SENSOR BASED LOCALIZATION FOR MULTIPLE MOBILE ROBOTS USING VIRTUAL LINKS

A Thesis

by

ANDREW JOHN RYNN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2003

Major Subject: Mechanical Engineering

SENSOR BASED LOCALIZATION FOR MULTIPLE MOBILE ROBOTS USING VIRTUAL LINKS

A Thesis

by

ANDREW JOHN RYNN

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Sooyong Lee
(Chair of Committee)

Nancy Amato
(Member)

Craig Smith
(Member)

Dennis O'Neal
(Head of Department)

August 2003

Major Subject: Mechanical Engineering

ABSTRACT

Sensor Based Localization for Multiple
Mobile Robots Using Virtual Links. (August 2003)

Andrew John Rynn, B.S., Texas A&M University
Chair of Advisory Committee: Dr. Sooyong Lee

Mobile robots are used for a wide range of purposes such as mapping an environment and transporting material goods. Regardless of the specific application, the navigation of the mobile robot is usually divided into three separate parts: localization, path planning and path execution. Localization is the process of determining the location of the robot with respect to a reference coordinate system. There are many different approaches to localizing a mobile robot which employ a wide variety of sensors.

The objective of my research is to develop a method for the localization of multiple mobile robots equipped with inexpensive range sensors in an indoor environment. Each mobile robot will be equipped with a rotating infrared sensor and a rotating CMOS camera. The multiple mobile robot system will be treated as a linked robot for localization.

The proposed localization method is verified via both simulation and experiment. Through the use of the virtual link length and relative heading information, a system of mobile robots can be effectively localized using detected environmental features.

To my family

ACKNOWLEDGMENTS

First and foremost, I would like to thank my parents for their continuous support throughout my lengthy academic journey. Without their support as well as that of my brother and sisters the path would have been much more arduous.

Dr. Sooyong Lee has been of tremendous assistance throughout my unexpectedly lengthy time in graduate school. He has provided me with much needed direction and assistance throughout the course of my research. Initially, he served to pique my interest in the field of robotics through his high caliber of instruction. I would like to thank him for tolerating me for the last three years.

I would like to thank Dr. Craig Smith and Dr. Nancy Amato for their membership on my thesis committee and all the time which that entailed. They have provided valuable input in the clarification of integral portions of this thesis.

I also must thank David Sorensen and Waqar Malik for their continuous assistance throughout the course of my research. They have both been very valuable as sounding boards and always assisted me whenever I asked. Waqar was a tremendous help in the acquisition of usable experimental data which serves to strengthen the overall thesis. Both David and Waqar have been great to work with and have provided for many hours of entertainment occasionally interrupted by some hard work.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Mobile robots are used for a wide range of purposes such as mapping an environment and transporting material goods. Regardless of the specific application, the navigation of the mobile robot is usually divided into three separate parts: localization, path planning and path execution. Localization is the process of determining the location of the robot with respect to a reference coordinate system. There are many different approaches to localizing a mobile robot which employ a wide variety of sensors.

The objective of my research is to develop a method for the localization of multiple mobile robots equipped with inexpensive range sensors in an indoor environment (meaning that the environment is known and contains straight lines and vertices). Through the use of multiple robots, the limited range of inexpensive range sensors can be overcome and localization becomes possible. As an added benefit, the use of multiple robots allows for more efficient task completion. Additional robots could significantly reduce the time required to map an environment or search for an object.

Each mobile robot will be equipped with a rotating infrared sensor and a rotating CMOS camera. The multiple mobile robot system will be treated as a linked robot for localization. This requires an unobstructed line of sight between each pair of robots forming a "link". An individual robot cannot extract any useful information solely from individual local sensing due to the limited sensing range. The range of the infrared sensor is small and susceptible to error. In order to facilitate collaboration and exploit geometry, a team must be able to extract salient information from the collective sensing of all its members. By collecting and fusing sensor information

The journal model is *IEEE Transactions on Automatic Control.*

from multiple, distinct positions, the effective resolution of the space can be improved beyond that of the individual measurements. A method will be developed to localize multiple mobile robots in an indoor environment. It is tested in a simulation and verified via experiment. Thus, by considering the sensor reading generated by all the robots as one big virtual sensor, it is successfully shown that the robots can be individually localized.

CHAPTER II

BACKGROUND

Omnidirectional cameras are used in [1] and [2] for relative localization of multiple mobile robots via triangulation. In [3] a method is proposed for the absolute localization of multiple mobile robots using omnidirectional cameras. A method to localize a single robot using an omnidirectional camera and a geometrical matching method based on the subdivision of the robot's field of vision is presented in [4]. Omnidirectional cameras are very useful for locating known shapes that contrast with the environment. However, it is not possible to effectively detect corners in a typical room where the walls are all the same color.

Ultrasonic range sensors are used to localize a single mobile robot in an indoor environment in [5], [6] and [7]. A team of miniature mobile robots is localized using ultrasonic range sensors in [8]. Ultrasonic range sensors are very inexpensive and can effectively detect walls and corners. However, they have a limited effective range.

The localization method used in [8] was effective in simulation, but the ultrasonic range sensors' close proximity to the floor created some difficulty in processing experimental data. The localization method uses dead-reckoning in conjunction with the sensor data. The developed method has a fair amount of accuracy and is inexpensive to implement. One drawback to the approach is that the initial position of each robot must be known.

Multiple robots equipped with laser range sensors are used to create a three dimensional map of an environment in [9]. Laser range sensors are used in conjunction with an omnidirectional video camera in [10]. The sensor data is compared to the global environment map for localization. Laser sensors give very accurate readings of the environment and have much longer effective ranges than infrared or ultrasonic sensors. However, they are far more expensive.

A method to localize a mobile robot in a perfectly known indoor environment with ideal sensors is presented in [11]. The environment is divided into visibility sectors which represent the region of points from which the same number of environmental vertices are visible. From the range sensor scan data, a unique label derived from the critical points of the data is used to determine the current visibility sector for the robot. Once the visibility sector is determined, the features detected in the sensor scan can be placed in the environment. Visibility sectors are used in conjunction with consecutive sensor scans for dynamic obstacle detection in [12]. In this method, it is required that a local maxima (which indicates a concave corner) be detected by the robot's sensor for localization. If a corner is not found, the robot can not be localized.

In [5], the visibility sector localization approach is extended by eliminating the assumptions of a perfectly known environment and ideal sensors. Relaxed visibility sectors are employed to localize the robot. They are found by computing the visibility sectors and merging them using heuristic rules. By using relaxed visibility sectors, the robot can be localized in a greater portion of the environment. Using the relaxed visibility sectors, it is no longer necessary to detect a concave corner for localization. In [6], a method is proposed to integrate the relaxed visibility localization scheme into a navigator. The above localization methods are further extended in [13] where scannable visibility sectors are used to enable a larger localizable area through the use of relaxed localization.

CHAPTER III

LOCALIZATION METHOD

A software environment in which range sensor scans can be simulated for multiple robots has been developed. The mobile robot localization method treats the robots as a linked system whereby the distance between the robot centers and relative headings are known. This information can be found via a rotating camera or by maintaining the distance between the robots within the range sensors' effective range. This necessitates a line of sight between each successive robot in the system.

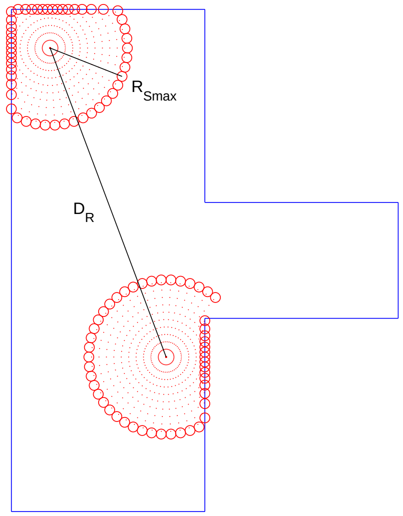

Fig. 1. T-Shaped Environment with Two Robots

The distance between the robots will be found via a rotating camera and is represented as D_R. The limits of the range sensor are $R_{S_{min}}$ and $R_{S_{max}}$. The effective range of the sensor used to determine the distance between the robots is given by $R_{D_{min}}$ and $R_{D_{max}}$. It will be assumed that $R_{S_{max}} < D_R < R_{D_{max}}$ and $D_R \geq R_{D_{min}}$. Figure 1 shows a T-shaped environment containing two robots; each robot can only detect a small portion of the environment.

The initial step of the localization process is to process the sensor scan data. It is necessary to locate any corners and saturation points in the data. Using this information, the Cartesian offsets can be determined from the detected environmental features. Next, a list of potential locations is generated for each robot. Using the relative distance and heading information, potential location pairs for the robots can be determined. Once a potential location pair has been found, it is checked for a line of sight; if one exists, the absolute heading for each robot is calculated.

The localization method was developed using two robots. However, it is easily extended to work for any number of robots. This is discussed in chapter IV and a simulation example for a three robot system is shown in chapter VI.

Once the localization method is proven successful in simulation, it is tested experimentally using an inexpensive range sensor and an inexpensive CMOS camera. A rotating range sensor and camera are placed in prescribed positions and orientations in the environment. The environment is scanned and the resulting data is used to determine the configuration of each robot. The localization results are then compared to the known locations and headings.

A. Localization With Two Corners

The following definitions are used in this section.

wall 1 Environment wall clockwise of corner.

wall 2 Environment wall counter-clockwise of corner.

$\mathbf{vec_{wall_1}}$ Vector along wall 1 in the counter-clockwise direction.

$\mathbf{vec_{wall_2}}$ Vector along wall 2 in the counter-clockwise direction.

$\mathbf{vec_1}$ Vector normal to vec_{wall_1}.

$\mathbf{vec_2}$ Vector normal to vec_{wall_2}.

$\mathbf{vec_{corner}}$ Vector from robot center to corner.

If each robot detects a corner in its sensor scan, a limited number of possible locations for each robot can be determined by determining the Cartesian offsets from the corner in the sensor scan for each robot. The relative heading angle, ϕ is defined as the counter-clockwise angle from the heading of the first robot to the vector connecting the two robots. A second angle, γ, is defined as the counter-clockwise angle from the robot heading to the located corner. Figure 2 shows two robots located in a T-Shaped Environment and their respective headings. It also illustrates ϕ and γ for both robots.

To determine the Cartesian offsets from the corners, the located corners were divided into two categories. The first group contains those corners for which both walls forming the corner are found in the sensor scan. This condition holds for all concave corners and for convex corners in which the corner does not block the robot's line of sight to a segment of the environment. The second group consists of convex corners for which only one wall that forms the corner is detected.

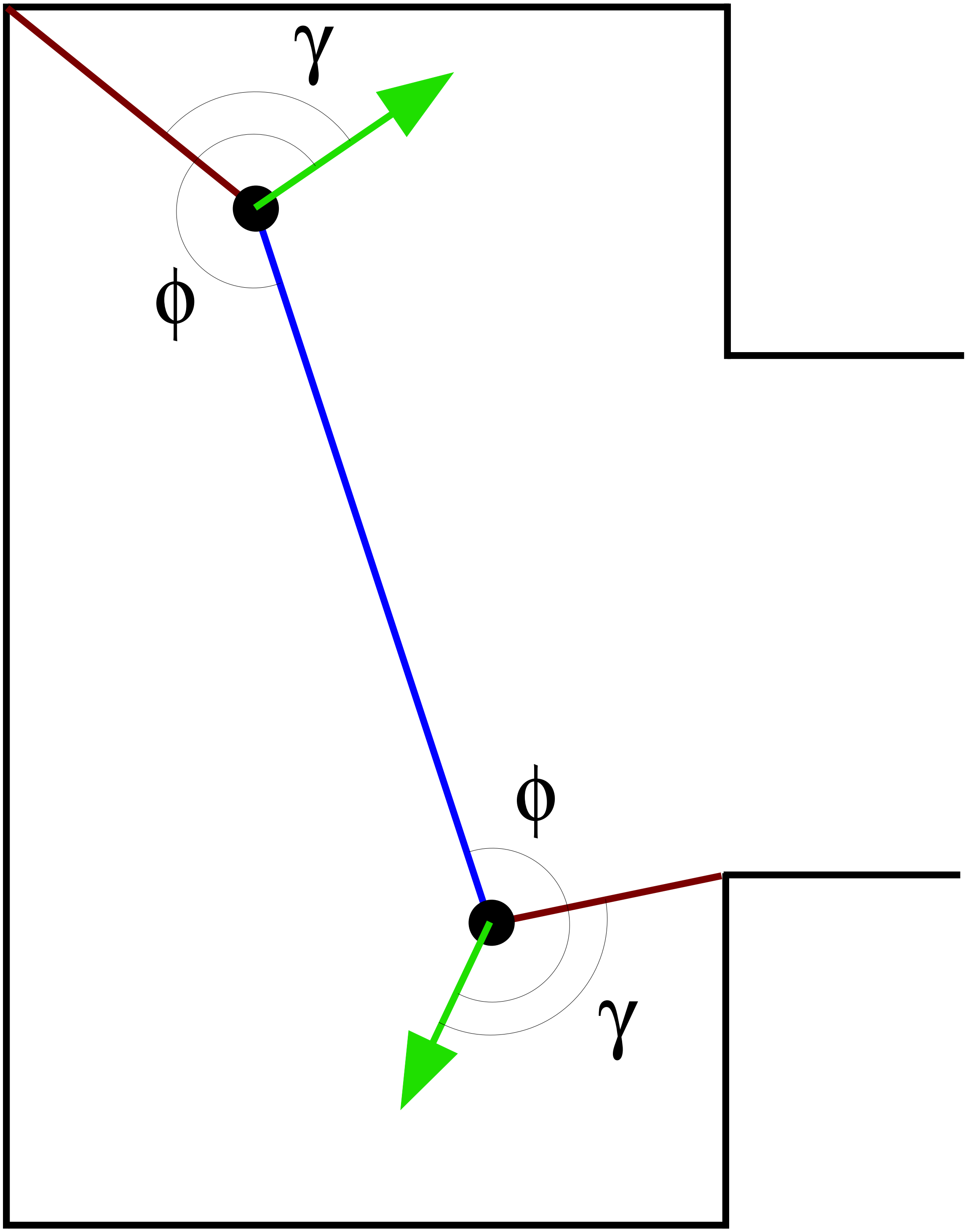

Fig. 2. Two Robots in T-Shaped Environment

1. Determining the Offset Distances for a Corner with Two Detected Walls

Assumptions

- The wall vectors can be reliably determined for both walls forming the corner.
- From the sensor scan, the saturation points (the points at which the sensor response transitions to and from its maximum effective range) can be reliably determined.

The location of the corner is given by equation 3.1, where θ is defined relative to the sensor scan. The saturation points, given by equations 3.2 and 3.3, are used in conjunction with the located corner to define the two walls which form the corner. The wall vectors are defined using equations 3.4 and 3.5.

$$\begin{aligned}
\text{corner} &= \begin{bmatrix} \text{r(cornerLoc)}\cos(\theta(\text{cornerLoc})) \\ \text{r(cornerLoc)}\sin(\theta(\text{cornerLoc})) \end{bmatrix} && (3.1) \\
\text{sat}_{\text{pt}_1} &= \begin{bmatrix} \text{r}(\text{satLoc}_1)\cos(\theta(\text{satLoc}_1)) \\ \text{r}(\text{satLoc}_1)\sin(\theta(\text{satLoc}_1)) \end{bmatrix} && (3.2) \\
\text{sat}_{\text{pt}_2} &= \begin{bmatrix} \text{r}(\text{satLoc}_2)\cos(\theta(\text{satLoc}_2)) \\ \text{r}(\text{satLoc}_2)\sin(\theta(\text{satLoc}_2)) \end{bmatrix} && (3.3) \\
\text{vec}_{\text{wall}_1} &= \frac{\text{corner} - \text{sat}_{\text{pt}_1}}{\|\text{corner} - \text{sat}_{\text{pt}_1}\|} && (3.4) \\
\text{vec}_{\text{wall}_2} &= \frac{\text{sat}_{\text{pt}_2} - \text{corner}}{\|\text{sat}_{\text{pt}_2} - \text{corner}\|} && (3.5)
\end{aligned}$$

To locate the robot center, the offsets normal to each wall are used. The fist step in calculating the offsets is to find the vectors normal to each wall as shown in equations 3.6 and 3.7. The offsets with respect to the corner in the sensor scan are found using equations 3.8 and 3.9.

$$\begin{aligned}
\mathrm{vec}_1 &= [0\ \ 0\ \ 1] \times \mathrm{vec}_{\mathrm{wall}_1} && (3.6)\\
\mathrm{vec}_2 &= [0\ \ 0\ \ 1] \times \mathrm{vec}_{\mathrm{wall}_2} && (3.7)\\
\mathrm{offset}_1 &= (\mathrm{corner} \cdot \mathrm{vec}_1)\mathrm{vec}_1 && (3.8)\\
\mathrm{offset}_2 &= (\mathrm{corner} \cdot \mathrm{vec}_2)\mathrm{vec}_2 && (3.9)
\end{aligned}$$

Once the offsets have been determined in the sensor scan, it is necessary to determine them in the environment. To do this, the coordinates must be transformed. First, matrices composed of the two wall vectors are defined for both the sensor scan and environment coordinate systems as shown in equations 3.10 and 3.11. There will be a unique matrix for each candidate corner in the environment. The transformation matrix necessary to place the robot in the environment coordinates is found using equation 3.12 and is given by equation 3.13. One can then find the offsets from the walls in the environment using equation 3.14. Finally, the Cartesian offsets are given by equations 3.15 and 3.16.

$$\begin{aligned}
\mathrm{A}_{\mathrm{scan}} &= \begin{bmatrix} \mathrm{vec}_{\mathrm{wall}_1} \\ \mathrm{vec}_{\mathrm{wall}_2} \end{bmatrix} && (3.10)\\
\mathrm{A}_{\mathrm{env}} &= \begin{bmatrix} \mathrm{vec}_{\mathrm{wall}_{1_{\mathrm{env}}}} \\ \mathrm{vec}_{\mathrm{wall}_{2_{\mathrm{env}}}} \end{bmatrix} && (3.11)\\
\mathrm{A}_{\mathrm{scan}}\mathrm{T} &= \mathrm{A}_{\mathrm{env}} && (3.12)\\
\mathrm{T} &= \mathrm{A}_{\mathrm{scan}}^{-1}\mathrm{A}_{\mathrm{env}} && (3.13)\\
\begin{bmatrix} \mathrm{offset}_{\mathrm{a}} \\ \mathrm{offset}_{\mathrm{b}} \end{bmatrix} &= \begin{bmatrix} \mathrm{offset}_1 \\ \mathrm{offset}_2 \end{bmatrix} T && (3.14)
\end{aligned}$$

$$\mathrm{dx_{corner}} = \mathrm{offset_a} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \mathrm{offset_b} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.15}$$

$$\mathrm{dy_{corner}} = \mathrm{offset_a} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \mathrm{offset_b} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.16}$$

Figure 3 shows the concave sensor scan for a robot in the top left corner of the T-Shaped environment shown earlier. A generalized concave corner with the saturation points, walls and offsets labeled is given by figure 4.

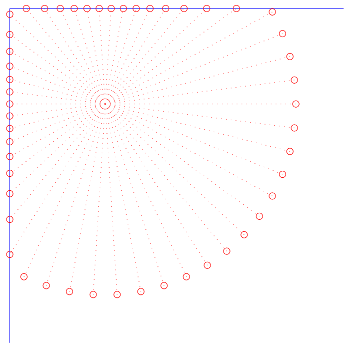

Fig. 3. Sensor Scan in Concave Corner

Figure 5 shows the convex sensor scan for a robot above the lower convex corner in the T-Shaped environment. Figure 6 shows the sensor scan for a generalized convex corner with the offsets, walls and saturation points labelled.

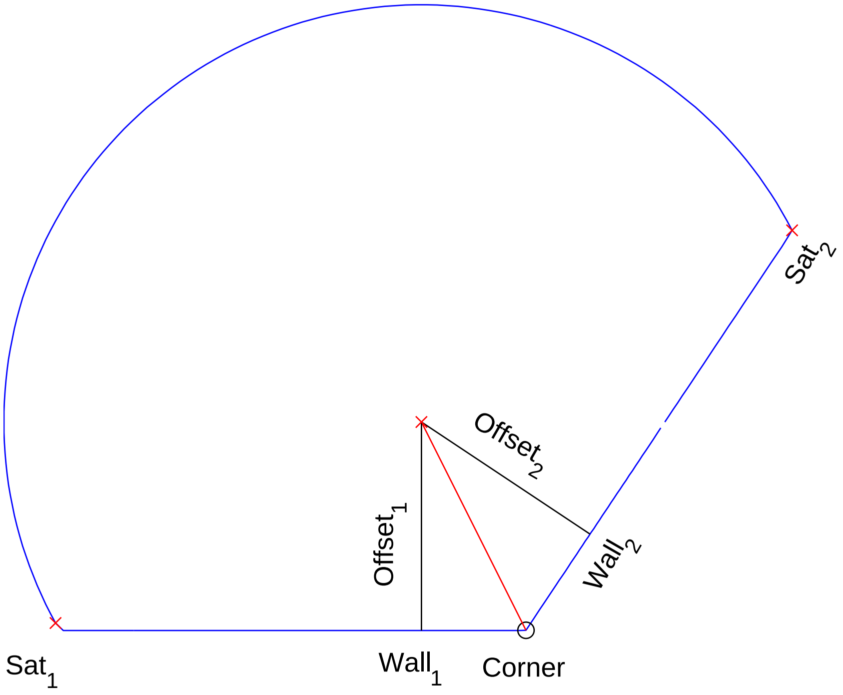

Fig. 4. Concave Sensor Scan with Offsets

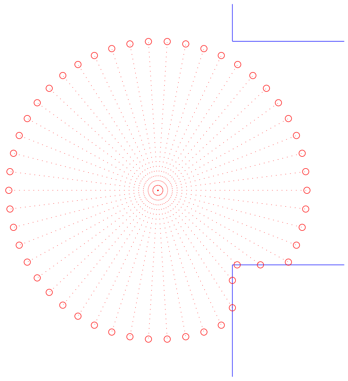

Fig. 5. Sensor Scan in Convex Corner with Two Detected Walls

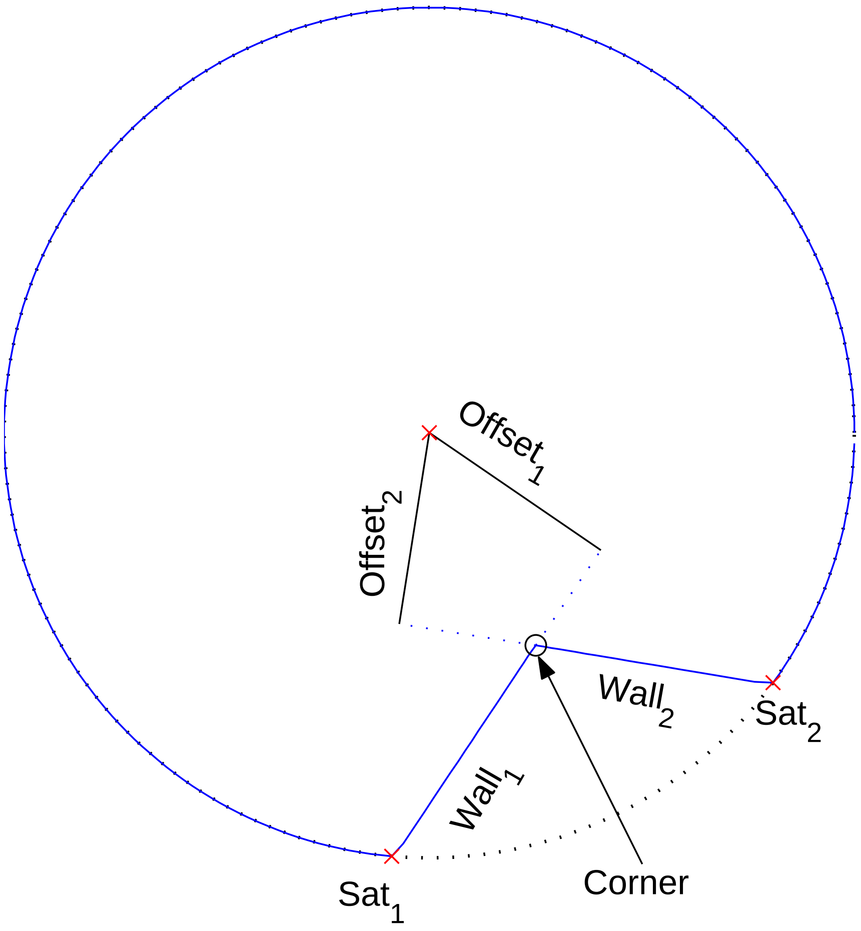

Fig. 6. Convex Sensor Scan with Offsets

2. Determining the Offset Distances for a Convex Corner with One Detected Wall

Assumptions

- A wall vector can be reliably determined for only one of the two walls forming the convex corner.
- From the sensor scan, the saturation points (the points at which the sensor response transitions to and from its maximum effective range) can be reliably determined.

The location of the convex corner is given by equation 3.17. The saturation points are defined using equations 3.2 and 3.3 as in the case where both walls are detected. Equation 3.18 defines the vector for the detected wall (in the case where the second saturation point lies on the detected wall). The normal vector is defined by equation 3.19. The offsets, with respect to the scan coordinates, are found via equations 3.20 and 3.21. Equations 3.22 and 3.23 are used to define the wall vector and normal vector for the offsets in the sensor scan and environment coordinates respectively.

$$\mathrm{corner} = \begin{bmatrix} \mathrm{r(convLoc)}\cos(\theta(\mathrm{convLoc}) \\ \mathrm{r(convLoc)}\sin(\theta(\mathrm{convLoc}) \end{bmatrix} \tag{3.17}$$

$$\mathrm{vec_{wall}} = \frac{\mathrm{sat_{pt_2}} - \mathrm{corner}}{||\mathrm{sat_{pt_2}} - \mathrm{corner}||} \tag{3.18}$$

$$\mathrm{vec_{n}orm} = [0\ \ 0\ \ 1] \ \times \ \mathrm{vec_{wall}} \tag{3.19}$$

$$\mathrm{offset_1} = (\mathrm{corner} \cdot \mathrm{vec_{n}orm})\mathrm{vec_{n}orm} \tag{3.20}$$

$$\mathrm{offset_2} = (\mathrm{corner} \cdot \mathrm{vec_{wall}})\mathrm{vec_{wall}} \tag{3.21}$$

$$\mathrm{A_{scan}} = \begin{bmatrix} \mathrm{vec_{wall}} \\ \mathrm{vec_{norm}} \end{bmatrix} \tag{3.22}$$

$$\mathrm{A_{env}} \quad = \quad \begin{bmatrix} \mathrm{vec_{wall_{env}}} \\ \mathrm{vec_{norm_{env}}} \end{bmatrix} \tag{3.23}$$

Figure 7 shows the convex sensor scan for a robot in the T-Shaped environment near the lower convex corner. As shown in the figure, the robot can not detect both walls that form the convex corner. A generalized convex corner with one detected wall is shown in Figure 8. This figure corresponds to equation 3.18; if the first saturation point lay on the wall, instead of the second, the equation would have to be modified slightly as shown in equation 3.24.

$$\mathrm{vec_{wall}} = \frac{\mathrm{corner} - \mathrm{sat_{pt_1}}}{\|\mathrm{corner} - \mathrm{sat_{pt_1}}\|} \tag{3.24}$$

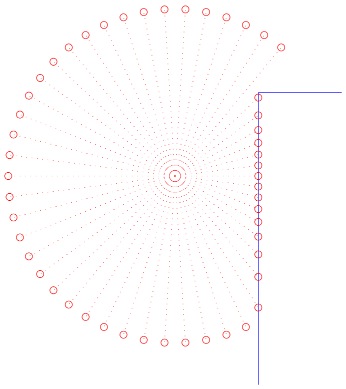

Fig. 7. Sensor Scan in Convex Corner

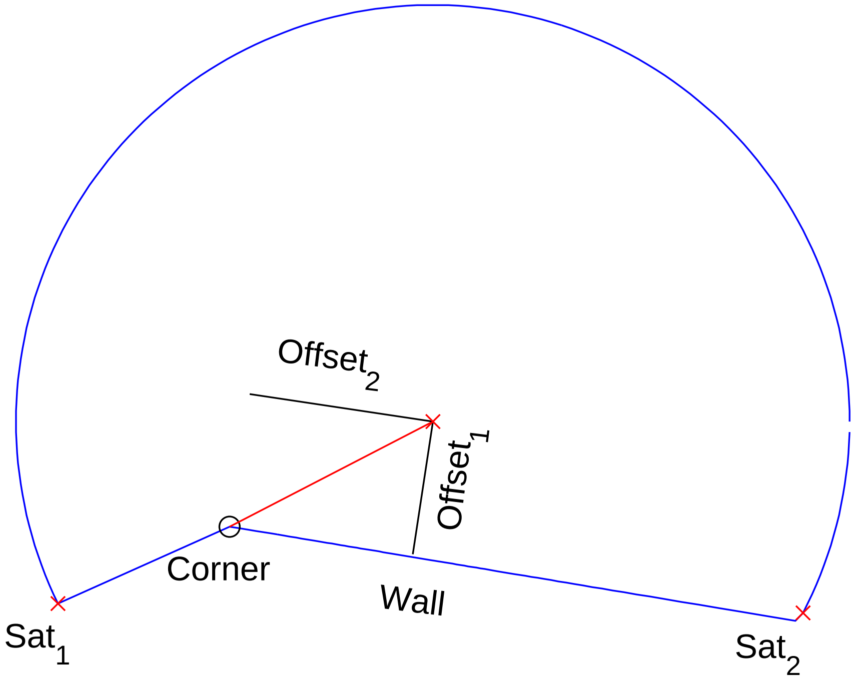

Fig. 8. Convex Sensor Scan with Offsets

B. Localization with One Corner and One Wall

If only one of the two robots detects a corner and the second robot detects a wall, localization can still be effectively performed. The relative heading angle, ϕ, is defined as it was in the case where both robots detect a corner. For the second robot, which only detects an edge in its sensor scan, γ is defined as the counter-clockwise angle from the robot heading to the point on the edge closest to the robot.

For the first robot, a list of possible locations can be generated from the sensor scan as was discussed previously. In the case of the second robot, where only a normal offset from a wall is known, the possible locations are given as a range of points offset from the walls of the environment. Once the possible locations for each robot have been determined independently, it is necessary to use the known relative distance and heading information to locate the robots.

This is done by determining the possible location for the second robot corresponding to each potential location for the first robot. Given a possible location for the first robot, its absolute heading can be determined. First, it is necessary to determine the vector from the robot location to the corner location in the environment coordinate system as shown in equation 3.25. From this vector the angle from the robot center to the corner can be determined using equation 3.26 and the absolute heading for the robot can be calculated via equation 3.27.

$$\begin{aligned}
\mathrm{vec_{corner}} &= [\mathrm{corner_x}\ \mathrm{corner_y}] - [\mathrm{robot_{i,x}}\ \mathrm{robot_{i,y}}] && (3.25)\\
\theta_{\mathrm{corner}} &= \tan^{-1}\left(\frac{\mathrm{vec_{corner_y}}}{\mathrm{vec_{corner_x}}}\right) && (3.26)\\
\mathrm{heading_{abs}} &= \theta_{\mathrm{corner}} - \lambda_{\mathrm{robot}} && (3.27)
\end{aligned}$$

Once the absolute heading for the first robot (in that potential location) is known, it is possible to determine the absolute value of ϕ. Using this value of ϕ in conjunction with the known distance between the two robots, possible center points for the second robot can be generated using equations 3.28-3.30. Each pair of robots is checked for a line of sight; if a line of sight exists, the pair of locations is retained.

$$\phi_{\mathrm{abs}} = \phi + \mathrm{heading} \tag{3.28}$$

$$\mathrm{d}_{\mathrm{x}_{\mathrm{robots}}} = \mathrm{d}\cos(\phi_{\mathrm{abs}}) \tag{3.29}$$

$$\mathrm{d}_{\mathrm{y}_{\mathrm{robots}}} = \mathrm{d}\sin(\phi_{\mathrm{abs)}} \tag{3.30}$$

1. Determining the Cartesian Offsets for a Detected Wall

Assumptions

- The robot can reliably detect a wall.
- From the sensor scan, the saturation points (the points at which the sensor response transitions to and from its maximum effective range) can be reliably determined.

From the sensor scan for the robot that only detects a wall, it is only possible to determine regions of points in which the robot may be located. The Cartesian offset for this range are found similarly to those found for the corners. Equations 3.2 and 3.3 are used to define the saturation points as before. The wall vector and the vector normal to the wall are defined using equations 3.31 and 3.32 respectively. The Cartesian offsets are determined from the robot center to the point on the wall closest to the robot.

$$\begin{aligned} \mathrm{vec}_{\mathrm{wall}} &= \frac{\mathrm{sat}_{\mathrm{pt}_2} - \mathrm{sat}_{\mathrm{pt}_1}}{\|\mathrm{sat}_{\mathrm{pt}_2} - \mathrm{sat}_{\mathrm{pt}_1}\|} \qquad (3.31) \\ \mathrm{vec}_{\mathrm{norm}} &= [0\ \ 0\ \ 1] \times \mathrm{vec}_{\mathrm{wall}} \qquad (3.32) \end{aligned}$$

The offset from the wall in the sensor scan coordinates is found using equation 3.33. The wall vector and normal vector for the sensor scan and the environment are given by equations 3.34 and 3.35. The calculation of the Cartesian offset is concluded as in the case where a corner and two walls are detected; this is shown in equations 3.12-3.16. To determine the regions of potential locations in the environment, the detected wall segment length is compared to the environment wall lengths. Using the known wall length, length of the wall segment in the sensor scan and maximum effective sensor range in conjunction with the normal offset, the region of points parallel to each wall that represents the range of possible locations for the robot can be found.

$$\text{offset} = (\text{sat}_{\text{pt}_1} \cdot \text{vec}_{\text{norm}})\text{vec}_{\text{norm}} \tag{3.33}$$

$$\text{A}_{\text{scan}} = \begin{bmatrix} \text{vec}_{\text{wall}} \\ \text{vec}_{\text{norm}} \end{bmatrix} \tag{3.34}$$

$$\text{A}_{\text{env}} = \begin{bmatrix} \text{vec}_{\text{wall}_{\text{env}}} \\ \text{vec}_{\text{norm}_{\text{env}}} \end{bmatrix} \tag{3.35}$$

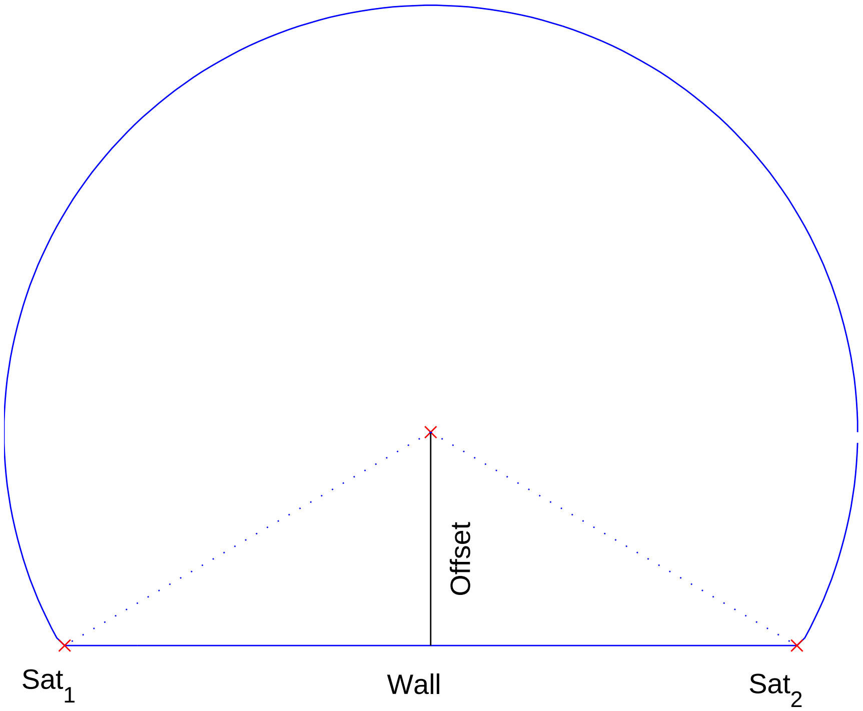

Fig. 9. Environmental Edge Sensor Scan with Offset

C. Required Information for Localization

The virtual link length and relative heading information is required to localize the system of robots relative to each other. This information can be found via a rotating CMOS camera as described in appendix A.

Absolute localization of the robot system is more difficult. If one robot in the system detects an environmental corner, yielding a set of potential absolute configurations for that robot, the entire system may potentially be localized as all necessary information is available.

However, locating one corner is not sufficient to guarantee a unique solution (due to environmental symmetry, multiple solutions may be possible). It is not possible to accurately quantify the information required to guarantee a unique solution to the localization method. Environmental complexity and the number of robots in the environment as well as the configuration of each robot in the system dictate the amount of information required to localize the system.

D. Over-Determined System

There are two ways in which the system may be over-determined. The first being if more environmental features are detected than are required to localize the system. An instance in which there is a surplus of virtual link information (link length and relative heading), is the second.

For the first case, the extra information would serve to reduce the potential locations for each robot thereby decreasing the size of the set to be searched for a solution. In the second case, the extra information can be used to reduce experimental error. The extra virtual link information may also allow for further reduction of the solution set from the localization algorithm.

CHAPTER IV

EXTENSION OF LOCALIZATION METHOD TO MULTIPLE PAIRS OF ROBOTS

Thus far, the localization method has been discussed primarily for a pair of robots. The method can easily be extended to accommodate three or more robots. This is done by localizing successive pairs of robots. Once each pair of robots has been localized, the intersection of neighboring pairs is found.

A. Determining Which Links to Select When Multiple Options are Available

In the case when there are multiple link configurations available for localization, this occurs when the system is over-determined as described at the end of the previous chapter, there are two ways to select the link configuration. One could select each possible combination of virtual links and find a solution for each system; the localization results for each virtual link combination could then be averaged. This would serve to reduce any experimental error from the sensor scan and potentially cull the solution set further.

However, localizing each possible system would be very computationally expensive and often unnecessary. Selecting the configuration with the shortest maximum link length would increase efficiency and minimize error due to the camera as the error in the camera increases rapidly as the link length increases.

B. Localization of n Robot System

In order to localize a system of more than two robots, it is necessary that each robot in the system be a unique color (or have some other unique characteristic identifiable with a camera) that contrasts with the walls of the environment. Using the rotating sensor scan information, the potential locations for each individual robot can be determined. The potential virtual links for each robot are then sorted in ascending length order and stored in a list, L_i.

The flowchart shown in figure 10 describes an efficient localization method for a multiple robot system containing n robots. Initially, the localization flag for each robot is set to false. For each pair of robots that can be uniquely localized, the flags are set to true. If no potential virtual link yields a unique localization solution for a robot, another robot is added to the set and the localization for that set of robots is found as described in the following section. If the three robot set can not be localized, an additional robot is added to the localization set. This is repeated until a unique solution is found or the localization fails with an n robot set.

C. Intersecting Solution of Robot Pairs to Localize Set of n Robots

The method used to find the possible locations for each pair of robots in an n robot system is shown in the following pseudocode:

```
For i = 1 to n-1
    Find possible locations for robots i and i + 1
    Store possible locations in possLocsABi in the form [($X_i$, $Y_i$) ($X_{i+1}$, $Y_{i+1}$)]
End
```

Once the possible locations for each pair of robots has been determined, it is necessary to intersect the solution sets to determine the possible location set for all robots in the set. The possible locations of the robot set can be written as [(X_1, Y_2) (X_2, Y_2) (X_3, Y_3) ... (X_n, Y_n)]. Where X and Y are the sets of the possible x and y coordinates for each robot.

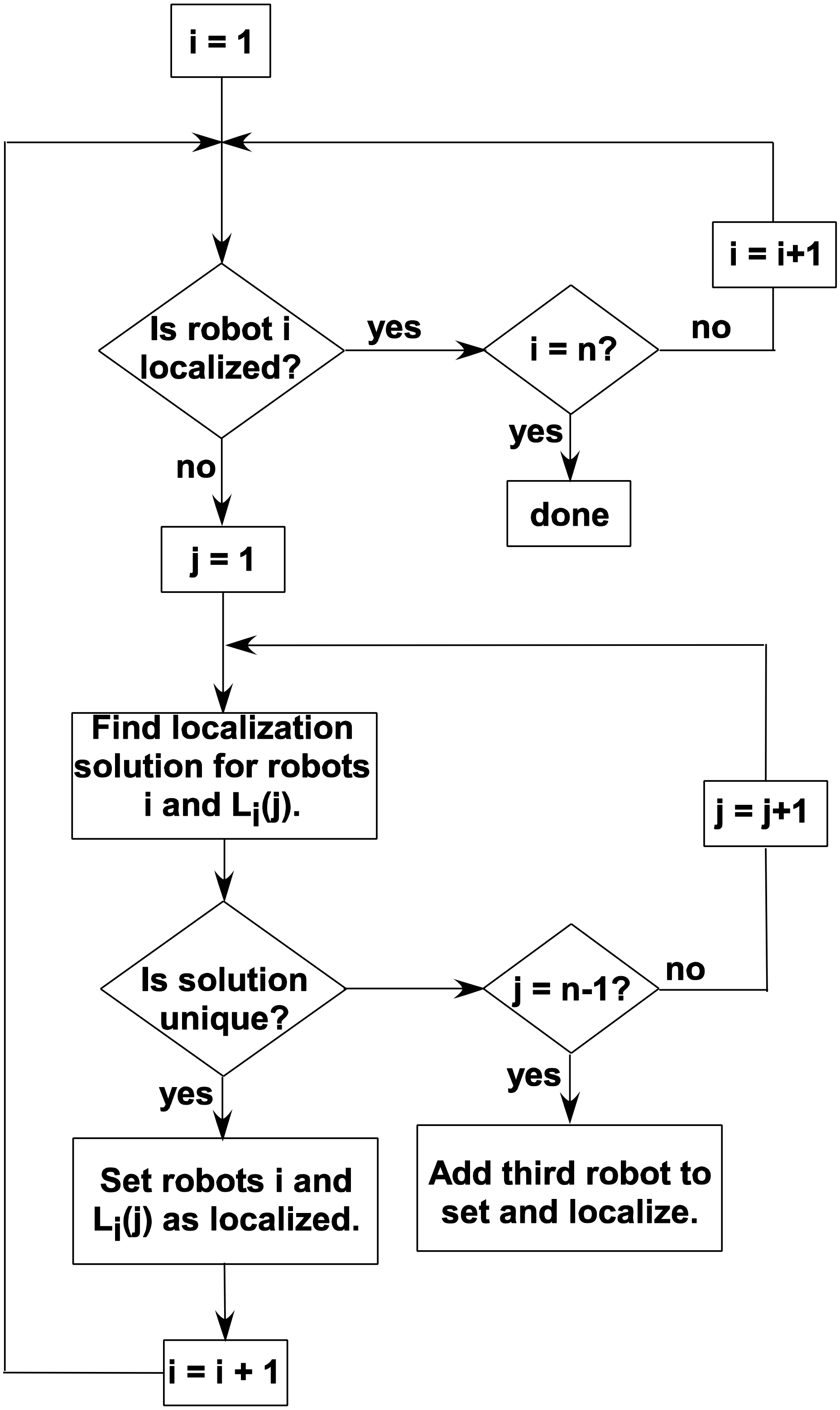
i = 1
Is robot i localized?
yes
i = n?
no
i = i+1
yes
done
no
j = 1
Find localization solution for robots i and $L_i(j)$.
Is solution unique?
j = n-1?
no
j = j+1
yes
Add third robot to set and localize.
yes
Set robots i and $L_i(j)$ as localized.
i = i + 1

Fig. 10. Localization Algorithm Flowchart

The set of possible locations stored in the array possLocsAB can be written as [possLocsAB(1) possLocsAB(2) possLocsAB(3) ... possLocsAB(n-1)] where the third and fourth column of possLocsAB(i) must coincide with the first and second column of possLocsAB(i+1) in order for the solution to be a possibility.

For example, let possLocsAB be given by table I. If the possible locations for robot 2 are intersected, the resulting possible locations for robots 1, 2 and 3 are [(1,1) (4,4) (3,11)] and [(1,12) (4,7) (1,11)]. By finding the intersection of the possible locations for robot 3 from the sets robot-robot2-robot3 and robot3-robot4, the solution for the system of robots can be found as [(1,1) (4,4) (3,11) (1,11)].

Table I. Possible Locations for 4 Robot System

Pair 1		Pair 2		Pair 3	
possLocsAB(1)		possLocsAB(2)		possLocsAB(3)	
Robot A_1	Robot B_1	Robot A_2	Robot B_2	Robot A_3	Robot B_3
(Robot 1)	(Robot 2)	(Robot 2)	(Robot 3)	(Robot 3)	(Robot 4)
(1,1)	(4,4)	(4,4)	(3,11)	(3,2)	(1,8)
(1,12)	(4,7)	(4,7)	(1,11)	(9,7)	(3,10)
(4,1)	(1,4)	(7,8)	(4,13)	(3,11)	(1,11)
(4,12)	(1,7)				

CHAPTER V

SIMULATION EXAMPLES

A. Two Robots

For the first simulation example, robot 1 is located at (1.2, 1.2) with a heading of 30° and robot 2 is located at (0.3, 2.5) with a heading of 75°. Figures 11 and 12 show simulated sensor scans for the two robots in a T-Shaped Environment.

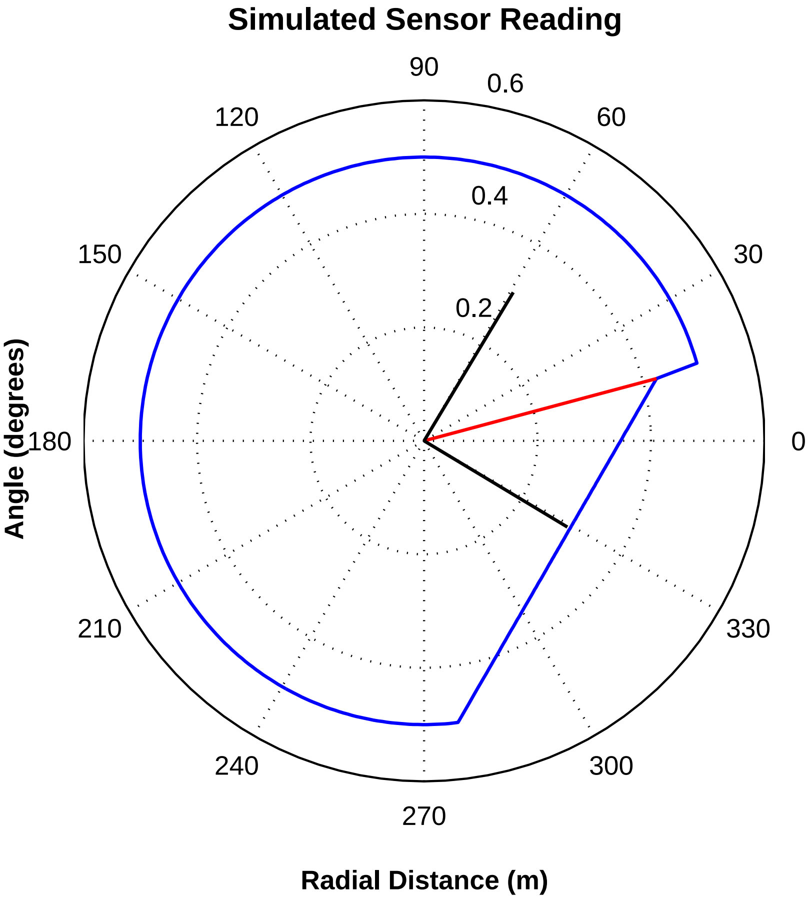

Fig. 11. Convex Corner Sensor Scan with Offsets for Robot 1

Using the sensor scans, the potential locations of the robots are found as shown

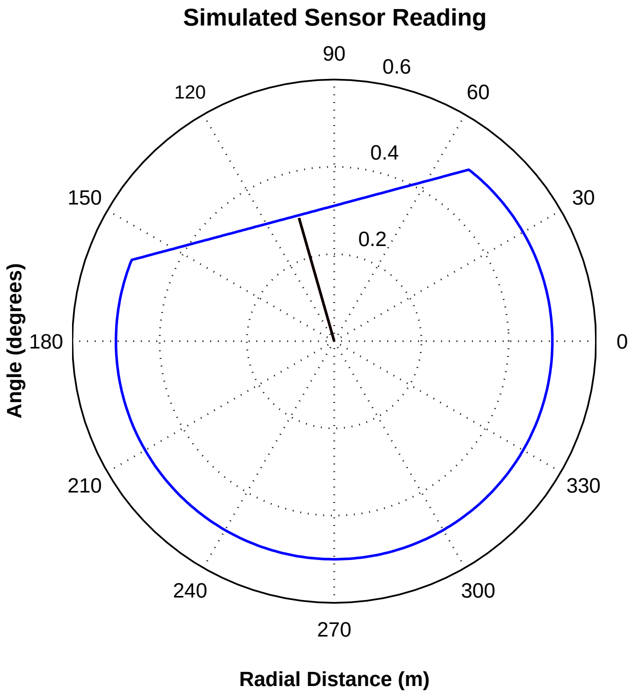

Fig. 12. Edge Sensor Scan with Offset for Robot 2

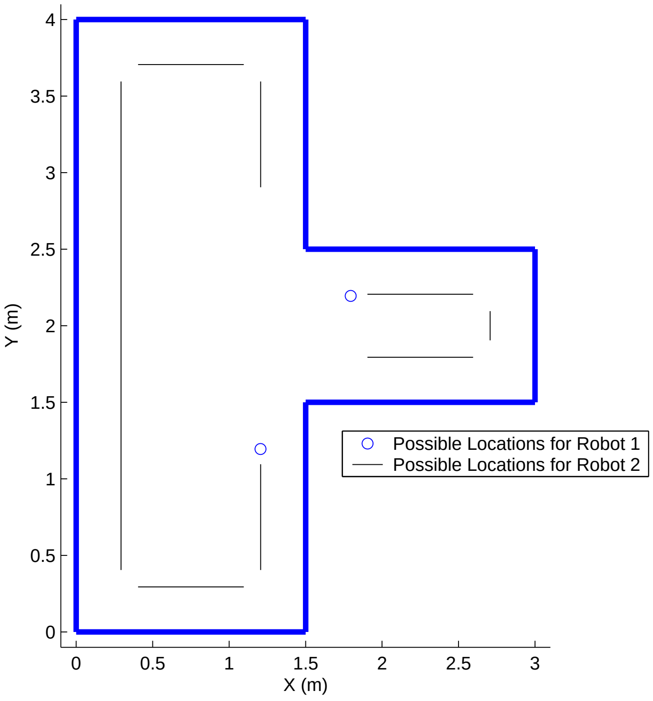

Fig. 13. Possible Locations for Robots 1 and 2

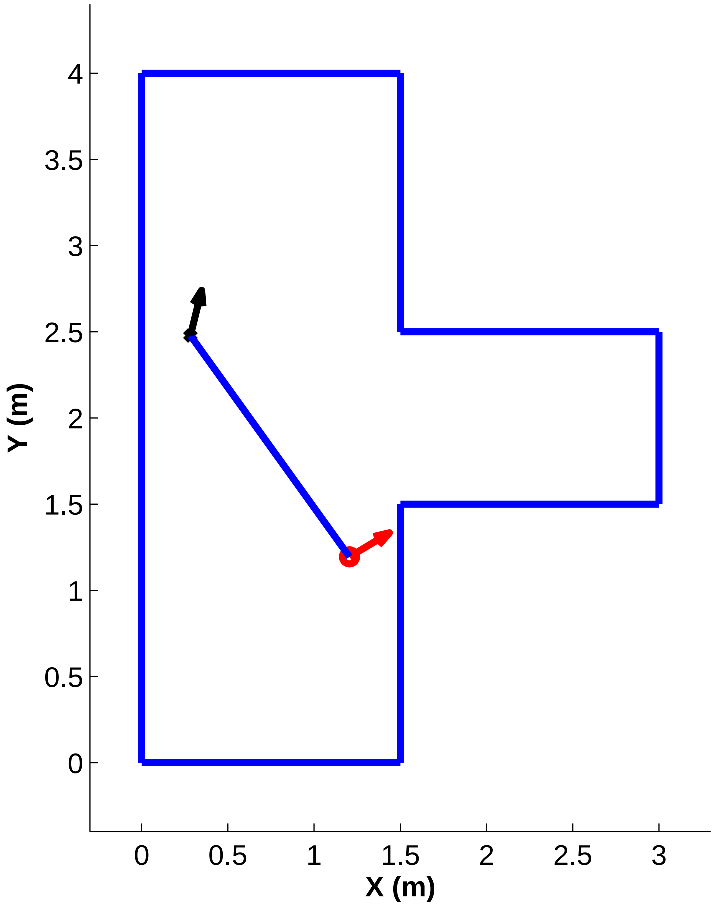

Fig. 14. Localization Result for Two Robots in T-Shaped Environment

in Figure 13. The localization results are given in Figure 14 and show that the correct solution has been found.

B. Three Robots

1. Two Robots with Multiple Solutions

In the second simulation example, robot 1 is located at (1.3, 0.2) with a heading of 157.5° and robot 2 is located at (0.15, 3.25) with a heading of 45°. Figures 15 and 16 show the sensor scans for the two robots with the determined offsets.

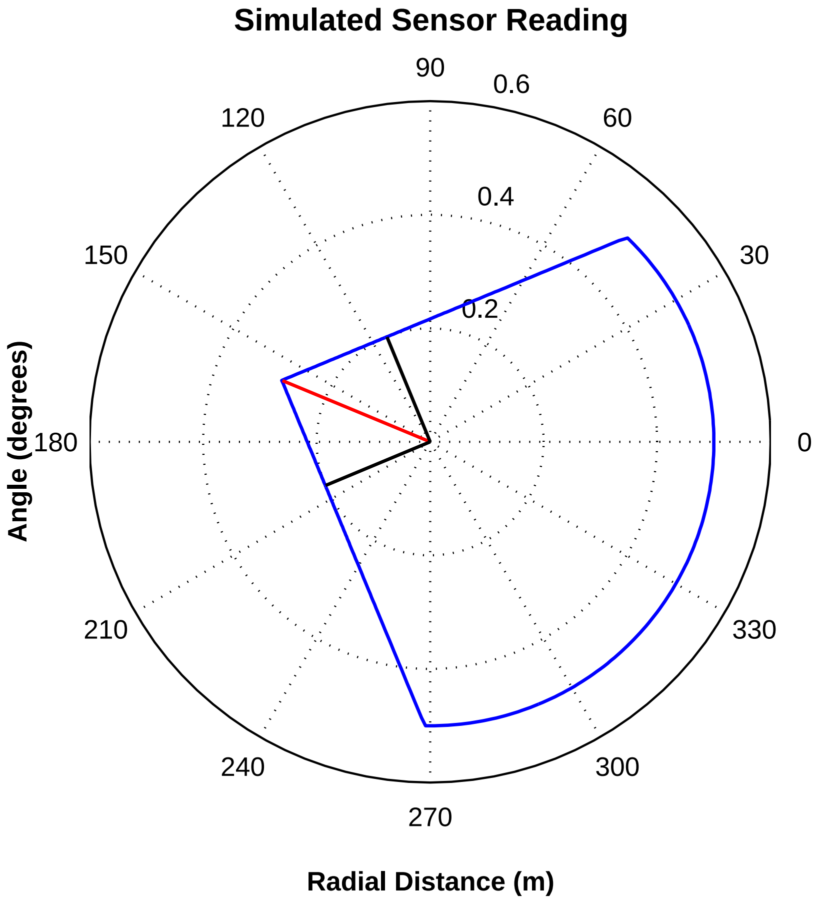

Fig. 15. Concave Corner Sensor Scan with Offsets for Robot 1

Using the information from the sensor scans, potential locations for the center of each robot are determined as shown in figure 17. As shown in figure 18, the

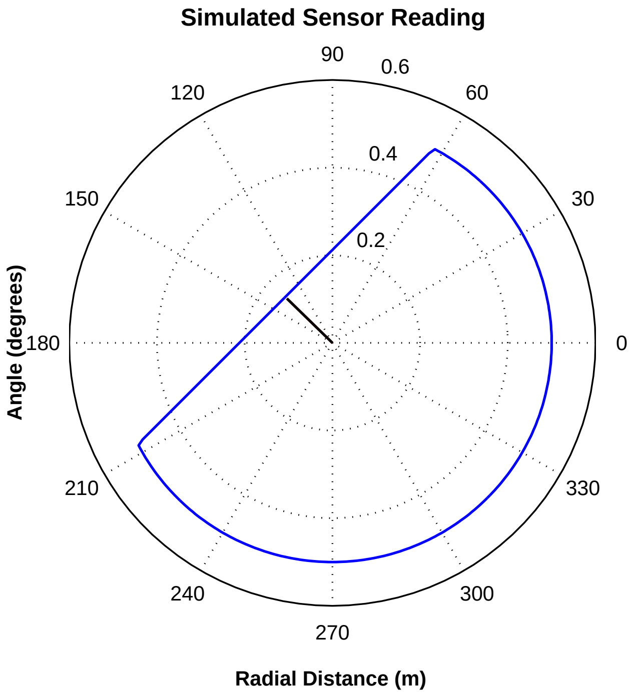

Fig. 16. Edge Sensor Scan with Offset for Robot 2

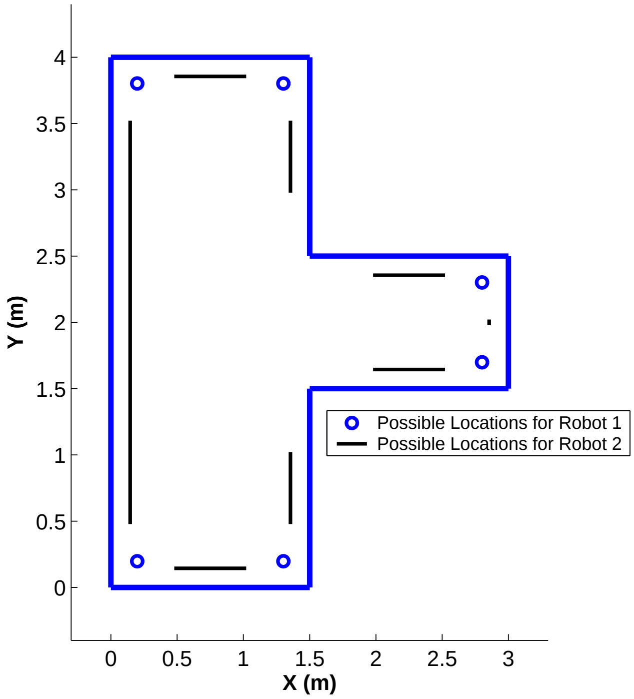

Fig. 17. Possible Locations for Robots 1 and 2

localization method returns two possible sets of locations due to symmetry in the environment. In this scenario, a third robot could be used to uniquely localize the robot system.

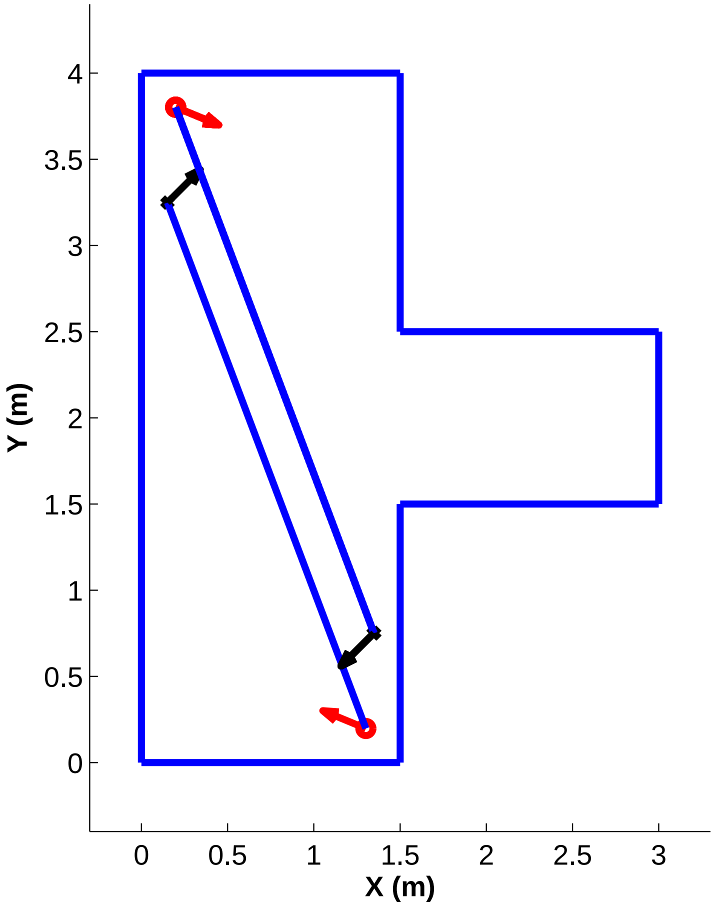

Fig. 18. Multiple Localization Results for Two Robots in T-Shaped Environment

2. Addition of Third Robot to Find Unique Solution

If a third robot is added at (1.4, 2.4) with a heading of 90°, a unique location can be determined for each robot. Figure 19 shows the sensor scan for the third robot with the calculated offsets. The possible locations for the second pair of robots, robots 2 and 3, are show in figure 20.

The localization result for the second pair of robots is shown in figure 21. By finding the location of the second robot in the localization result for the first pair of robots (shown in figure 18) corresponding to the location of the first robot in the result for the second pair of robots, a unique solution for the system of three robots is found as shown in figure 22.

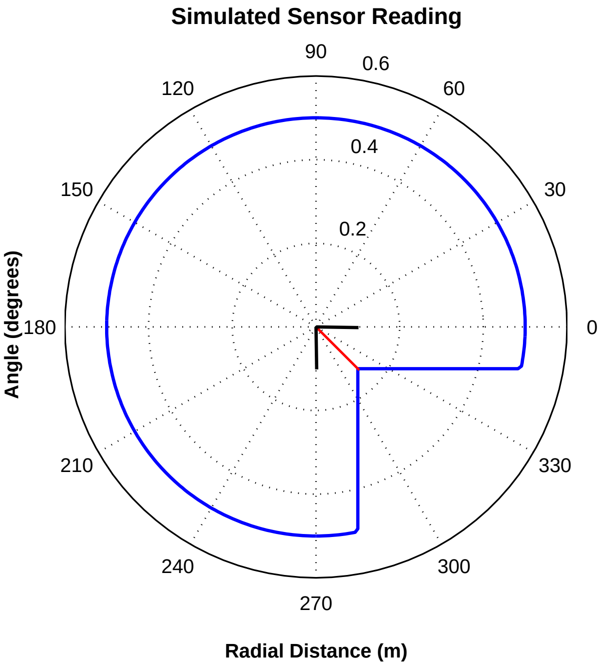

Fig. 19. Convex Corner Sensor Scan with Offsets for Robot 3

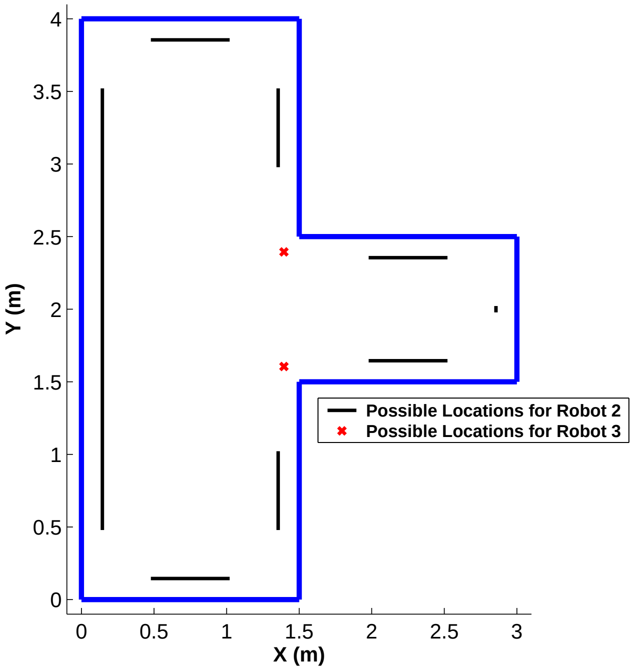

Fig. 20. Possible Locations for Robots 2 and 3

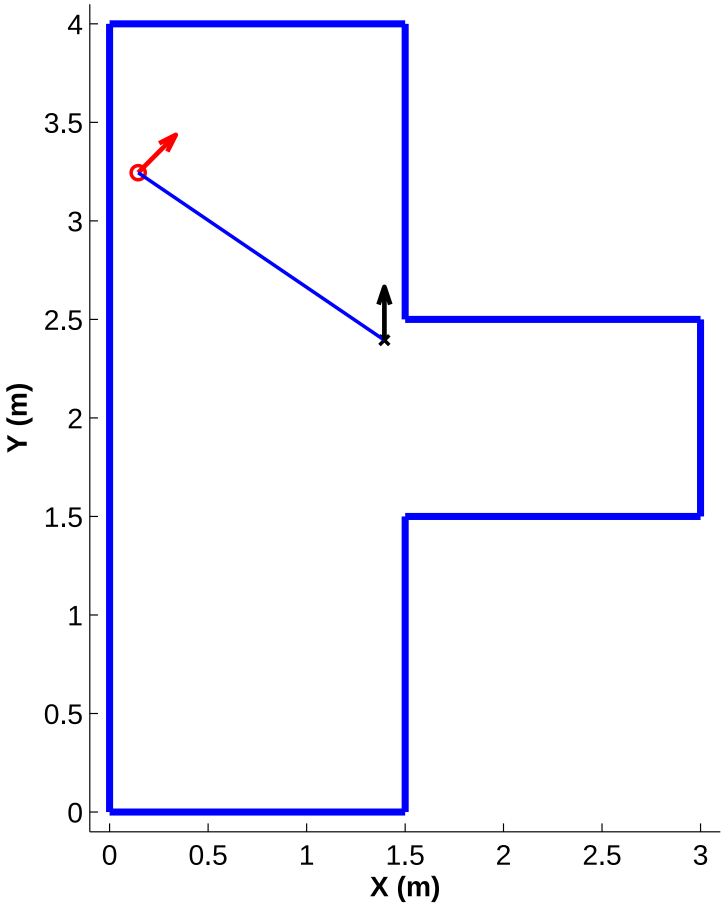

Fig. 21. Localization Results for Second Pair of Robots

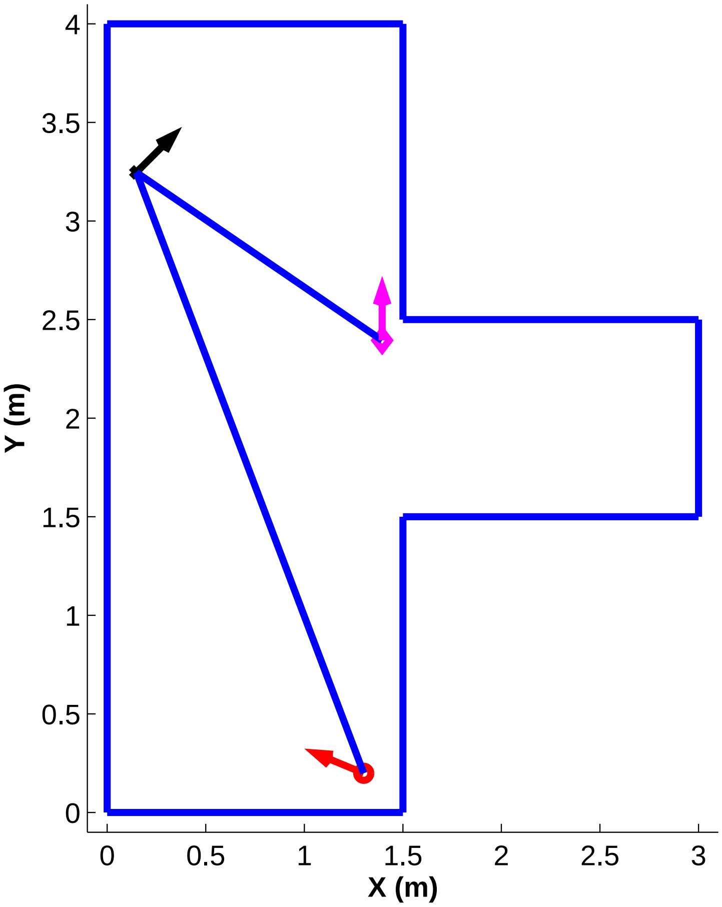

Fig. 22. Localization Results for Three Robot System

CHAPTER VI

EXPERIMENTAL VERIFICATION

The proposed localization method has proven effective in simulations. The next step is to test it experimentally. The virtual link length and relative heading information can be determined using a rotating CMOS camera. The method used to be to find the corners, edges and saturation points in the simulated sensor scans is no longer valid. The approach used for the experimental data will be presented in the ensuing sections.

Although a rotating CMOS camera can be effectively used to determine the virtual link length and relative heading information as shown in appendix A, the effective range of the available camera (CMUCam) is insufficient.

The CMUCam is an inexpensive vision system for mobile robots. It includes an Omnivision OV6620 CMOS camera on a chip and a Ubicom microcontroller running at 75 MHz for on-board image processing. The image size acquired by the vision system is 80 x 143 pixels. Using this vision system, it is only possible to accurately measure the link length for distances up to 1.1 m.

As a result of the vision system's limitations, the virtual link length and relative heading information will be calculated from the known locations of the robots so the virtual link information can be provided to the localizer.

A rotating Sharp GP2D12 Infrared range sensor is used to detect the walls in the environment. The sensor contains an IR transmitter and receiver and has a nominal effective range from 10-80 cm.

A. Calibration

The first step in the experimental verification of the localization method is the calibration of the infrared sensor. The infrared sensor was affixed to a plexiglass base which was mounted to a servomotor.

The nominal effective range of the sensor is from 10 cm to 80 cm. The sensor was calibrated from 9 cm to 55 cm. The distance to the wall was measured from the plexiglass base which was about 1 cm closer to the wall than the sensor. For each distance, the sensor was placed normal to the wall and 2000 values were read. Figures 23 and 24 show the normalized histogram and normal probability density functions for the calibration data. The plots indicate that although the data is not precisely Gaussian, the distribution can be approximated as such.

The mean and variance for the 2000 sensor readings taken at each distance are found. The calibration curve is determined using the MATLAB function *pchip* which finds the Piecewise Cubic Hermite Interpolating Polynomial for each interval. Figure 25 shows the calibration curve for the sensor.

An issue of concern is the accuracy of the sensor towards the end of the calibrated range (0.5 m). One way to determine the effective operating range of the sensor is to determine the confidence interval for each point on the calibration curve. This was done by calculated the 95 percent confidence interval for each of the data points using equation 6.1.

$$CI_{95} = \bar{x} - \frac{1.96\sigma}{\sqrt{n}} \leq \mu \leq \bar{x} + \frac{1.96\sigma}{\sqrt{n}} \tag{6.1}$$

Figure 26 shows the calibration curve along with the 95 % confidence interval. From this curve, it can be seen that as the nominal sensor reading decreases (i.e. the sensor is further from the wall) the 95 % confidence interval for the distance from the

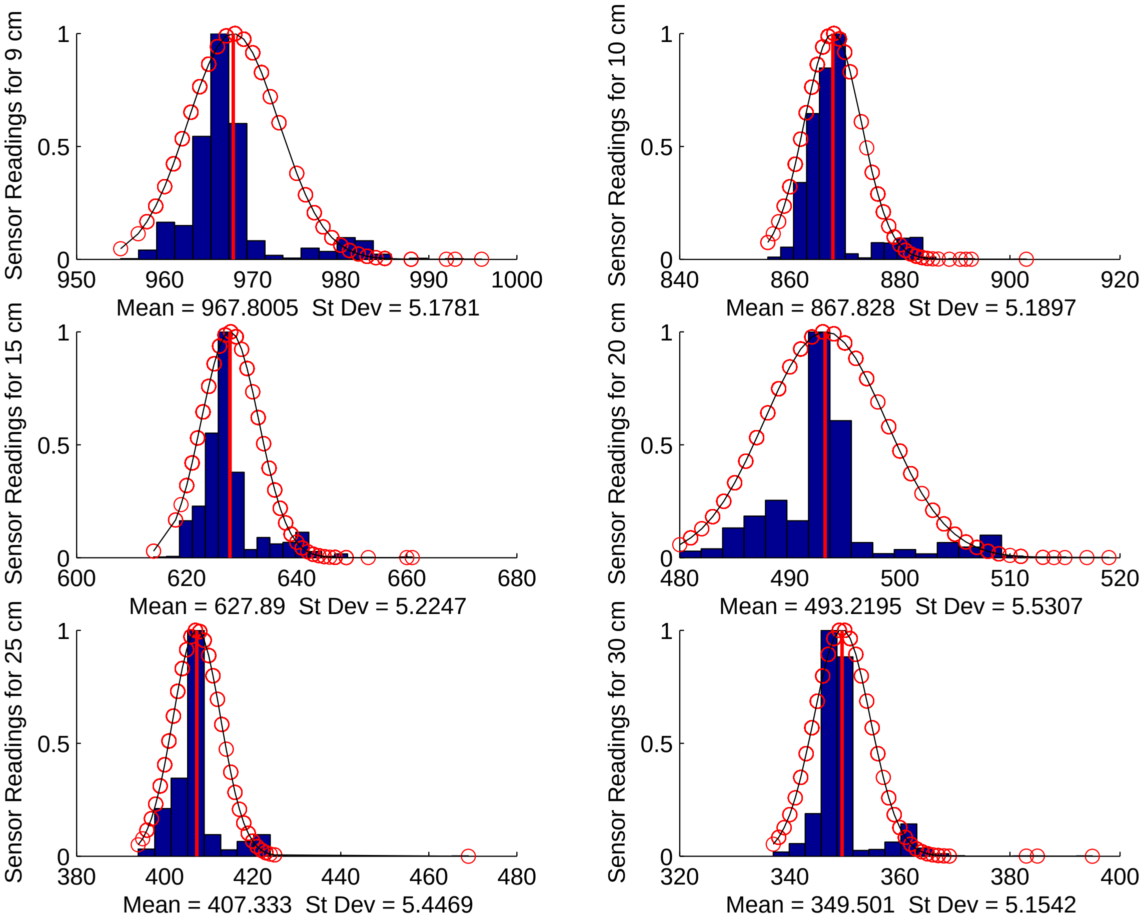

Fig. 23. Normalized Histogram and Normal Probability Density Function for Calibration Data from 9 cm - 30 cm

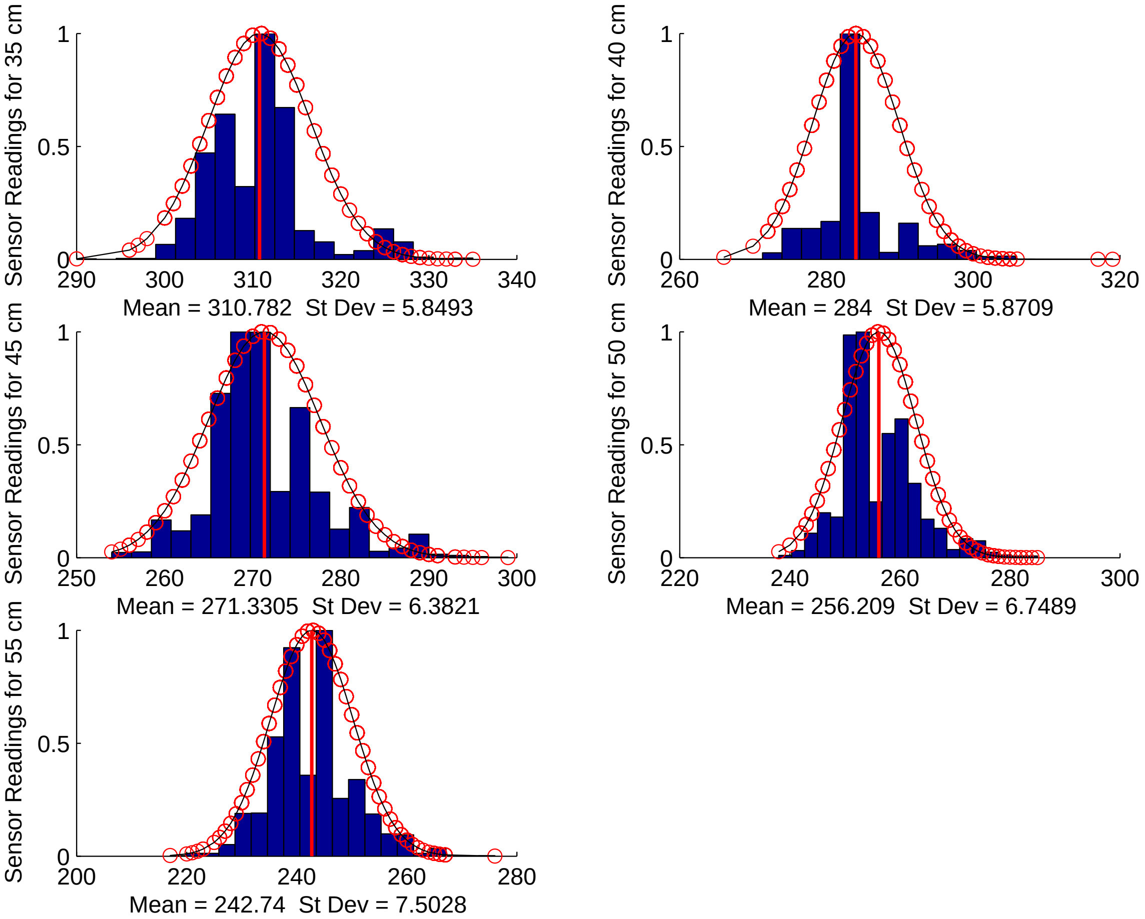

Fig. 24. Normalized Histogram and Normal Probability Density Function for Calibration Data from 35 cm - 50 cm

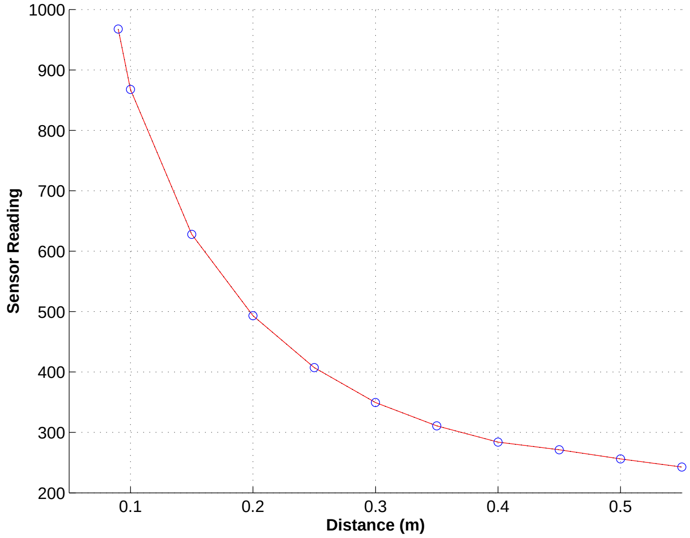

Fig. 25. Calibration Curve for Infrared Sensor

Table II. Nominal Distance and Confidence Interval for Various Nominal Sensor Reading Values

Nominal Sensor Reading Value	**Nominal Distance** (cm)	**Lower Limit** (cm)	**Upper Limit** (cm)	**Range** (cm)
800	11.41	11.20	11.62	0.42
450	22.52	21.89	23.14	1.25
350	30.00	29.08	31.29	2.21
275	39.82	37.67	44.50	6.83

wall increases significantly. Table II shows how the confidence interval for the distance corresponding to the sensor reading widens as the sensor reading value decreases. As the confidence interval gets larger, the sensor readings become too inaccurate to be used in the localization method. Therefore, all experimental data was truncated at 30 cm. While this limits the effective range of the infrared sensor, it greatly increases the likelihood of usable experimental data.

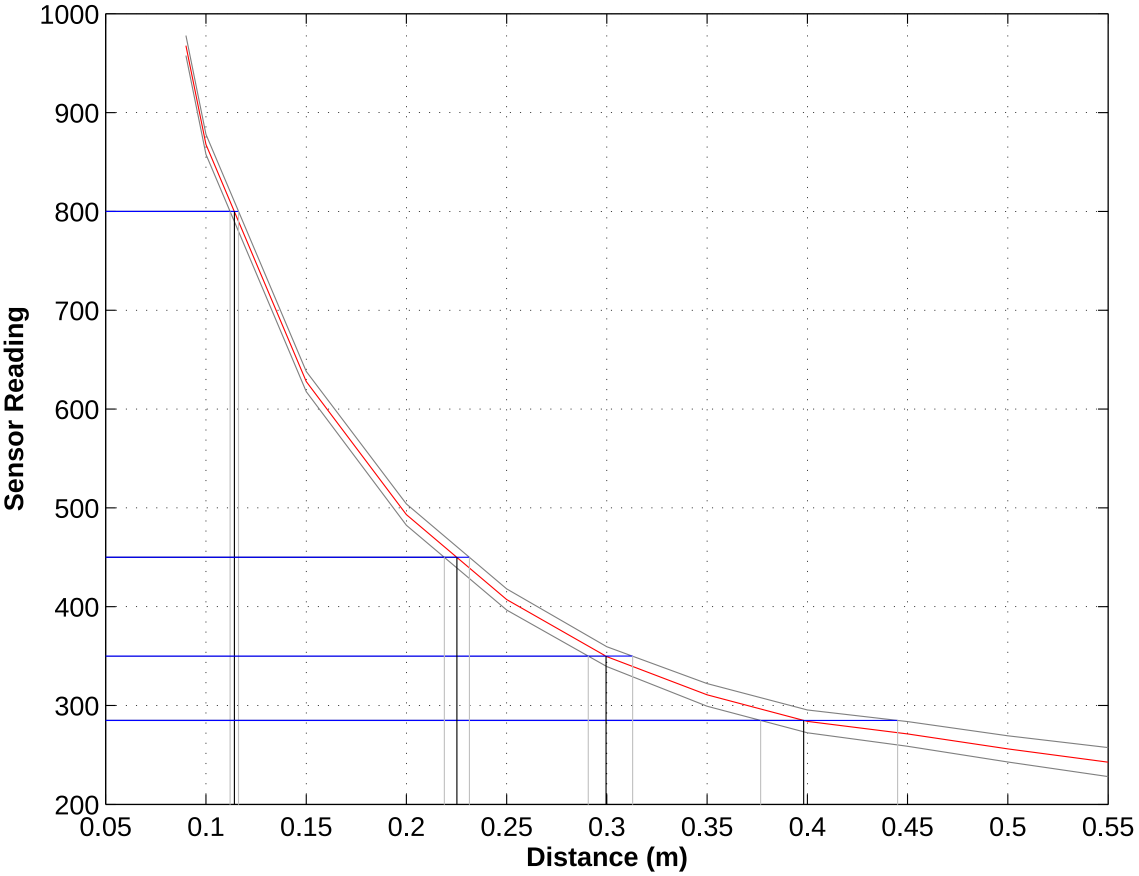

Fig. 26. Calibration Curve for Infrared Sensor with 95 % Confidence Interval

B. Determining the Cartesian Offsets for Experimental Data

The method used to determine the offsets from the experimental data is significantly different from that used for the simulation data. In the experimental data, determining the corner point is much more difficult than it is for the simulation data. To find the corner point in the experimental data, it is necessary to calculate a least squares linear fit for each wall. Once an equation for each wall is found, the corner point is best approximated by the intersection point of the two lines.

In order to determine an equation for each wall, it is necessary to filter the data. Figures 27 and 29 show raw experimental sensor scans for convex and concave corners. The data is filtered by replacing the value for each point with the average of itself and a specified number of points before and after the point. Figures 28 and 30 show the corresponding filtered sensor scans. In this case, the five points before and after each data point were averaged with the data point value to determine the new value.

Once the sensors scan is filtered, an equation can be found for each detected wall in the y-intercept form as shown in equation 6.2. The least squares equation was determined using the *polyfit* function in MATLAB. To find the intersection point of the two lines, equation 6.3 is used.

$$\mathrm{y_{wall}} = \mathrm{m_{wall}x_{wall}} + \mathrm{b_{wall}} \tag{6.2}$$

$$\begin{bmatrix} \mathrm{corner_y} \\ \mathrm{corner_x} \end{bmatrix} = \begin{bmatrix} \frac{\mathrm{m_{wall_2}b_{wall_1} - m_{wall_1}b_{wall_2}}}{\mathrm{m_{wall_2} - m_{wall_1}}} \\ \frac{\mathrm{corner_y - b_{wall_1}}}{\mathrm{m_{wall_1}}} \end{bmatrix} \tag{6.3}$$

$$\theta_1 = \tan^{-1}(\mathrm{m_{wall_1}}) \tag{6.4}$$

$$\theta_2 = \tan^{-1}(\mathrm{m_{wall_2}}) + \pi \tag{6.5}$$

$$\mathrm{vec_{wall_1}} = \frac{[\cos(\theta_1) \quad \sin(\theta_1)]}{\|\cos(\theta_1) \quad \sin(\theta_1)\|} \tag{6.6}$$

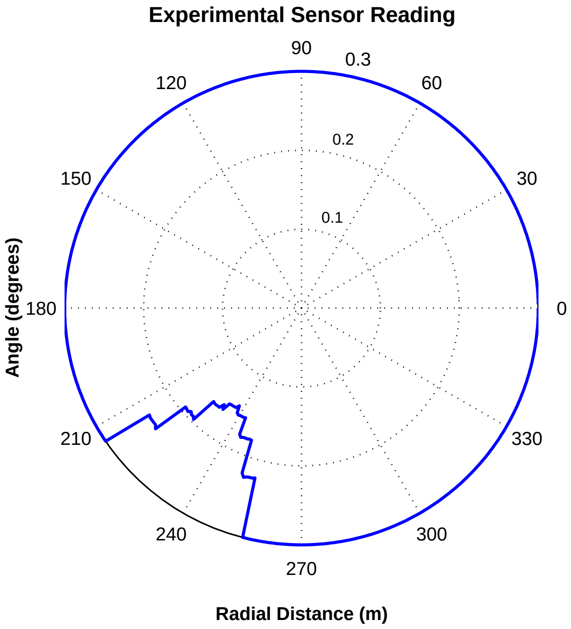

Fig. 27. Raw Experimental Sensor Scan of Convex Corner

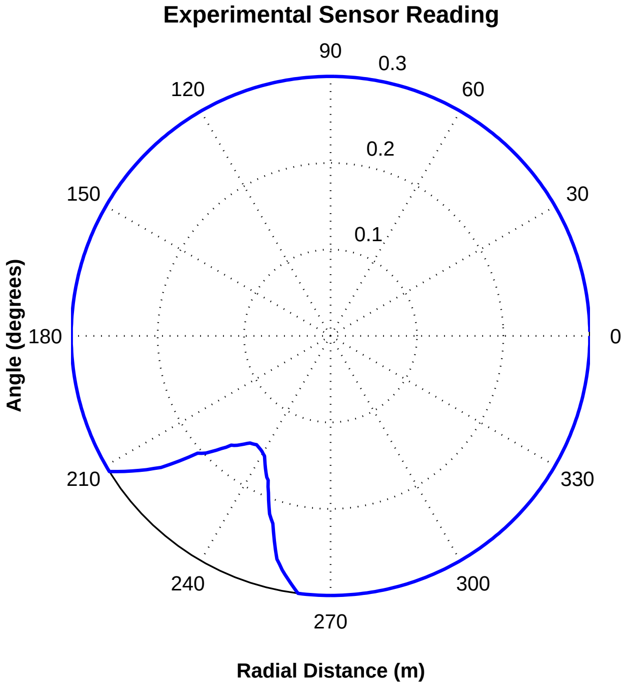

Fig. 28. Filtered Experimental Sensor Scan of Convex Corner

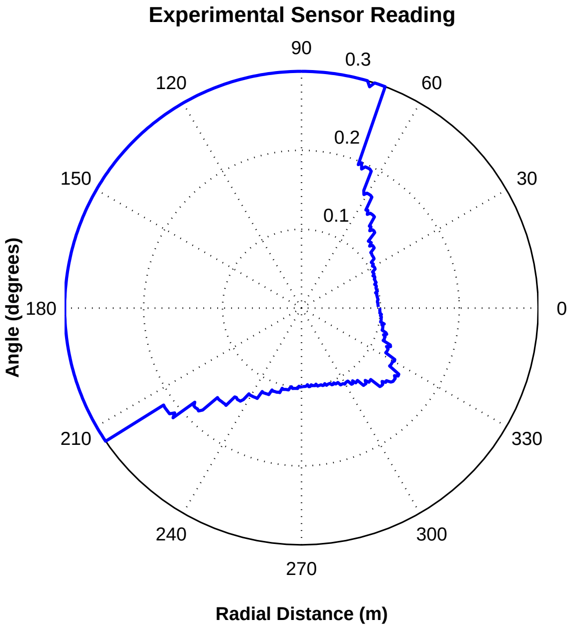

Fig. 29. Raw Experimental Sensor Scan of Concave Corner

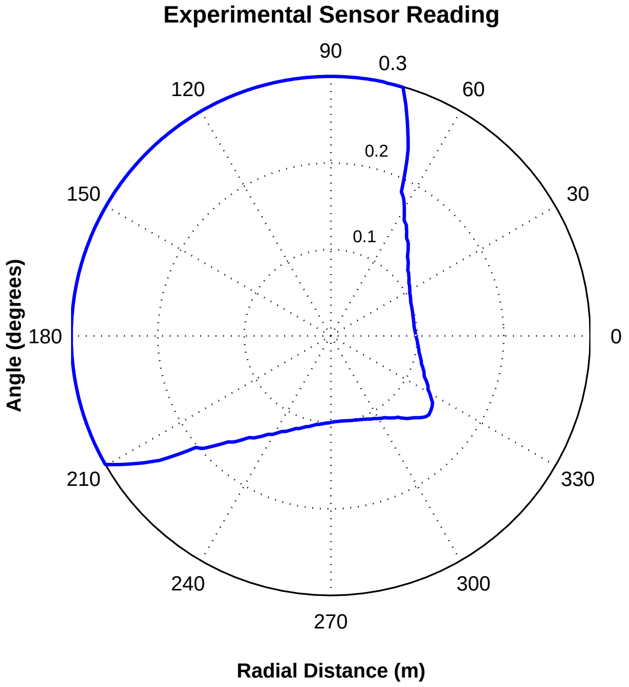

Fig. 30. Filtered Experimental Sensor Scan of Concave Corner

$$\mathrm{vec}_{\mathrm{wall}_2} = \frac{[\cos(\theta_2) \;\; \sin(\theta_2)]}{\|\cos(\theta_2) \;\; \sin(\theta_2)\|} \tag{6.7}$$

Once the cartesian coordinates for the corner are determined, the vectors for the walls are determined using equations 6.4-6.7. The vectors normal to each wall are found via equations 6.8 and 6.9. The offsets are found by projecting the vector to the corner along these vectors as shown in equations 6.10 and 6.11. Equations 6.12-6.16 are used to transform the offsets into environmental coordinates for each potential corner. The Cartesian offsets are found using equations 6.17 and 6.18. Figures 31 and 32 show experimental sensor scans with the calculated offsets superimposed for both the convex and concave corners.

$$\mathrm{vec}_1 = [0 \;\; 0 \;\; 1] \times \mathrm{vec}_{\mathrm{wall}_1} \tag{6.8}$$

$$\mathrm{vec}_2 = [0 \;\; 0 \;\; 1] \times \mathrm{vec}_{\mathrm{wall}_2} \tag{6.9}$$

$$\mathrm{offset}_1 = (\mathrm{corner} \cdot \mathrm{vec}_1)\mathrm{vec}_1 \tag{6.10}$$

$$\mathrm{offset}_2 = (\mathrm{corner} \cdot \mathrm{vec}_2)\mathrm{vec}_2 \tag{6.11}$$

$$\mathrm{A}_{\mathrm{scan}} = \begin{bmatrix} \mathrm{vec}_{\mathrm{wall}_1} \\ \mathrm{vec}_{\mathrm{wall}_2} \end{bmatrix} \tag{6.12}$$

$$\mathrm{A}_{\mathrm{env}} = \begin{bmatrix} \mathrm{vec}_{\mathrm{wall}_{1_{\mathrm{env}}}} \\ \mathrm{vec}_{\mathrm{wall}_{2_{\mathrm{env}}}} \end{bmatrix} \tag{6.13}$$

$$\mathrm{A}_{\mathrm{scan}}\mathrm{T} = \mathrm{A}_{\mathrm{env}} \tag{6.14}$$

$$\mathrm{T} = \mathrm{A}_{\mathrm{scan}}^{-1}\mathrm{A}_{\mathrm{env}} \tag{6.15}$$

$$\begin{bmatrix} \mathrm{offset}_{\mathrm{a}} \\ \mathrm{offset}_{\mathrm{b}} \end{bmatrix} = \begin{bmatrix} \mathrm{offset}_1 \\ \mathrm{offset}_2 \end{bmatrix} T \tag{6.16}$$

$$\mathrm{dx}_{\mathrm{corner}} = \mathrm{offset}_{\mathrm{a}} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \mathrm{offset}_{\mathrm{b}} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{6.17}$$

$$\mathrm{dy}_{\mathrm{corner}} = \mathrm{offset}_{\mathrm{a}} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \mathrm{offset}_{\mathrm{b}} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad (6.18)$$

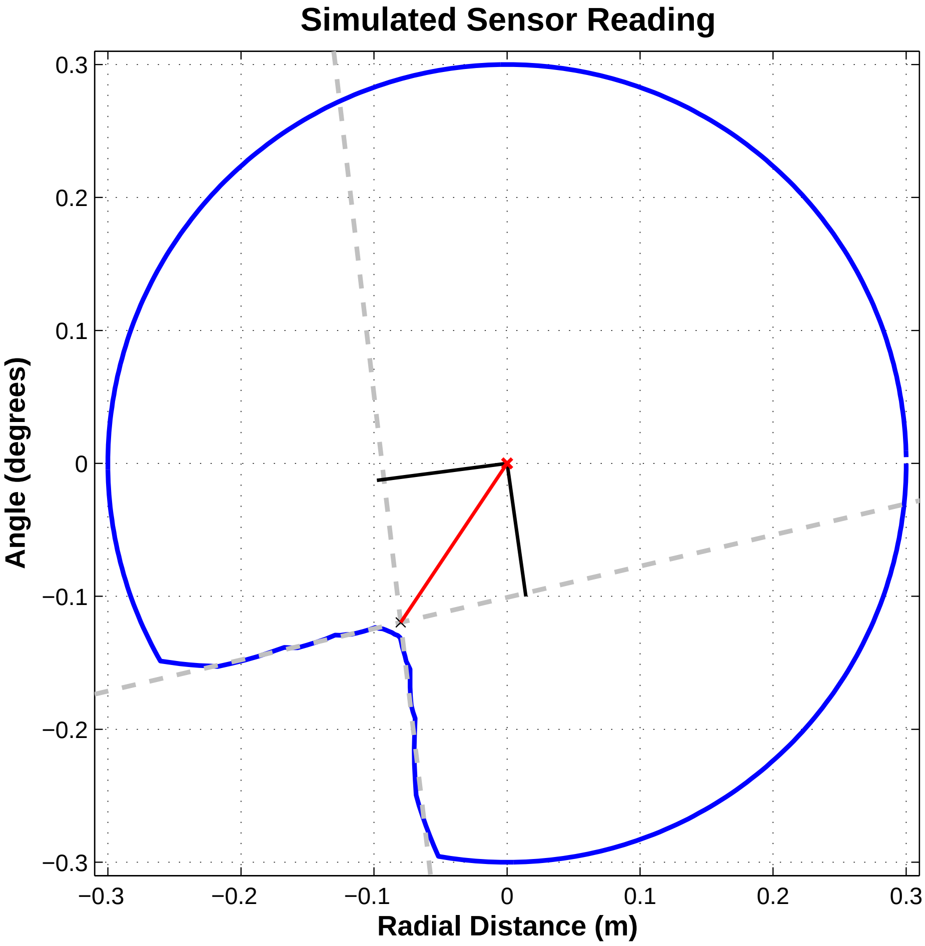

Fig. 31. Filtered Experimental Sensor Scan of Convex Corner with Offsets

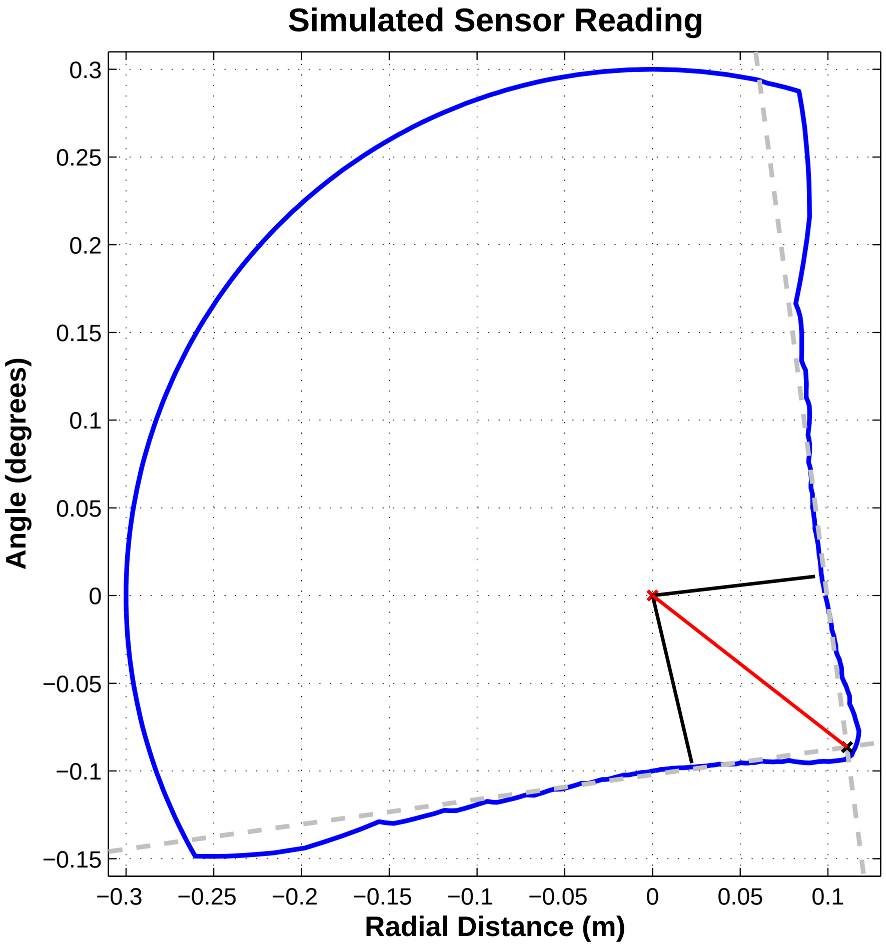

Fig. 32. Filtered Experimental Sensor Scan of Concave Corner with Offsets

C. Localization Results using an Infrared Range Sensor

For the experimental validation, the infrared sensor was placed in two different locations and scans were recorded. The virtual link length and relative heading information was calculated from the known location information, as explained earlier. Robot 1 was placed at (0.65, 0.85)m with a 180° degree heading and robot 2 was placed at (0.1, 0.1)m with a 180° degree heading. Figures 31 and 32 show the filtered sensor scans and the determined corner point for robots 1 and 2 respectively. Using the information from the sensor scans in addition to the virtual link length and relative heading information, the robots can be accurately localized.

Figure 33 shows the expected locations for robots 1 and 2 in the T-Shaped environment. Table III compares the experimental results to the nominal values.

Table III. Experimental Results in Localization

Category	**Robot 1**		**Robot 2**	
	Location	*Heading*	*Location*	*Heading*
Nominal	(0.65, 0.85) m	180°	(0.10, 0.10) m	180°
Experimental	(0.64, 0.86) m	181°	(0.10, 0.098) m	181°

Although the experimental results are encouraging and verify the localization method, there are some inherent limitations with both the camera and the IR range sensor. The limited distance over which the camera can effectively measure the virtual link length severely limits the utility of the proposed method. It is possible, that a camera with a higher resolution may mitigate this problem. With the continuous improvement in optical sensors and communications protocols, it is quite possible that a higher resolution camera in conjunction with a wireless network card to transfer

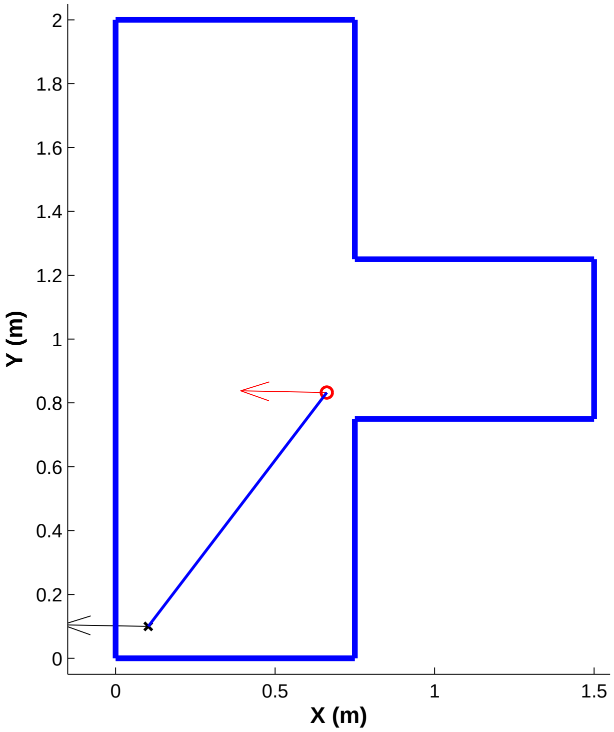

Fig. 33. Experimental Localization Results

the vision data to a PC will become available in the near future. Alternatively, a less hardware intensive localization method could be developed that only requires use of the relative heading information and not the virtual link length. This method would not require a higher resolution camera as the current available camera is capable of accurately determining the relative heading information over a significant range.

The experimental issues with the range sensor most likely can not be solved via improved hardware. As shown in figure 34, the walls of the environment appear curved. The figure is the same experimental data used to find the experimental results. However, the data was truncated at 0.5 m instead of 0.3 m. The walls appear curved due to the changing incidence angle of the IR sensor with respect to the wall. As is evident by the experimental sensor scans shown earlier, this effect can be reduced if the sensor remains close to the wall and its measured range is limited. However, this solution would severely restrict the movement of the robots. It would be better to develop another way of detecting the walls and/or corners in the environment that is not as sensitive to the "curved" walls.

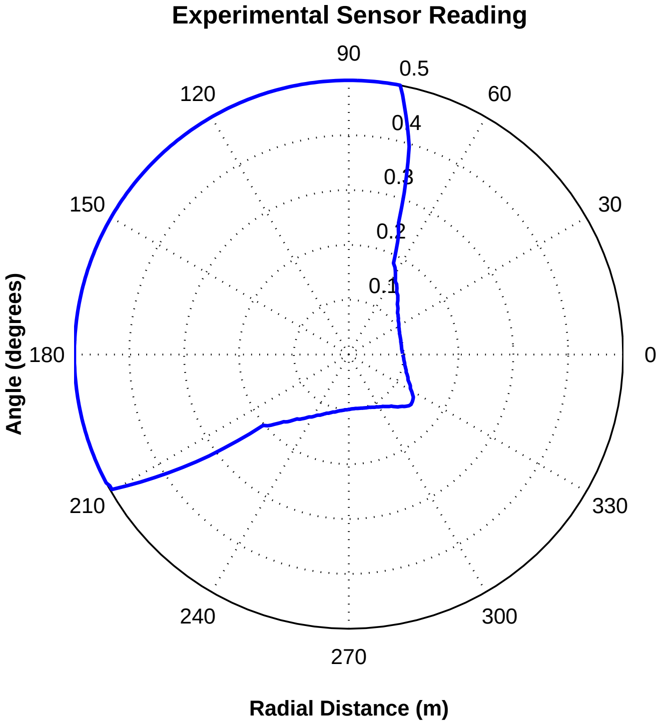

Fig. 34. Experimental Concave Sensor Scan Exhibiting the Effect of Changing Incidence Angle

CHAPTER VII

CONCLUSION

A localization method for multiple mobile robot systems using virtual links has been developed and verified. The method can be implemented using inexpensive rotating infrared range sensors and a rotating CMOS camera. A method that can be used to determine the virtual link length and relative heading information has been presented. However, due to limitations in available hardware the method was not implemented.

The simulation and experimental results verify that the localization method is sound. The issues of concern are hardware related. If an inexpensive higher resolution CMOS camera is not available for use in the localization process, a method which does not require the link length must be developed.

There is also a challenge associated with the infrared range sensor. The walls of the environment appear curved due to the changing incidence angle as the sensor rotates. This problem can be mitigated by maintaining a short distance between the sensor and the walls. However, this solution is not attractive as it limits the flexibility of the robot system. As an alternative to restricting the operating region of the robot system, it would be better to develop a more flexible method for detecting the walls and corners in the environment that is not sensitive to the curved appearance of the walls in the sensor scan.

REFERENCES

[1] K. Kato, H. Ishiguro and M. Barth, "Identifying and Localizing Robots in a Multi-Robot System Environment," in *International Conference on Intelligent Robots and Systems Proceedings Record*, IEEE/RSJ, Kyongju, South Korea, pp. 966-971, 1999.

[2] J. Spletzer, A.K. Das, R. Fierro, C.J. Taylor, V. Kumar and J.P. Ostrowski, "Cooperative Localization and Control for Multi-Robot Manipulation," in *International Conference on Intelligent Robots and Systems Proceedings Record*, IEEE/RSJ, Maui, Hawaii, pp. 631-638, Oct 29 - Nov 3, 2001.

[3] T. Nakamura, A. Ebina, M. Imai, T. Ogasawara and H. Ishiguro, "Real-time Estimating Spatial Configuration between Multiple Robots by Triangle and Enumeration Constraints," in *International Conference on Intelligent Robots and Systems Proceedings Record*, IEEE/RSJ, Takamatsu, Japan, pp. 2048-2054, 2000.

[4] E. Mustapha Mouaddib and B. Marhic, "Geometrical Matching for Mobile Robot Localization," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 542-552, October 2000.

[5] J. Kim, R. Pearce and N. Amato, "Robust Geometric-Based Localization in Indoor Environments using Sonar Range Sensorsm," in *International Conference on Intelligent Robots and Systems Proceedings Record*, IEEE/RSJ, Lausanne, Switzerland, pp. 421-424, October 2002.

[6] J. Kim, N. Amato and S. Lee, "An Integrated Mobile Robot Path (Re)Planner and Localizer for Personal Robots," in *International Conference on Robotics and*

Automation Proceedings Record, IEEE, Seoul, South Korea, pp. 3789-3794, May 21-26, 2001.

[7] Z. Feng-ji, G. Hai-jiao and K. Abe, "A Mobile Robot Localization Using Ulatrasonic Sensors in Indoor Environment," in *International Workshop on Robot and Human Communication*, IEEE, Sendai, Japan, pp. 52-57, 1997.

[8] R. Grabowski and P. Khosla, "Localization Techniques for a Team of Small Robots," in *International Conference on Intelligent Robots and Systems Proceedings Record*, IEEE/RSJ, Maui, Hawaii, pp. 1067-1072, Oct 29 - Nov 3, 2001.

[9] S. Thrun, W. Burgard and D. Fox, "A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping," in *International Conference on Robotics and Automation Proceedings Record*, IEEE, pp. 321-328, San Francisco, California, April 2000.

[10] D. Laurent, Mouaddib El Mustapha, Pgard Claude and Vasseur Pascal, "A Mobile Robot Localization Based on a Multisensor Cooperation Approach," in *International Conference on Industrial Electronics, Control and Instrumentation Proceedings Record*, IEEE, Taipei, Taiwan, pp. 155-160, Aug 5-10 1996.

[11] S. Lee, N. Amato and J. Fellers, "Localization based on Visibility Sectors using Range Sensors," in *International Conference on Robotics and Automation Proceedings Record*, IEEE, San Francisco, California, pp. 3505-3511, April, 2000.

[12] J.Y. Lee and S. Lee, "Consecutive Scanning Scheme: Application to Localization and Dynamic Obstacle Detection for a Mobile Robot," in *Proceedings of the International Mechanical Engineering Congress and Exposition*, ASME, New Orleans, Louisiana, Nov 17-22, 2002.

[13] J. Kim, R. Pearce and N. Amato, "Feature-Based Localization using Scannable Visibility Sectors," in *Proceedings of the IEEE International Conference on Robotics and Automation* (to appear), Taipei, Taiwan, 2003.

[14] K. Joarder and D. Raviv, "Autonomous obstacle avoidance using visual fixation and looming," in *Proceedings of SPIE*, vol. 1825, Boston, Massachusetts, pp. 733-744, 1992.

[15] D. P. Huttenlocher, M. E. Leventon, and W. J. Rucklidge, "Visually-guided navigation by comparing edge images," in *Algorithmic Foundations of Robotics*, pp. 85-96, Ken Goldberg, Randall Wilson and Dan Halperinm, Eds. San Francisco, California, 1994. Publisher: Wellesley, Massachusetts: A.K. Peters, 1995.

[16] T. D. Williams, "Depth from camera motion in a real world scene," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2 No. 6, pp. 511-516, 1980.

[17] E. Sahin and P. Gaudiano, "Visual looming as a range sensor for mobile robots," in *Fifth Intl. Conf. on Simulation of Adaptive Behaviour*, Zurich, Switzerland, pp. 114-119, 1998.

APPENDIX A

USING A ROTATING CAMERA TO DETERMINE VIRTUAL LINK LENGTH AND RELATIVE HEADING

A. Image Sensor

We have seen so far that by using the infrared range sensor reading we can detect the occurrence of edges, corners and points of interest near individual robots. These information are of little use if we cannot find the relative distance and heading between the robots. In the majority of the mobile robots that use sonar sensors, the range information is obtained from the time-of-flight. However from these measures, the only information that can be obtained is the distance to the closest point to the robot that reflected the wave, back to the sensor. This is not sufficient to characterize the object, whose presence was detected. In our example, the sonar sensor will not be able to differentiate between the edges, the walls, the corners and other robots. Moreover, the wide opening angle presented by most sonar sensors introduces a uncertainty factor along the direction of measure. From a range value we can only say that there is a region in which every point is a possible location for the detected obstacle. Considering real sensors, that are subject to errors, the result is that for each range measure, instead of having one arc where every point has the same probability of being the echo generator, we have a spacial probability distribution. Due to these reasons, we have decided to use a CMOS camera to find the distance to other robots.

1. Technical Details

Visual looming has been used to extract distance information from images in the mid nineties. Visual looming technique estimates the distance to an object using the change in the projection size of the object that results from known robot displacements. Visual looming has been used for obstacle avoidance [14]. Extracting the depth of an object from an image is dealt in [15] [16] [17]. We have used a modification of the visual looming technique, in which we calculate the pixel width of a known cylindrical object mounted on top of the robot. This pixel width is inversely proportional to the distance from the camera, Eq. A.1. The geometric relations shown in Figure 35, assumes the camera is viewing an object of width 'w'. Two different position of the object is shown, at distances d_{s1} and d_{s2} from the camera. The equation for a fixed width, w, of the cylinder and the constant focal length, f, is

$$\frac{p_1}{f} = \frac{w}{d_{s1}} \tag{A.1}$$

The object should be along the focal axis of the camera. The camera needs to be calibrated beforehand for a given width of the cylindrical object. A calibration figure for the experimentation performed is shown in the Figure 36.

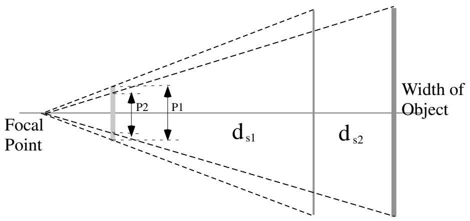

Fig. 35. Geometric Representation of Two Configurations

Given the width of the object in any picture, the distance can be calculated by interpolating along the curve shown in Figure 36.

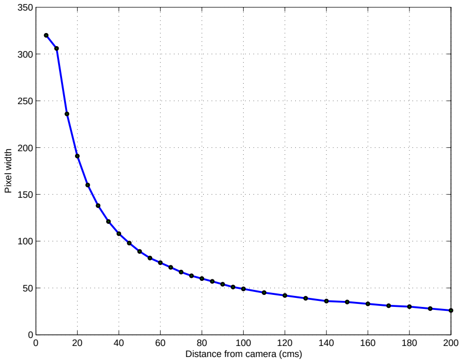

Fig. 36. Calibration Data for the Camera

2. Using a CMOS Camera to Determine the Length and Relative Orientation of the Virtual Link

To find the distance and heading between robots, we have used a CMOS camera. The image size is $80X143$ pixels and allows for serial communication with the computer or a microcontroller. The camera is mounted on a stepper motor and image processing part of the camera can successfully track an object of a given color at 15 fps. The encoder reading provides us with the relative heading between two robots and an extension of *The Looming effect* provides us with the distance to the other robot. Pictures from the image processing is shown in Figure 37. Calculating the width of the image from Figure 37 is easier and less computationally extensive compared to stereoscopic imaging.

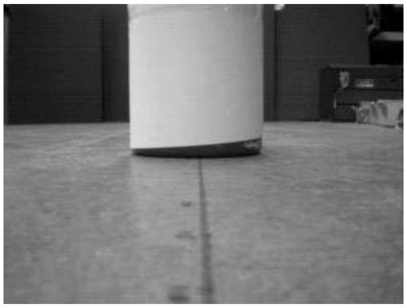

Fig. 37. Grayscale Image of the cylinder used to Calibrate the camera

VITA

Andrew John Rynn earned both his Bachelor of Science and Master of Science degrees at Texas A&M in the Department of Mechanical Engineering. He completed his Bachelor of Science in May 2000 and his Master of Science in August 2003. The focus of his graduate course work was systems and controls; his research was conducted in the field of robotics.

During his undergraduate career, Andrew worked as a co-operative education student for The Boeing Company in St. Louis, Missouri. During his tenure at Boeing, Andrew worked in advanced manufacturing research and development for two terms in addition to time spent in strength and design, flight test and C-17 production support. During his second rotation in advanced manufacturing technology, he had the opportunity to work as an integral member of a team developing a robotic drilling project for F-18 E/F production.

After receiving his undergraduate degree, Andrew spent the following two summers working as an engineering associate at Ford Motor Company in Dearborn, Michigan. He spent his first summer developing and analyzing aerodynamic models for Ford Taurus NASCAR test drive data. His second term was served in the climate control area. In this position, his primary responsibility was to manage climate control systems test data and attempt to correlate the test results and customer satisfaction.

After defending his thesis research, Andrew began work at Lockheed Martin in Fort Worth, Texas. He is working as a stress analyst on the JSF Wingbox.

Andrew can be reached at his residence in Fort Worth - 5175 Faith Dr. Apt 312, Fort Worth, TX 76120 - for the foreseeable future. However, his parents can be reached - to track him down if necessary. Their address is - 6 Ash Branch Ct., The Woodlands, TX 77381.