C Term:
  - Meet goals for beacons and optical flow
  - Fuse all techniques into one position update
  - GUI supports configuring beacons
- D Term:
  - Meet documentation goals for teams
  - Provide the designs and the code for all components
  - Write the final report

5.4 Implementation Details

Having decided on the sensing techniques we will use, we now describe our detailed system Specification. Ideally, the description in this section serves as the full plan for our implementation of the system during B Term, and most of the major design decisions have been made and clearly presented.

5.4.1 Radio and Ultrasonic Beacons

Important implementation details of the beacon system include the number of beacons needed, the radio communications protocol, and the ultrasonic signal format. The number of beacons can either be picked to minimize cost, and then the beacons can be designed to meet this criteria, or the characteristics of the beacons can be determined and the number of beacons needed derived from that. In order to support arbitrary practice areas, the second method is preferred. In this section we cover relevant calculations and possible implementation details for beacons.

Overview of components

Each beacon consists of a PSoc5 LP processor, NTX-1004PZ piezo transducer, XY-MK 433 MHz radio transmitter, XY-FST 433 MHz radio receiver, and other supporting components. During normal operation, each beacon will listen for a radio signal telling it to emit an ultrasonic signal. We expect that a few other components such as resistors, LEDs, and a battery will be required for the final beacon.

Format for Ultrasonic Signal

The ultrasonic signal should be designed to maximize both the distance it can be detected from and accuracy with which it's timing can be detected. Using a simple pulse of fixed frequency and amplitude is not ideal because it trades off energy transmitted (and thus distance) with accuracy. This is because a longer pulse contained more energy and can be detected from further, but at the same time the receiver cannot distinguish timing within the length of the pulse. Therefore, a chirp signal will be used. Compare the usual function relating amplitude to time of a sine wave to the quadratic chirp signal.

$$x(t) = A\cos(\omega t + \phi)$$

$$x(t) = A\cos\left(2\pi\left(\tfrac{f_1 - f_0}{2T}t^2 + f_0 t\right) + \phi\right)$$

This function will generate a chirp with amplitude A, starting from frequency f_0 going to f_1 over time interval T. Figure 5.4.1 shows the specific waveform with the parameters we expect to use. This is a linear chirp signal because the instantaneous frequency of the signal changes linearly with time.

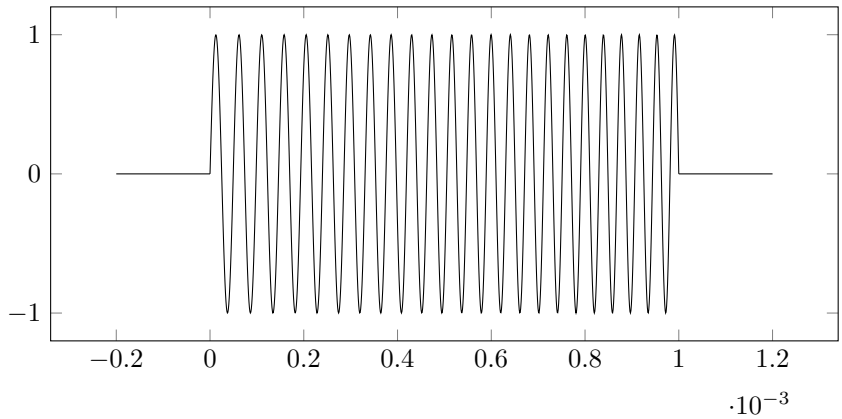


Figure 6: Chirp signal with $\phi = 0$, $A = 1$, $f_0 = 20\,\text{kHz}$, $f_1 = 27\,\text{kHz}$, and $T = 1\,\text{ms}$

This is the waveform we will be emitting from the ultrasonic sensors. The benefit is that it can be high power and still allow the receiver to determine precisely where in the waveform it is listening. One method for doing this is to take the discrete short time Fourier transform, which will show us how the power at various frequencies changes over time. By applying the FFT at a particular instant during the chirp, we can determine the position in time within the chirp of the current FFT. For instance, in the signal above if we apply and FFT at time $t_n = 1.0$ and discover high power at frequency $f = 22\,\text{kHz}$, we know that

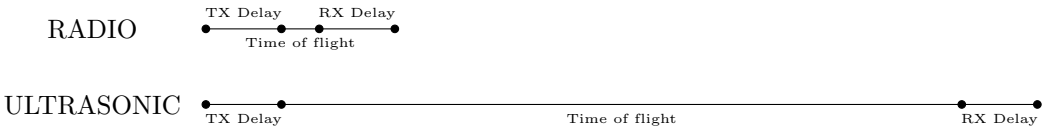


Figure 7: Timing of radio and ultrasonic signals. Experiments indicate 46.175 µs total RF delay and 1 ms total ultrasonic delay.

Item	Cost per Beacon
PSoc 5LP	$10.00
RF Tx/Rx Pair	$1.68
piezo speaker	$1.65
9v battery	$1.19
battery connector	$0.54
LCD display	$3.90 (optional)
resistors and capacitors	$5.00 (estimate)
prototyping board	$5.00 (estimate)
Total	28.96

Table 1: estimated bill of material for beacons

the chirp began at time $t_0 = t_n - \frac{f-f_0}{f_1-f_0}T = 1.0 - \frac{22-20}{27-20}0.001 = 0.99857$. Another simpler strategy is to slide the waveform you expect to hear across samples of recorded raw input and check for the point of highest correlation. In other words, slide the waveform in figure 5.4.1 and match it to the received signal. Both of these techniques are used in practice, and we are considering both in our implementation.

Our beacon system relies on accurately knowing the distance to a beacon, and therefore knowing exactly when the start of the signal left the transmitter and when the start of the signal arrived at the receiver. Figure 5.4.1 shows the sources of delay we are accounting for, and we describe our methodology for measuring these delays in the Experimental Results section.

Path Loss

We calculate the free space path loss (FSPL) of the radio signals at the farthest distance from the beacons. For this calculation, we assume the worst case where the beacon is on the other side of the length field, which is 16.5 m away. Over the more realistic distance of half the width of the field (5.6 m), the path loss is only 40 dB.

$$\text{FSPL} = 20\log_{10}\left(\frac{4\pi R f^2}{c^2}\right) = 20\log_{10}\left(\frac{4\pi * 16.5 * 433 \times 10^6\,\text{Hz}}{3 \times 10^8\,\text{m}\,\text{s}^{-1}}\right) = 49.521$$

Self-Localization of Beacons

Beacons that can automatically discover each other and their reliative positions will tremendously improve the easy of setup for the beacon system. We believe this is possible, and in this section we describe a protocol for doing so.

When the first beacon is turned on, it will listen for radio packets for a brief period of time to check if it is the first beacon to be turned on. If it hears nothing, it will assume it

is the first beacon and assign itself ID 0. This beacon will be responsible for sending the synchronizing radio pulse as well as handle the distribution of IDs to the other beacons. At this point, beacon 0 will send packets advertising that ID 1 is available. When the next beacon is turned on, it will get the message that ID 1 is avilable, assign itself that ID, and respond with an ACK message confirming that ID 1 is now taken. Beacon 0 will then begin advertising that ID 2 is available. This process continues for N beacons. Once each beacon has its asigned ID number, beacon 0 will start organizing the self-localization of each beacon. First, beacon 0 will send a packet telling everyone that beacon 0 will transmit ultrasonic. All beacons other than beacon 0 will hear this and start a timer, and beacon 0 will itself transmit ultrasonic. Each beacon will receive this ultrasonic and compute its distance to beacon 0. After wiating a fixed period of time (such as 50 ms) for this process to complete, beacon 0 will send a packet telling everyone that beacon 1 willtransmit ultrasonic. This process continues until every beacon has recorded its distance to every other beacon. At this point, beacon 0 will tell each beacon sequentially to report its distance calculations to beacon 0. Beacon 0 will then compute the location of every beacons by minimizing the least squares error in the system of equations defining the beacon locations. Finally, it will report the resulting positions to all the beacons. These communications can also be monitors from a laptop with an inexpensive USB software defined radio, such that the status and result of this process can be shown in a GUI and debugging information can be communicated.

Number of beacons needed

The number of beacons can be decided once the range and beam pattern of the beacons is known. To do this, a rigorous test that measures the sound level at various angles and distance will be performed. Once this is known, we can offer a tool or set of steps to determine how many beacons are needed and where beacons should be placed.

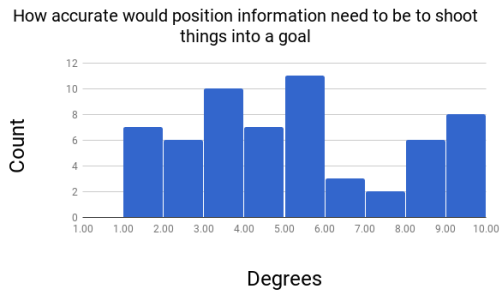
6 Conclusion

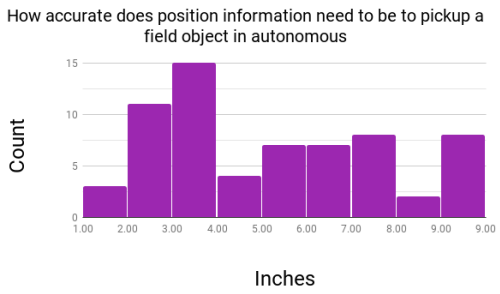
Over the course of A Term, we conducted thorough background research on various methods for localization and evaluated their performance. Based on numerous report for each localization technique, a survey of FRC students and alumni, and our own knowledge of the challenges of localization for FRC we narrowed down the list of possible techniques to encoders, IMUs, beacons, cameras and tags, and optical flow. Each of these methods has been shown to work in other indoor mobile robot environments, and together they make up for the weakness of each individual method. Our strategy will be to apply each of these five techniques to the degree we believe is necessary to meet our design criteria, rather than to choose one method and optimize it deeply. Our performance will be judged against the design criteria stated above, and we have developed goals and a timeline to guide our work moving forward.

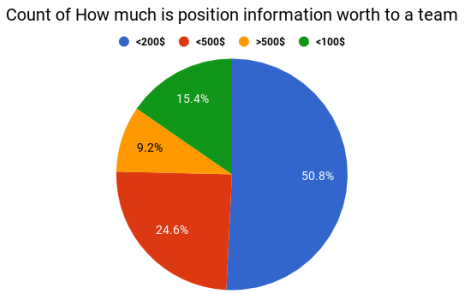
7 Appendix

7.1 Appendix A

Survey Responses







7.2 Appendix B

Measured Distance (m)	Measured Total Time (μs)			Average Delay
0.0630	45.44	42.80	34.48	40.90646
0.1425	52.72	50.48	52.09	51.76286
0.1505	64.16	63.36	60.24	62.58616
0.2210	40.33	36.79	36.40	37.83926
0.2415	49.52	45.76	43.92	46.39919
0.2460	47.47	53.84	44.71	48.67251
0.2965	34.36	34.00	43.76	37.37234
0.3085	79.36	62.16	59.52	67.01230
0.3390	39.92	57.27	38.96	45.38220
0.3770	41.75	40.75	45.53	42.67541
0.3600	38.40	38.40	37.68	38.15880
0.0070	35.60	36.08	34.32	35.33331
Average Delay (μs)				46.175

Table 2: The time of flight of radio over tens of centimeters is insignificant compared to the delay within the transmitter and receiver.

References

[1] Dart Ultra Wideband UWB Technology | Zebra.

[2] Pozyx - centimeter positioning for Arduino.

[3] P. Bahl and V. N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, volume 2, pages 775–784 vol.2, 2000.

[4] Duarte Dias and Rodrigo Ventura. Barcode-based Localization of Low Capability Mobile Robots in Structured Environments. 2012.

[5] E. DiGiampaolo and F. Martinelli. Mobile Robot Localization Using the Phase of Passive UHF RFID Signals. *IEEE Transactions on Industrial Electronics*, 61(1):365–376, January 2014.

[6] M. Drumheller. Mobile Robot Localization Using Sonar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(2):325–332, March 1987.

[7] S. S. Ghidary, T. Tani, T. Takamori, and M. Hattori. A new home robot positioning system (HRPS) using IR switched multi ultrasonic sensors. In *1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings*, volume 4, pages 737–741 vol.4, 1999.

[8] Jinwook Huh, Woong Sik Chung, Sang Yep Nam, and Wan Kyun Chung. Mobile Robot Exploration in Indoor Environment Using Topological Structure with Invisible Barcodes. *ETRI Journal*, 29(2):189–200, April 2007.

[9] Marcoe Keith. LIDAR an Introduction and Overview, 2007.

[10] Hong-Shik Kim and Jong-Suk Choi. Advanced indoor localization using ultrasonic sensor and digital compass. In *2008 International Conference on Control, Automation and Systems*, pages 223–226, October 2008.

[11] L. Kleeman. Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2582–2587 vol.3, May 1992.

[12] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.

[13] Weiguo Lin, Songmin Jia, T. Abe, and K. Takase. Localization of mobile robot based on ID tag and WEB camera. In *IEEE Conference on Robotics, Automation and Mechatronics, 2004.*, volume 2, pages 851–856 vol.2, December 2004.

[14] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, November 2007.

[15] Peter O'Donovan. Optical Flow: Techniques and Application, April 2005.

[16] S. S. Saab and Z. S. Nakad. A Standalone RFID Indoor Positioning System Using Passive Tags. *IEEE Transactions on Industrial Electronics*, 58(5):1961–1970, May 2011.

[17] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674, November 2011.

[18] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka Priyantha. Tracking Moving Devices with the Cricket Location System. In *Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services*, MobiSys '04, pages 190–202, New York, NY, USA, 2004. ACM.

[19] Min Sun. Optical Flow.

[20] Juan Tardos, José Neira, Paul M. Newman, and John J. Leonard. Robust Mapping and Localization in Indoor Environments Using Sonar Data. *The International Journal of Robotics Research*, 21, April 2002.

[21] Lidar UK. How does LiDAR Work?, 2017.

[22] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997.

[23] W. Xu and S. McCloskey. 2d Barcode localization and motion deblurring using a flutter shutter camera. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 159–165, January 2011.

[24] H. Yucel, R. Edizkan, T. Ozkir, and A. Yazici. Development of indoor positioning system with ultrasonic and infrared signals. In *2012 International Symposium on Innovations in Intelligent Systems and Applications*, pages 1–4, July 2012.