Cricket as a Positioning System for Control Applications

Aleksandr Kushleyev
Travis Young

MERIT Program
Summer 2005
University of Maryland

Advisors:
Dr. Krishnaprasad
Sandy Klemm
Zachary Kulis

Table of Contents

1. Abstract

The Cricket System is a research technology that allows indoor location sensing using a combination of RF and ultrasonic ranging. It is particularly useful in indoor laboratory environments, where Global Positioning System (GPS) signals are not available. The goal of this project was to set up an absolute positioning system using Cricket and verify whether its performance is appropriate for use as feedback in testing various control algorithms. The types of experiments that were conducted ranged from simple distance measurements to integrating Cricket in various autonomous routines of a mobile robot. These experiments showed that a single distance measurement is accurate on the order of one centimeter, however a slightly larger error was observed in the position estimates. In addition, by varying the configuration of active beacons using feedback from the robot, a more efficient use of time was achieved, resulting in better performance when the robot was in motion.

2. Introduction

In the field of mobile robotics and control theory, positioning systems are an important component. In order for an autonomous robot to make decisions based on feedback from its environment, it needs to be able to locate itself in that environment. This can be done with a variety of techniques, such as infrared, sonar, and ultrasonic detectors. A common technique for determining the position of a robot is to use some form of relative positioning, whereby the robot counts the number of times it rotates its wheels, multiplies that by the diameter of the wheel, and comes up with the distance it has traveled from its starting point. However, an initial position fix is needed to use this method, and the measurements are dependent on previous ones, meaning that error will accumulate over time. An absolute positioning system is needed that provides independent measurements and will therefore not have the problems caused by relative methods.

The purpose of this study is to determine whether the Cricket System, which uses ultrasonic ranging to locate an object, is sufficiently reliable and accurate as an absolute positioning system for use in feedback-dependent control laws for mobile robotics. The accuracy of the Cricket System will be tested, and an absolute coordinate system will be established in the lab. This coordinate system will be used to test various position and orientation algorithms and their accuracy. Once the Cricket System is sufficiently stable, it will be integrated with a set of control laws that allows a robot to maintain a certain distance from an obstacle. This will verify the ability of the Cricket System to function in the place of relative positioning methods.

The ability of the Cricket System to function in an indoor environment is important because there is no established indoor positioning system. The Global Positioning System, a popular absolute system, uses radio signals which cannot penetrate indoors. With a growing interest in robotics in the home, a system will need to be established that allows a robot to navigate reliably where GPS signals do not reach. The Cricket System may be able to fill this need.

3. Experimental Set-up

3.1 Robotic Platform

The robot used in this project is the Pioneer2-AT8 manufactured by ActiveMedia robotics. It is equipped with a Pentium processor and running a Linux operating system. Sensors such as Cricket are handled using the ME (Modular Engine) program on the robot. ME delegates time slices to each module inside the system each "turn." Practically, this means that the Cricket Module is accessed approximately every 5 milliseconds. During this time, the serial port is checked for data. When data has been posted, the module reads in this data and proceeds to extract the required information, most importantly the Beacon IDs and their corresponding distances. The robot's motion is based on MDLe (Motion Description Language Extended), which specifies upper level movements that the robot needs to perform. The language uses strings of atoms, each containing an interrupt and control quark. The control quark specifies the action to perform, and the interrupt quark determines when to stop executing the control quark.

3.2 The Cricket System

The Cricket System is comprised of units, each which is set to be either a "Beacon" or a "Listener." A beacon is configured to broadcast an ultrasound pulse to be picked up by any listener. Each beacon has a unique ID number. The beacon sends an RF pulse with this number, followed immediately by an ultrasonic pulse. When a listener receives this RF pulse, it starts a timer and when it receives the ultrasonic pulse, it stops the same timer. The listener therefore records Time-of-Flight information from various beacons. Using temperature-corrected speed of sound measurements, the listener can then calculate its distance in centimeters from the Beacon. Correlating at least three Cricket measurements, a position can be determined relative to the coordinate system set up by the beacons.

3.3 Cricket Beacons

The beacons are configured to wait a random amount of time within an interval (defined to be 500 to 1500 milliseconds) between transmissions. If the beacons detect that another beacon has sent, they will then wait for another specified interval (45 milliseconds) before attempting a transmission of their own. This aims to prevent interference between beacon transmissions.

3.4 Cricket Listeners

A Cricket unit in listener mode was attached to the robot through a serial port. Cricket units in beacon mode were free-standing or attached to the ceiling. Each Cricket unit runs code on an Atmel microcontroller using the TinyOS operating system.
The listener was configured with a protocol to gather data from transmitting beacons, form this information into a packet, and send it over the serial port approximately every second. The packet is composed of the following pieces of data:

1. Header: Simply the character "S" to signal the beginning of a packet
2. Num: The number of bytes remaining in the packet
3. Status: A byte where each bit starts for a corresponding beacon 1-8.

For example, ‘01011000’ means beacons 2, 4, and 5 are contained in the packet.

4. Beacon Data: The beacon data, consisting of the distance (in centimeters) between the Listener and the Beacon. The data is ordered in the same manner as Status.

The Cricket System sends bytes as characters. This causes problems when they are converted back to integers. The current packet structure therefore sends each number after “S” as three separate bytes, corresponding to the one, tens, and hundreds place of the number.

4. Calibrating Cricket

The first experiments with the Cricket system aimed to get a sense of the accuracy of the ultrasonic ranging. It is important to note that the Cricket System uses an on-board temperature sensor to correct the speed of sound, which is more accurate then non-temperature-corrected Cricket data. To test the temperature corrected data, one Listener and one Beacon were used. They were initially placed 30 centimeters apart, lying on the floor with their ultrasonic units facing each other. The Cricket range measurement was recorded, and then the Beacon was moved 30.48cm (one floor tile) farther, and so on.

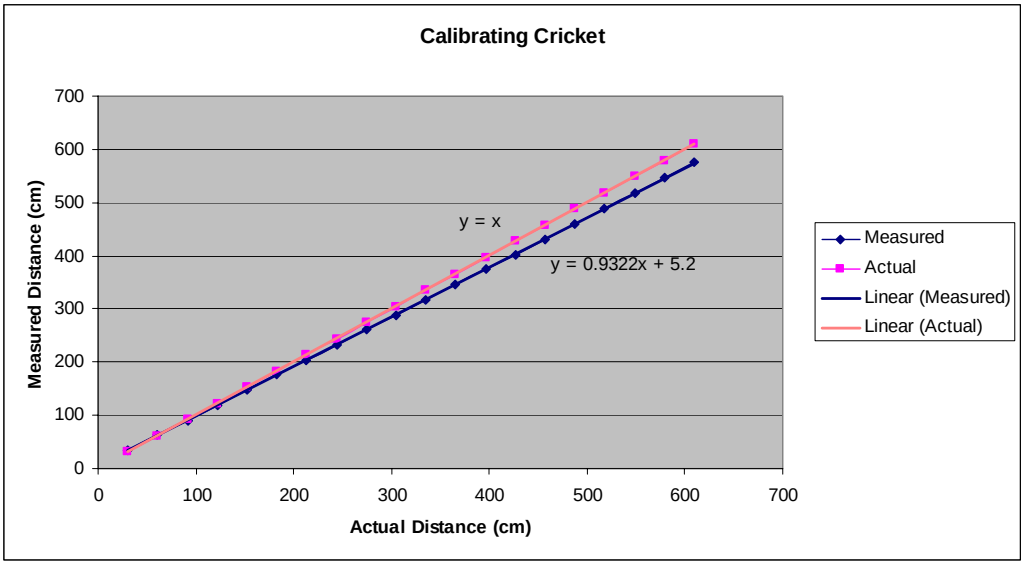


Figure 1- The difference between what the measurements should be (pink) and what they are (blue).

As can be seen from the graph, the measured values originally start off greater than the actual measurements. However, the slope of the measured values is less than the actual. It is likely that the initial offset is due to some sort of timing delay within the system. The lesser slope is due to a calculated speed of sound that is lower than it should be. However, the points are linear, which means the original temperature-corrected model can be modified slightly (in this case, subtracting 5.2 cm and multiplying by the ratio of the slopes, 1.07) to achieve desired accuracy. With the corrected model, each individual

unit of the Cricket System can provide centimeter level accuracy distance measurements, in this experiment verified up to 6 meters.

5. Creating a Coordinate System

Two beacons were placed on the ceiling so their ultrasonic emitters faced directly downwards. They were spaced 183 cm (three ceiling tiles) apart. Establish beacon 1 as the origin of the coordinate system (0, 0), and beacon 2 as 183 cm along the positive x-axis (183, 0).

The Cricket System will record two distance measurements from the locations of the two known beacons. Using the known height of the ceiling, the range measurements can be projected downward onto the plane of the floor. This forms two circles centered at the two beacons with radii equal to the two respective distance measurements. These circles can interest at no points, one point, or two points. No points gives no solution, one point gives a unique solution, and two points gives a non-unique solution. It is usually the two point case. However, the non-unique point is ignored for now by only taking the positive y-coordinate.

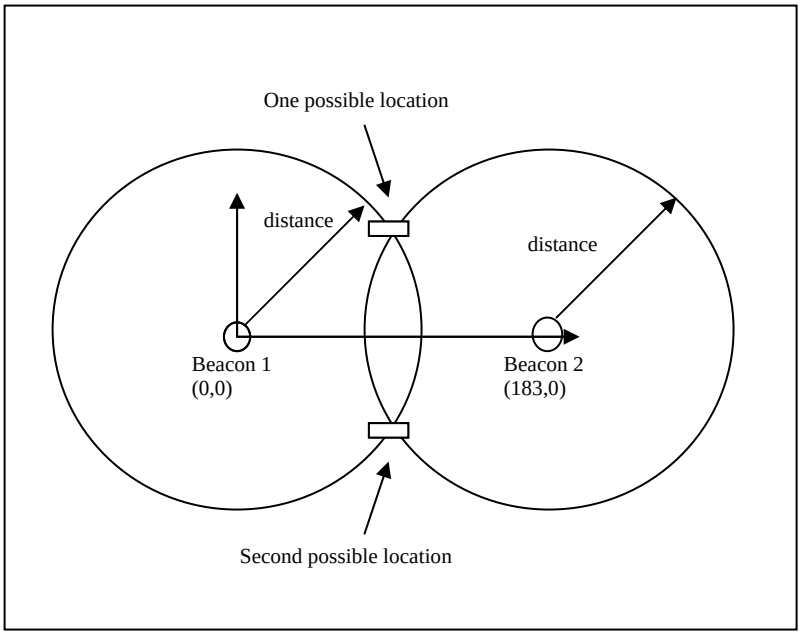
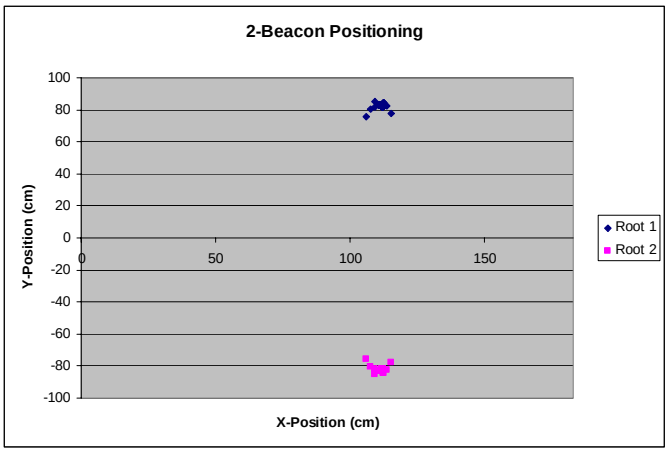


Figure 2. Position estimate using two beacons



First Root
X-Average: 111.265 cm
Y-Average: 82.592 cm

Second Root
X-Average: 111.265 cm
Y-Average: -82.592 cm

Figure 3. Two possible positions of the robot

This experiment has used two beacons to establish a primitive positioning system using Cricket. The system should now be expanded to solve for a unique point and to include more vigorous algorithms to use more than two Cricket measurements.

6. Testing Positioning Algorithms

The Cricket units were set up in a square, with beacons 1-4 at positions (183, 0), (366, 0), (183, 366), (183, 183). A more vigorous algorithm was needed to deal with the new configuration. Three different algorithms were developed and tested for precision and accuracy.

6.1 Algorithm 1, Circle Solving:

This algorithm is an improvement on the circle solving algorithm used in the previous experiment. It now solves the equations of two circles with arbitrary centers (a1, b1) and (a2, b2). Given the equations:

$$(x-a_1)^2+(y-b_1)^2=r_1^2 \quad (1)$$

$$(x-a_2)^2+(y-b_2)^2=r_2^2 \quad (2)$$

It can be shown that y can be solved using the following quadratic equation:

$$(\lambda^2+1)y^2-2(\beta\lambda-\lambda a_1+b_1)y+(\beta^2-2\beta a_1+a_1^2+b_1^2-r_1^2)=0 \quad (3)$$

where:

$$\beta=((r_2^2-r_1^2-d^2)/2+a_1(a_1-a_2)+b_2(b_1-b_2))/(a_1-a_2) \quad (4)$$

$$\lambda=(b_1-b_2)/(a_1-a_2) \quad (5)$$

and the x position is found with the equation:

$$x=\beta-\lambda y \quad (6)$$

This equation fails only in the case where the x coordinates of the beacons are equal. This case can be avoided by simply switching the x and y coordinates, solving the system, and then switching the coordinates back. If the circles do intersect, both points can be derived from the equations. In order to choose the right root, a third beacon is required. Simply calculate the distance from each point to the third beacon and compare it with the beacon range measurement. Whichever point is closer to the actual is the correct point. The algorithm calculates the distance from every pair combination of the beacons. For four beacons, six positions are calculated. These are averaged for the final position estimate.

6.2 Algorithm 2, Bancroft:

The Bancroft Algorithm is an algebraic method used to solve a system of non-linear equations. It is used in GPS, and by including the height of the ceiling as a parameter, can also be used with the Cricket System. See the derivation by Bancroft [1]. Bancroft

provides the position as well as a "common mode error" value for all the beacons, which is a measure of the average error for all the beacons used in the calculation.

6.3 Algorithm 3, Error Minimization Algorithm:

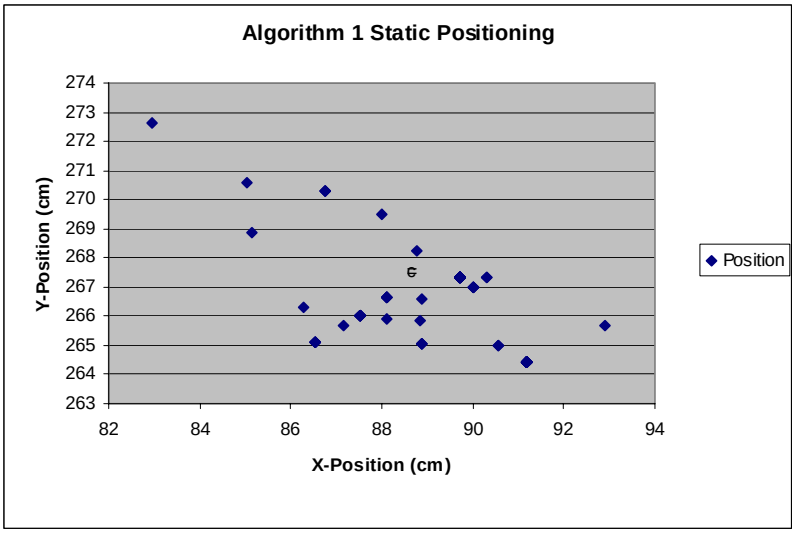
This algorithm uses a slightly different approach to find a position. It is an iterative method that seeks to minimize the error between the actual distances and calculated distances at an arbitrary point. Given a certain square in which the receiver is known to be:

- Divide the square into a grid of nine smaller squares. Place a point in the center of each of these squares.
- Calculate the distances from these virtual points to the Beacons.
- Subtract the actual distance measurements from the virtual distance measurements.
- The absolute value of the sum of this difference for each Beacon at a specific point is the calculated error.
- Theoretically, the point closest to the actual position of the Listener has the lowest error measurement. Take this square and subdivide again into nine squares.
- Repeat until square is less than a centimeter.

This algorithm uses the previous Bancroft Algorithm to achieve an initial position, then uses that point as the center of the 30 cm square used in the algorithm.

Measurements were taken while the receiver was stationary for approximately one minute using the three different algorithms.

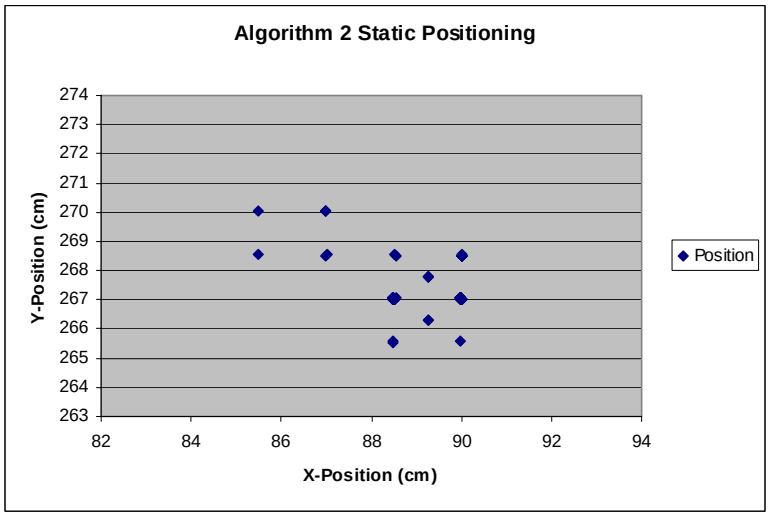
Positioning using Algorithm 1:



Average X: 89.186 cm
Average Y: 266.545 cm
Standard Deviation X: 1.721 cm
Standard Deviation Y: 1.635 cm

Figure 4. Shows the distribution of measurements for a single position

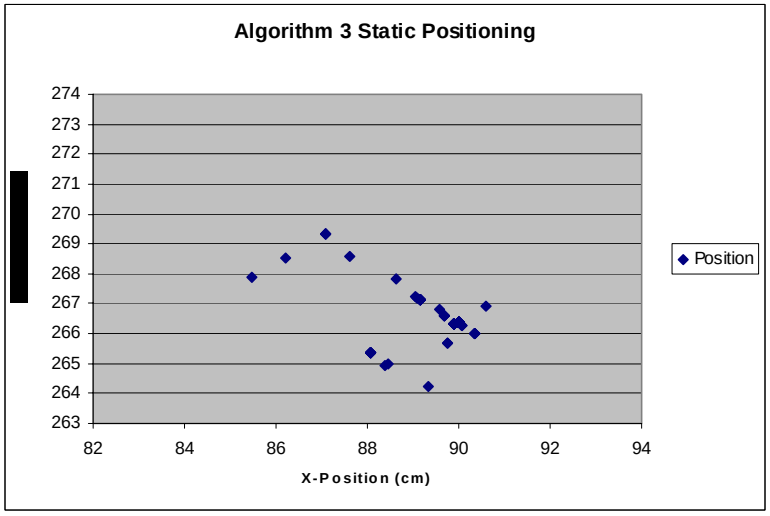
Positioning using Algorithm 2:



Average X: 89.185 cm
Average Y: 267.388 cm
Standard Deviation X: 1.049 cm
Standard Deviation Y: 0.879 cm

Figure 5. Shows the distribution of measurements for a single position

Positioning using Algorithm 3:



Average X: 89.451 cm
Average Y: 266.484 cm
Standard Deviation X: 0.977 cm
Standard Deviation Y: 0.857 cm

Figure 6. Shows the distribution of measurements for a single position

The data illustrates the development of the three algorithms, culminating in the error minimization algorithm using Bancroft. Algorithm 1 works well for two or more beacons, but has the problem of not being able to solve for the position if the circles do not have a point of intersection. Algorithm 2 is improved in that it provides a measure of error. However, it has difficulties solving for the non-unique case with two beacons. Algorithm 3 combines the stability of the first two algorithms but provides an iterative

approach that allows a position even if the circles are unsolvable. The standard deviation shows that each algorithm improved on the one before.

7. Static Measurements

In order to run the next set of experiments, a larger coverage area was necessary in order to get a better picture of the Cricket system. More beacons were placed on the ceiling to make up the following configuration:

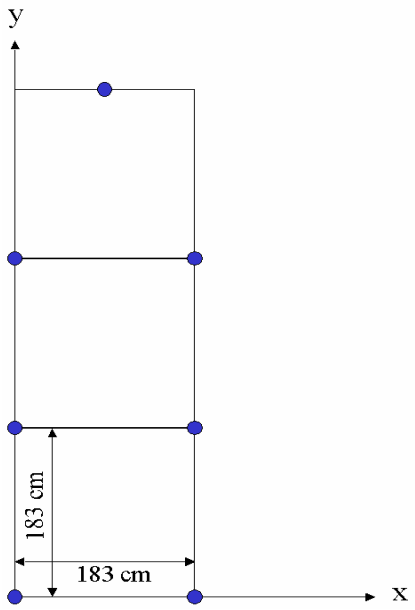


Figure 7. Final beacon placement. Seven circles denote the locations of the beacons

Since it was not certain that the performance of the system was consistent in different places on the coordinate system, one more static experiment was done. This time, only the error minimization algorithm was used, because it has been proven to have the best performance. The following measurements were taken at 30 sec intervals during a single run, only recording the position when the robot was stationary. Between these intervals, the robot was instructed to move forward for 50 cm.

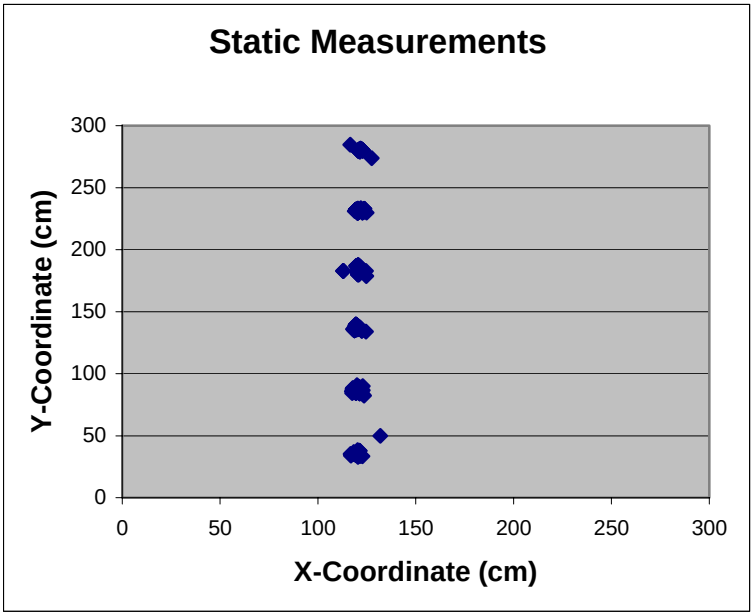


Figure 8. Static measurements done using all seven beacons

Point	Average X (cm)	Standard Deviation X (cm)	Average Y (cm)	Standard Deviation Y (cm)
1	120.3	1.69	35.9	2.01
2	120.2	1.35	86.4	1.60
3	120.2	1.12	137.1	1.36
4	120.1	2.22	183.7	2.43
5	120.7	1.41	231.4	0.96
6	121.7	1.62	280.2	1.47

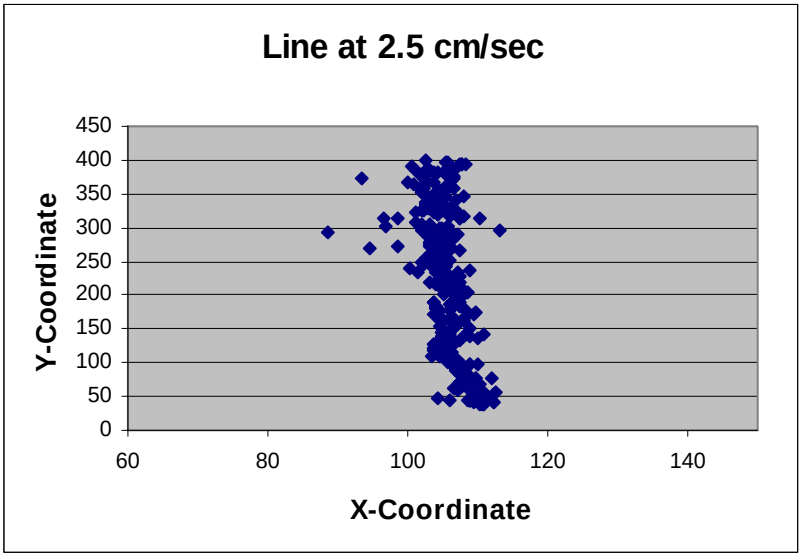
Figure 9. Table summarizing the results of the static measurements

The experimental results show that according to the Cricket system, the robot traveled straight, keeping its x-coordinate almost constant over a distance of two and a half meters. A more important result is that the standard deviation for both x- and y-coordinates is almost the same in each case, and only varies slightly between the different points in space.

8. Dynamic Measurements

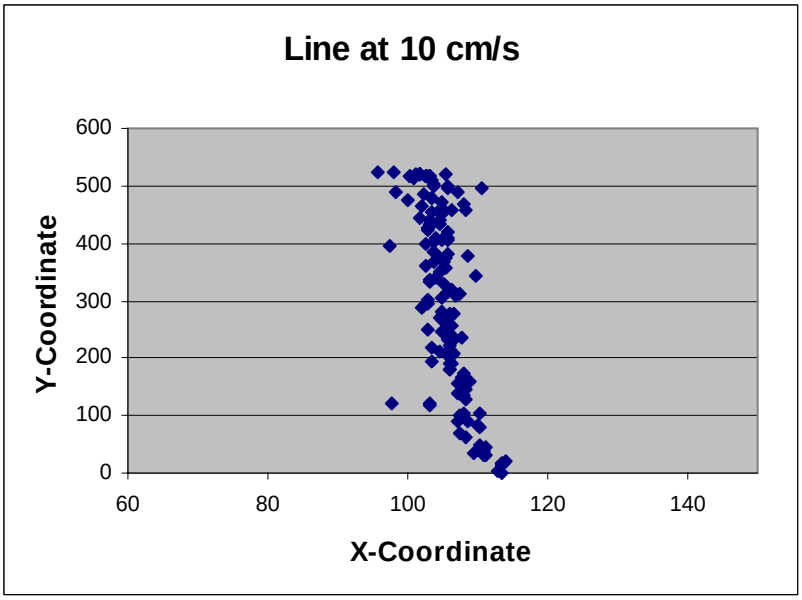
It was important to find out whether the performance of the Cricket system would change if the measurements were taken while the robot was in motion. In the following two experiments the position data was recorded while the robot was moving straight at 2.5 cm/s and 10 cm/s. Although we are unable to show that the actual path of the robot was a perfect line, it was assumed that the robot was able to stay close enough to the desired

line. In order to figure out how good the data is, a line was fit using all the points, and for each point, the distance to that line was calculated. These small distances were then used to calculate their average and standard deviation.



Average error: 1.75 cm
Standard Deviation: 1.75 cm

Figure 10. Position measurements taken when the robot was moving straight at 2.5 cm/s



Average error: 1.67 cm
Standard Deviation: 1.69 cm

Figure 11. Position measurements taken when the robot was moving straight at 10 cm/s

The graphical representation of the data shows that the linearity of the data decreases with the speed. However the statistical data shows the opposite – the standard deviation for the 10 cm/s experiment is smaller than for the 2.5 cm/s one. What needs to be taken into account is that the standard deviation was found from the line fit through all the data

points. There is no guarantee, though, that this line represents the true path of the robot. The unexpected curvature in the Figure 11 can be explained by the fact that the beacons do not send their pulses at the same time. Therefore if the robot is in motion, each distance measurement is done from a different position. Correlating these measurements without a model, that would take into account robot's velocity, is likely to produce less accurate results. One solution for this problem would be managing the number of beacons used to calculate the position. The smaller the number, the less time would pass in the process of collecting data, leading to better position estimates.

9. Tracking a circle

In another experiment, the robot was instructed to move with a constant speed of 2.5 cm/s and rotational speed of 0.033 rad/s, theoretically going a circle with a 1.5 meter diameter. However, since the exact center of the circle on the data plot is rather difficult to find, no statistical analysis was done. The accuracy of measurements, though, should not depend on the type of curve that the robot is tracking, whether it is a line or a circle.

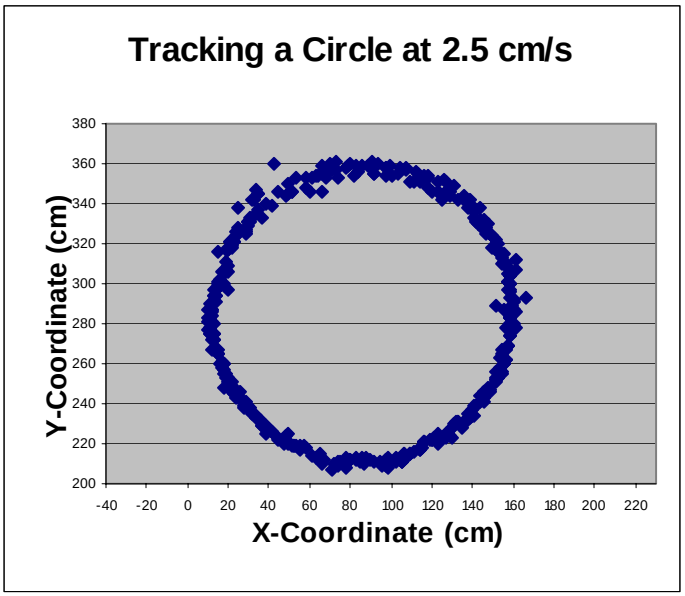


Figure 12. Position measurements taken when the robot was moving in a circle with a diameter of 1.5 meters

From the Figure 12, one can see that the theoretical diameter of 1.5 meters was achieved. It can then be expected to see similar behavior from the Cricket system while moving along any other arbitrary curve.

10. Using Two Listeners for Orientation

Besides the position of the robot, another useful piece of information is its orientation in the Cricket coordinate system. This can be calculated by taking measurements from two different Listeners placed on the front and back end of the robot (45 cm in experiments). Given the position of each end of the robot, one can create a direction vector and

determine its orientation with respect to the Cricket coordinate system. In order to collect two set of data, two separate instances of the Cricket module are run. These modules alternate, calculating their positions. When both modules have position data, they are brought together to calculate orientation by constructing a triangle using the two positions and solving for one of its angles.

The following graph shows the robot at three different orientations for an interval of time:

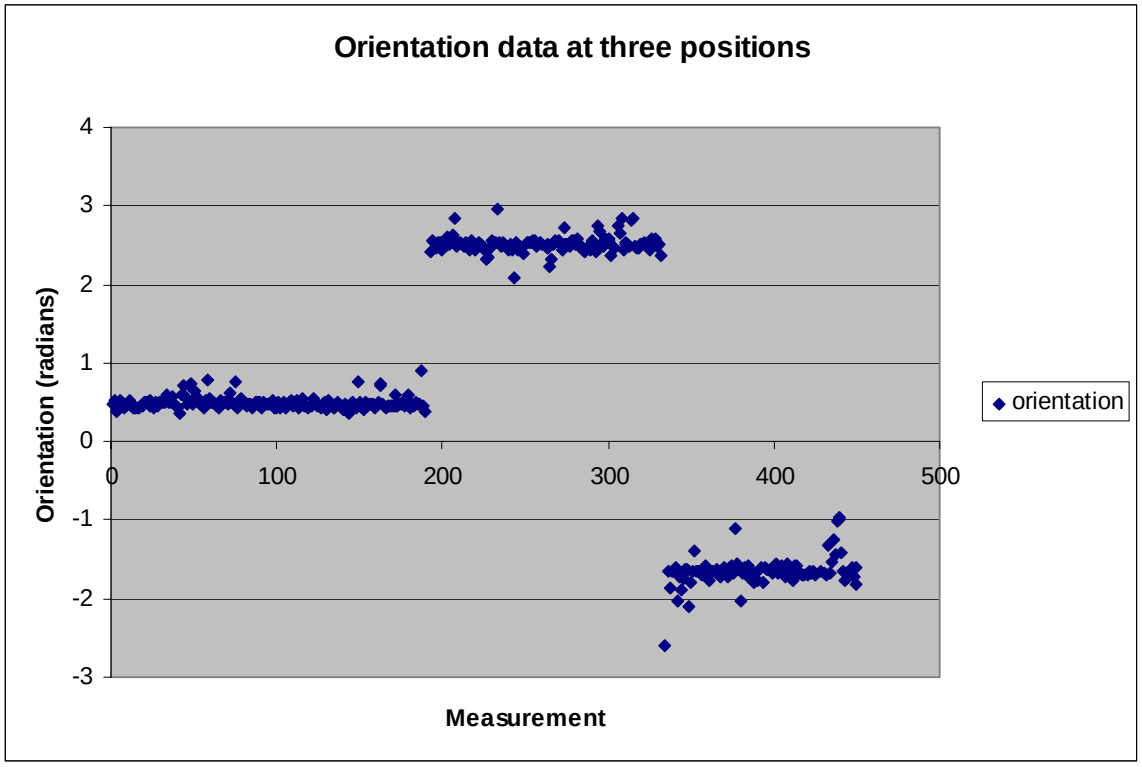


Figure 13. Orientation measurements taken at three different angles

First Orientation:
Average: 0.492 Radians
Standard Deviation: 0.075 Radians or 4.344 Degrees

Second Orientation:
Average: 2.514 Radians
Standard Deviation: 0.105 Radians or 6.064 Degrees

Third Orientation
Average: -0.638 Radians
Standard Deviation: 0.171 Radians or 9.844 Degrees

The next graph shows the robot while it spun in a clockwise circle from an initial heading of zero radians:

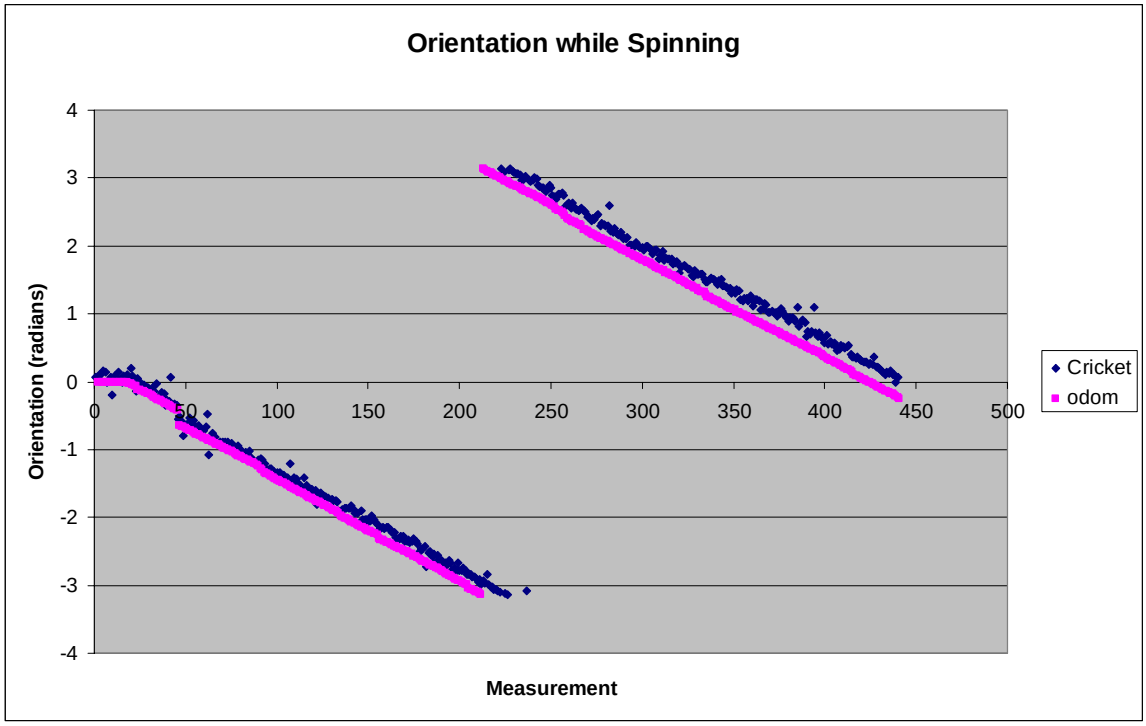


Figure 14. Cricket and odometry orientation while spinning clockwise

As was expected, the orientation data was not as accurate as the positioning data. This makes sense considering that both listeners introduce a source of error. If either listener does not receive data, the orientation cannot be calculated. If one listener receives faulty data, the orientation can vary widely. Nevertheless, for indoor use where orientation is not crucial, this method of orientation may prove sufficient. Unlike relative positioning methods, errors do not accumulate over time.

11. Boundary Tracking

One possible application of a positioning system such as Cricket is using it in conjunction with a navigation algorithm to give a robot desired behavior in certain environments. One such algorithm has been developed in the Intelligent Servosystems Laboratory. The goal of this algorithm is to avoid a collision with an obstacle by maneuvering along its boundary without changing the speed of the robot. The main idea is to keep the robot's heading parallel to the tangent vector at the closest point on the obstacle, thus theoretically tracing out the boundary of the object, only a certain distance away [2].

In order to calculate the robot's instantaneous behavior, the algorithm requires feedback from several components of the system. The needed information consists of:

- Position of the robot
- Its orientation
- Distance to the obstacle
- Curvature of the obstacle at the closest point

Position estimates can be easily obtained using the robot's odometer (if one is present), however accumulating error in such estimates can be significant without any feedback. Cricket provides such feedback, eliminating any long-term performance degradation effects inherent in odometry. Even though the Cricket position estimates are obtained relatively infrequently (ranging from 1 to 5 times per second, depending on the number of beacons) and are not as accurate as the odometer on small intervals, the error in such estimates does not change with time. In addition, since the odometer data is available continuously, the robot can make use of it between the updates from the Cricket module and calculate the relative displacement from the last Cricket update.

Knowing orientation is important because, based on this information, the algorithm provides such feedback so that the robot's heading aligns with the tangent vector on the obstacle's boundary. The robot may be able to provide its current heading with respect to the orientation at the startup (by integrating the angular velocity over time), however, the same exact problems arise as with the position estimates based on odometry. For this reason, use of an absolute positioning system such as Cricket is preferred. (See section on orientation for more details on obtaining the orientation data from Cricket **)**

In the real world, the position of obstacles in someone's path is usually unknown a priori. Therefore a robot would need to have some kind of sensing device, such as a laser scanner or a sonar, to try to figure out the required information about the object. However, because this project is not specifically concerned with such details, virtual obstacles were used. Since it may not be possible to describe an arbitrary obstacle using a continuous function of the x- and y-position, a set of discrete points was used to trace out its boundary. The number of points one should specify depends on the shape of the obstacle, but as a rule of thumb, more information should be provided for those segments where the curvature changes quickly.

11.1 Curvature and Distance to the Obstacle

Curvature, as one of the parameters of the boundary-tracking algorithm, needed to be determined from that set of points describing the obstacle's boundary. There are several approaches to curvature estimation and the one used in these experiments is based on fitting a circle through the three points which are closest to the robot. Finding the distance to the boundary requires only few additional steps. The whole process consists of the following steps:

- Getting a position estimate from Cricket. This is done by getting a reference to the Cricket module and accessing its public variable that stores the required information.
- Selecting three closest points based on that position. All the coordinates are stored in a text file, so in this step, the algorithm simply goes through all of them, selecting the three closest ones to the Cricket's position estimate.

- In order to find the curvature, the radius of a circle that passes through all three points was estimated in the following way:

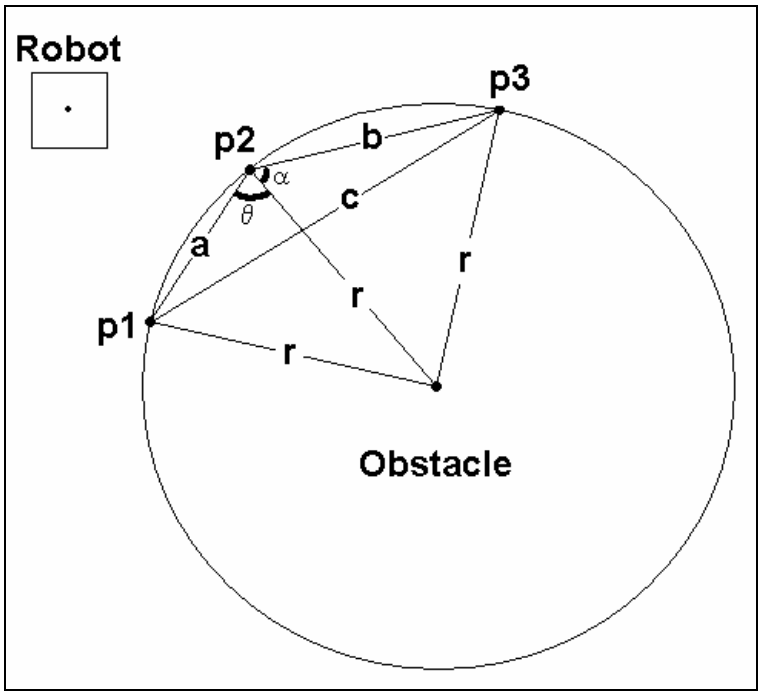


Figure 15. Fitting a circle through the three points and finding its radius.

Using the Law of Cosines:

$$r^2 = a^2 + r^2 - 2ar\cos\theta \quad (7)$$

$$r^2 = b^2 + r^2 - 2br\cos\alpha \quad (8)$$

$$a^2 = 2ar\cos\theta \quad (9)$$

$$b^2 = 2br\cos\alpha \quad (10)$$

$$a = 2r\cos\theta \quad (11)$$

$$b = 2r\cos\alpha \quad (12)$$

$$\beta = \alpha + \theta \quad (13)$$

$$c^2 = a^2 + b^2 - 2ab\cos\beta \quad (14)$$

$$\beta = \cos^{-1}\frac{\boldsymbol{c}^2 - \boldsymbol{a}^2 - \boldsymbol{b}^2}{-2\boldsymbol{ab}} \tag{15}$$

$$\frac{\boldsymbol{a}}{\boldsymbol{b}} = \frac{\cos(\beta - \alpha)}{\cos\alpha} \tag{16}$$

$$\boldsymbol{r} = \frac{\boldsymbol{b}}{2\cos(\beta - \alpha)} \tag{17}$$

To solve for alpha, an iterative algorithm was used to approximate alpha such that the two sides of the Equation 16 differ by no more than 0.01. Using the Newton's method, the desired accuracy was achieved in less than 10 iterations. Once the approximate radius is found, curvature is simply $1/\boldsymbol{r}$.

- Estimating the distance to the obstacle can also be done in various ways. For example, the distance from the closest point can be taken. However, if the radius of the circle that passes through the points is known, then we can find the location of the center of this circle. The distance to the obstacle then simply is the distance to the center minus the radius. The coordinates of the center were found the following way: Two circles with the known radius can be constructed with centers being any two of the three given points. These two circles intersect in two places, one of them being the center of our interest. [3] The third (unused) point can be used to pick the correct one.

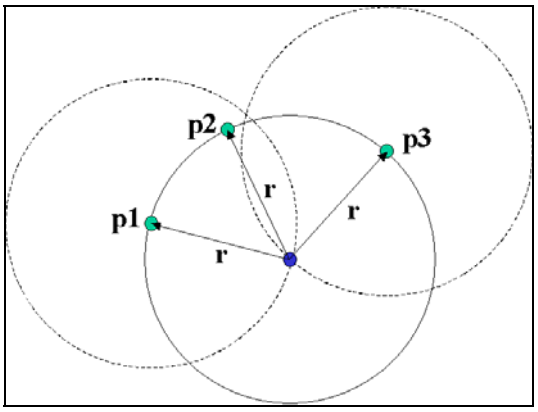


Figure 16. Finding the center of the circle that passes through three points.

11.2 Quarks and switching behavior

The structure of MDLe allows the robot to switch behaviors when certain interrupts are activated. This ability is useful for integrating obstacle avoidance with a set of other tasks, allowing the robot to change the mode of operation if an obstacle is in a close proximity. An interrupt quark was created to keep track of the obstacles and to switch to an obstacle avoidance mode if one is detected. The quark constantly monitors the heading of the robot and calculates how close to the obstacle the robot would come if it kept moving in that direction. If this distance is smaller than desired, the interrupt quark terminates the operation of the current action quark and switches to the appropriate collision avoidance action quark.

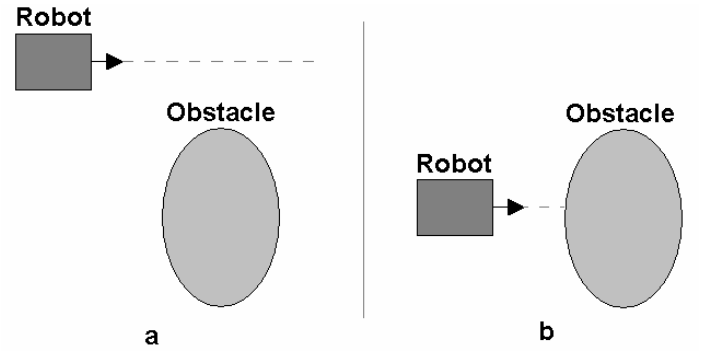


Figure 17. Robot will continue moving straight without the interrupt quark triggering (a). Interrupt quark will switch the behavior to obstacle avoidance (b).

Since the robot only needs to worry about the obstacles that are relatively close to it, an extra parameter is set to prevent robot from entering the obstacle avoidance quark when the object is far away, but the robot is heading straight for it. This is important because the actual path of the robot may eventually steer it away from the obstacle, making the collision avoidance mode unnecessary.

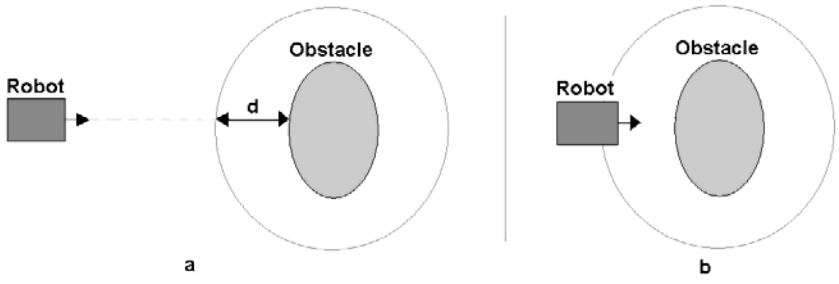


Figure 18. Interrupt quark is not activated when the distance to the obstacle is greater than d (a). When robot finally gets closer, the collision avoidance behavior will turn on. (b)

11.3 Boundary Tracking Experiment.
The following experiment demonstrates the concepts mentioned above. A circular virtual obstacle with a radius of 30 cm was placed at (90,270). The robot was first instructed to turn and go towards the obstacle until it got closer to it. Then the obstacle avoidance algorithm was activated, providing control feedback so that the robot maintained a distance of 30 cm from the boundary. The algorithm is designed so that the robot tracks its boundary forever. In order to complete the task, another interrupt quark would be required to terminate the obstacle avoidance algorithm once it is safe to continue moving towards the original destination.

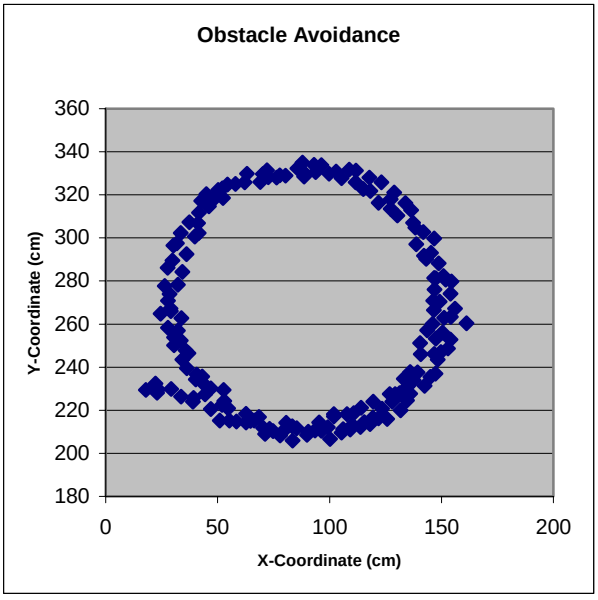


Figure 19. Graphical representation of the results of boundary tracking.

All the parts of the experiment went exactly as described in the procedure and the robot managed to drive around in a circle several times before it was stopped. This experiment demonstrates the versatility of the platform, where various modules are used to acquire information about the environment and MDLe is used to determine the appropriate mode for the robot in a given situation. Cricket's ability to provide absolute positioning data is essential for verifying whether a control law that is being tested is performing as designed.

12. Cricket as an Active Sensor Network

The original Cricket System was a passive network. The beacons on the ceiling were set to fire randomly (within a specified time slot), only waiting to avoid collisions if two beacons were trying to send their chirps at once. The listener did not communicate with the beacons in any way. In order to increase efficiency and add feedback capability to the sensor network, the code was changed so that the listener had the ability to control the timing of each beacon.

The listener was programmed to send out a radio packet, acting as a sync pulse, containing a string of eight bits, again corresponding to the eight beacons. Each beacon, after receiving the string, finds out whether the character in that string, which corresponds to its beacon number, is a 0 (do not send) or a 1 (send). If the beacon has been instructed to send, it counts the number of beacons sending, and its position in that list of active beacons. Each beacon then waits 100 milliseconds from the sync pulse times its place in the queue. For example if beacons 2,3 and 7 are instructed to send, then beacon 2 will send at 0 milliseconds, beacon 3 at 100 milliseconds, and beacon 7 at 200 milliseconds. All randomness is taken out of the system. In order to determine which beacons should be active, previous position estimate is used to calculate which beacons should be "visible" from that location. When the listener is provided such information, the sync pulse is sent out.

There are several advantages to the updated version of the Cricket code. First, it is more efficient in its use of time. When all the beacons fire randomly, it can be difficult to decide when the listener should send out a complete packet containing distance measurements. It took up to a second for the packet to be sent if all eight beacons were firing, but, in fact, roughly only half of that data was valid because some beacons were always out of range of the ultrasonic signal. In an active network, the listener can turn these unwanted beacons off. If it is only receiving data from two beacons, it can send out a packet every 200 milliseconds. This cuts down on unneeded erroneous distance measurements (by simply not taking them) and prevents the listener from waiting for data to arrive that isn't necessary. Receiving data more quickly also means that it should be slightly more accurate, due to the fact that the range measurements are taken within shorter intervals of time.

The Cricket units were mounted with antennas in order to boost the quality of radio communication in the lab. However, it was discovered that when the beacons were in close proximity to each other, a small number of them would fail to receive radio packets sent by the listener. When the antennae were removed and the beacons spread farther apart, this did not occur. It is highly likely, however not conclusive from our experiments, that the radio receivers in the Cricket units affect each other's ability to communicate. The listeners are currently equipped with antennas while the beacons are not.

13. Conclusion

The results of these experiments present an overview of the performance of Cricket as an absolute positioning system for control applications. Experiments were done to test various aspects of the system in both hardware and software domains. The position estimates showed a much higher accuracy than one would expect from any typical GPS system even if it were available indoors. This makes Cricket a useful tool in a robotics laboratory, providing positioning data free of long term accumulating error effects, which are inherent in relative positioning methods. Because of its versatility, the system can be configured for various laboratory environments and types of experiments that would be carried out. This provides an ability to make use of Cricket most efficiently by adapting it to a particular setting.

13.1 Future work

Improving dynamic measurements
Since the performance of Cricket degraded with the speed of the robot, it may be necessary to introduce a more sophisticated position estimation algorithm, which would take into account the robots motion.

Detecting and avoiding multiple obstacles
Simple experiments such as tracking a boundary of a single circular obstacle are easy to carry out, but are not very practical. In order to be more realistic, more complex virtual environments should be created, introducing variations in number of obstacles, their shapes and possibly even a danger level associated with each entity. This would require the interrupt quark, which monitors the positions of the obstacles, to do a little more processing than just looking for three closest points from a list, because those points may belong to different objects.

Coordinating multiple robots
Using Cricket as an active sensor network, discussed in this paper, for navigation of multiple robots may introduce some ambiguities. Since a robot has the ability to control the behavior of the beacons, putting another listener in the same environment would "confuse" the beacons. Depending on the desired types of experiments the ambiguity should be resolved by either separating the system into several sections, or only allowing one listener to control the beacons (in cases when formations of robots are used).

GPS and Cricket
Achieving autonomous operation in various settings is a goal of many robotics research laboratories. Therefore, seamless indoor and outdoor positioning using GPS and Cricket in their appropriate environments may be an interesting topic for future researchers. A robot may rely on GPS measurements while traveling outside and then switch to Cricket upon entering an equipped indoor facility. Global positioning can still be achieved indoors because the location of each such facility can be described in GPS coordinates.

14. Bibliography

[1] S. Bancroft. "An Algebraic Solution of the GPS Equations," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-21, No. 7, January 1985

[2] F. Zhang. "Boundary following using gyroscopic control," Institute for Systems Research,University of Maryland College Park, MD 20742, USA

[3] Circle Center's Cartesian Coordinates. <http://mathforum.org/library/drmath/view/53179.html> Accessed August 8, 2005.