Sensor-Based Self-Localization for Wheeled Mobile Robots

A.Curran and K.J.Kyriakopoulos
New York State Center for Advanced Technology in Automation and Robotics
and
Electrical, Computer and Systems Eng. Dept.
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Abstract

In this paper, we demonstrate a reliable and robust algorithm to localize a mobile robot in a relatively consistent with an apriori map indoors environment. This algorithm uses an Extended Kalman Filter that combines dead-reckoning, ultrasonic, and infrared sensor data to estimate current position and orientation. Through a thresholding approach, unexpected obstacles can be detected. Experimental results from implementation on our mobile robot, Nomad-200, are also presented.

1 Introduction

Consider a typical indoors structured environment as shown in figure 1. Given ultrasonic, infrared, odometry data and a representation of the environment, a robust method for localizing a moving mobile robot is desired. This localization method should not alter the environment by uses of beacons or markers or be influenced by unexpected objects not depicted in the apriori environment.

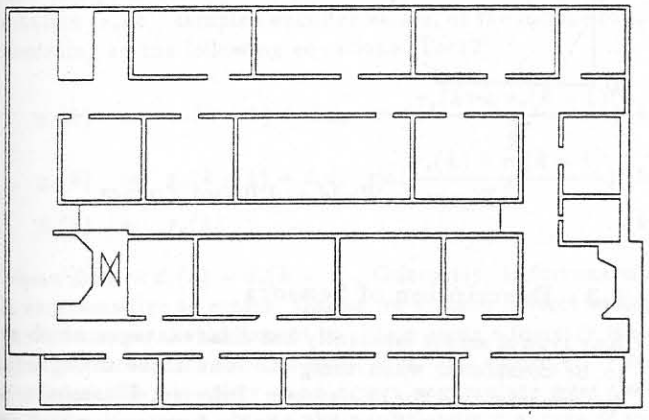

Figure 1: Typical Indoor Environment

Obtaining accurate position and orientation information of a mobile robot is not new. Various sensors have been used in the localization procedure, i.e. Vision [KK92], Optical Range finders [Cox91], Ultrasonic Beacons [Kle92], and Ultrasonic sensors [Cro89, AM91, HMB92]. The three main objectives in the design of a localization system is speed, cost and accuracy.

Vision guided localization by Kak and Kosaka, [KK92] resulted in superior accuracy but only with the reduction of speed and increased cost. Ultrasonic beacon estimation dramatically improves speed with an expected decrease in accuracy but also requires that the environment be redefined by the addition of ultrasonic beacons. Research in pure ultrasonic ranging has been directed towards statistical map matching, wall matching, and individual sensor matching techniques. A static ultrasonic localization method by statistical map matching, [HMB92], requires a large amount of computational time. A wall matching techniques proposed by Cox,[Cox91] performs well using optical rangefinders but its performance would likely decrease with the large beamwidth of ultrasonic sensors as compared to optical sensors since it requires a sensor with a small beamwidth. A similar wall matching technique by Crowley, [Cro89] uses ultrasonic sensors with nice results. Crowley's method requires finding at least 3 consecutive sensors which form a line, thereby indicating the existence of a wall in the environment. Therefore, many sensors containing relevant data may be ignored. Preciado et.al. implemented a recursive method which only incorporates sensor information which improves the estimate. [AM91]. Preciado et.al., does not attempt to accurately determine the expected sensor values by examining the entire sensor beamwidth but places a variance encompassing the entire beamwidth. In addition, using the recursive formula stated, angle orientation is initially updated. Subsequently, position is updated. This was not proven to be optimal as is a Kalman filter, which updates orientation and position simultaneously.

We want to design a system which can combine all relevant sensor information while the robot is navigating and use said information to accurately localize the robot. In addition this system should identify areas possibly containing obstacles not indicated in the apriori map. By defining these areas, an avoidance algorithm can be designed to optimally avoid detected obstacles [KS92b].

A system that uses Kalman filters and thresholding that successfully localizes a mobile robot with or without the addition of unknown obstacles in the environment is presented here.

As shown in figure 2 we are determining a rough estimation of postion through changes in the incremental encoders values, X_r, Y_r and θ_r. If we filter the sensor information, Z, to obtain only relevant information and combine this with

the rough estimation of position by the encoders, we can determine a better estimate of the postion using a Kalman filter. With this localized postion, we can better estimate postion future position by indicating changes between the encoder postion and real world postion. Finally, we use a nonholonomic motion controller which uses the Kalman estimated postion to conveys translation and rotation information to the mobile robot to Navigate the vehicle and complete the system loop

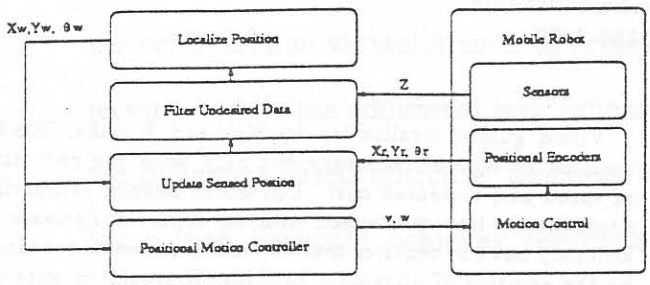

Figure 2: Program Layout

2 Problem Statement

In order to be able to mathematically state the problem, a descriptive analysis of the vehicle and sensor model is necessary.

2.1 Description of Robot

The robot is a 3-wheeled, cylindrical, zero-gyro radius robot. Sensor information is obtained from ultrasonic, tactile, and infrared sensor rings. In each ring, there are 16 individual sensors located at 22.5 degree increments around the robot. Odometry measurements are obtained from encoders located on a synchronous drive system. The robot can be seen in figure 3.

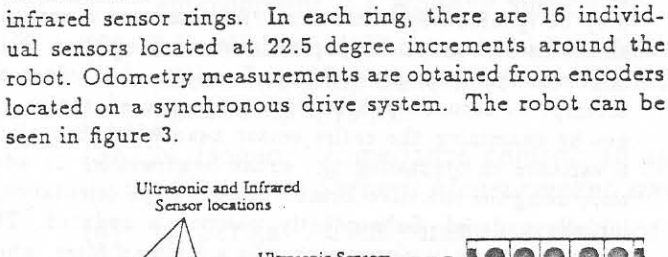

Figure 3: Mobile Robot Sensor Configuration

2.2 Vehicle Model

Assuming a two dimensional world, we can define the robot configuration w.r.t a world coordinate frame W by vector $X_w = [x_w\ y_w\ \theta_w]^T$ containing its position and orientation. We consider another coordinate frame R which is the world coordinate frame that defines the motion of the robot based on odometry. W and R are not the same, in general, because R includes the uncertainty accumulated by the integration of the error during rolling. Let's denote the configuration of the robot w.r.t R by $X_r = [x_r\ y_r\ \theta_r]^T$. The motion equations w.r.t to R can be given by the unicycle model [PK93] :

$$\dot{X}_r = f(X_r) \cdot u \tag{1}$$

where

$$f(X_r) = \begin{bmatrix} \cos(\theta_r) & 0 \\ \sin(\theta_r) & 0 \\ 0 & 1 \end{bmatrix} \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2}$$

and u_1, u_2 are the translational and rotational velocities respectively. The motion equations w.r.t W can be easily obtained if the odometry error (wheel slippage, etc.) is included as noise $n = [n_1\ n_2\ n_3]^T$ and an approximate discrete time model is used then

$$\dot{X}_w = f(X_w) \cdot u + n \tag{3}$$

It is obvious that the addition of noise creates the relative motion between frames R and W, depicted in figure 4.

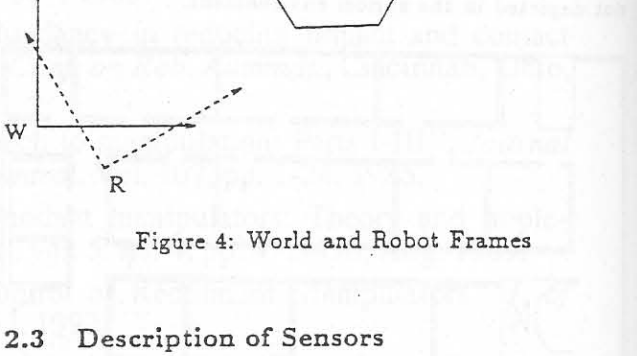

Figure 4: World and Robot Frames

2.3 Description of Sensors

Any type of sensor inherently has disadvantages which need to be considered when using it. The three main sensing instruments used for this system, Infrared, Ultrasonic, and positional encoders have distinct disadvantages which need to be addressed. In the following paragraphs, an overview of the problems associated with each sensor are covered.

2.3.1 Ultrasonic Sensors

Each ultrasonic sensor has a beamwidth of approximately 23.6° [BK91] By examining all 16 sensors, we can obtain a 360° panoramic view fairly rapidly. Unfortunately, ultrasonic sensors work upon a send/receive echo type format.

Therefore, no two sensors can be simultaneously activated unless specific software and/or hardware is included to distinguish the different sonar signals [BK88]. In order to decrease crosstalk, (the transmit pulse of sensor "i" being received and causing an erroneous value on another sensor), we must fire each sensor individually. In addition, since the Polaroid sensor modules combine a transmit and receive system into one compact package, we must blank the receive system such that the residual transmitted pulse on the sensors in not detected as a received pulse. Therefore, with the sonar system we can detect objects from a minimum range of 17 inches to a maximum range of 22 feet with a 30 degree resolution.

2.3.2 Infrared Sensors

Similar to the sonic sensors, infrared sensors work upon a send/receive format. These sensors emit an infrared light from one source, and measure the amount of reflected light with two light detectors. Since these devices measure light differences, they are highly biased by the environment. Object color, object orientation, and ambient light all can contribute to erroneous readings but since the transmission signal is light instead of sound, we may expect a dramatically shorter cycle time for obtaining all infrared sensor measurements. Considering all these problems as noise factors, we can assume and prove if necessary that for this type of infrared detector, infrared measurement are only acceptable for short distances. In our system, infrared sensors were used to provide information for the shorter to 17 in. area, where the ultrasonic sensors are not used.

2.3.3 Odometry Measurements

A low-level integration routine calculates odometric location using the current and previous translation ($d_e(k)$), and rotation ($r_e(k)$) sampled encoder values, of the mobile robot according to the following equations [Tec12].

$$x_r(k) = x_r(k-1) + \Delta d_e \cdot \cos\left(\frac{r_e(k) + r_e(k-1)}{2}\right) \quad (4)$$

$$y_r(k) = y_r(k-1) + \Delta d_e \cdot \sin\left(\frac{r_e(k) + r_e(k-1)}{2}\right) \quad (5)$$

$$\theta_r(k) = r_e(k) \quad (6)$$

where $\Delta d_e = d_e(k) - d_e(k-1)$. Odometry, unfortunately, is very sensitive to errors. Unless we assume perfect rolling conditions, we should expect to obtain some odometry measuring errors in the form of drift, bias, and slippage. When dead-reckoning is solely used errors accumulate over time as integration error, due to the nonholonomic nature of the rolling motion. Over short travel distances we can expect insignificant errors but over long paths these errors will grow.

2.4 Mathematical Statement of Problem

If dead-reckoning was used to obtain the configuration X_w from integration (i.e if integration of eq. 1 was only used) then over large distances, this position would contain large integration errors. Therefore, the knowledge provided by the system equations should be complemented by this obtained by the ultrasonic and infrared sensors. Since those sensor measurements are not overlapping (i.e. only one range value is considered valid for each sector), we can combine these measurements thereby creating a sensor vector Z, which contains 16 elements signifying the distance for each sector.

Therefore, we are looking for some function $\mathcal{F}$, that sequentially provides optimal estimates

$$\hat{X}_w(k) = \mathcal{F}(\hat{X}_w(k-1), Z(k)) \quad k = 1, 2, \ldots \quad (7)$$

of the configuration of the robot by combining the system equations knowledge and the sensor readings. Optimality is in the sense that the estimates should minimize an error criterion

$$J = E\left\{(\hat{X}_w(k) - X_w(k))^T \cdot (\hat{X}_w(k) - X_w(k))\right\} \quad (8)$$

where $E\{\cdot\}$ denotes the expected value of a random variable.

3 Approach of Solution

The above stated problem can be attacked through the use of the Extended Continuous-Discrete Kalman filter [Gel80]. In the following subsections, the adaptation of the Kalman filter to handle this problem is presented.

3.1 Model Equations

The system and measurement models are :

$$\dot{X}_w = f(X_w(t)) + n \quad (9)$$

$$z_k = h_k(X(t_k)) + v_k \quad k = 1, \ldots, 16 \quad (10)$$

where $n \sim N(0, Q)$, $v_k \sim N(0, R)$, are the odometry and sensor noises, while z_k are the sensor measurements for sector k. Finally, h_k, the k-th measurement function, is defined as the function which relates current configuration with the measurement that is expected to be received from the k-th sensors.

3.2 Measurement Functions

Given the current configuration X_w of the robot, we need to calculate the expected sensor values for each sector. This is actually the minimum distance ray intersecting a wall contained in the k^{th} sensor's beamwidth. Denoting x_k^i and y_k^i as the coordinates of the minimum distance intersection for sensor k, and $x_{s_k}^i$ and $y_{s_k}^i$ as the coordinates of the k^{th} sensor, at time instant t_i, (see figure 5), we formulate the distance equation as :

$$h_k(X_w(t_i)) = d_k^i = \sqrt{(x_k^i - x_{s_k}^i)^2 + (y_k^i - y_{s_k}^i)^2} \quad (11)$$

where $x_{s_k}^i$, $y_{s_k}^i$, x_k^i, and $y_{s_k}^i$ are all functions of X_w.

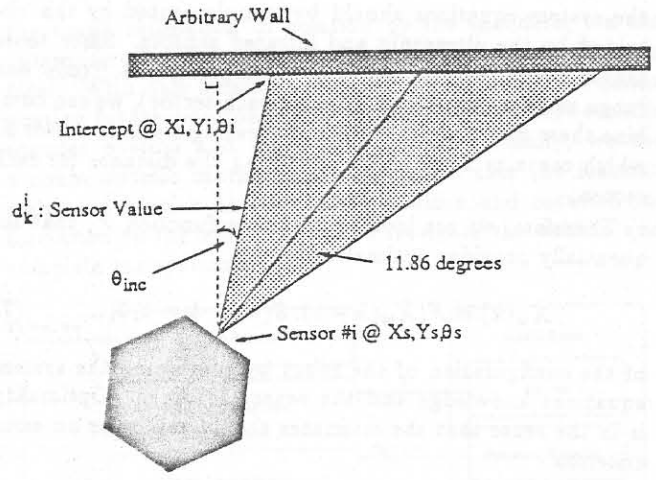

Figure 5: Derivation of Sensor Measurement

3.3 State Propagation

The state estimation propagation equation is :

$$\dot{X}_w = f(X_w(t)) \tag{12}$$

In the implementation of the integration of the above equation we encounter timing problems due to the timing delay between decision of a new control and low level implementation. If we assume that over short distances odometry is accurate, a fairly accurate odometry value based upon incremental encoder changes can be obtained. By reformulating the propagation equation we have :

$$\dot{X}_w \delta t = \begin{pmatrix} \cos(\theta_{diff}) & -\sin(\theta_{diff}) & 0 \\ \sin(\theta_{diff}) & \cos(\theta_{diff}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta x_r \\ \Delta y_r \\ \Delta \theta_r \end{pmatrix} \tag{13}$$

The rotation of the incremental odometry changes is necessary to realign the robot and fixed frames. θ_{diff} is the difference between the estimated and odometric angle of the wheels, which is equivalent to the difference between the world and mobile robot frames.

3.4 Covariance Matrix Propagation

The error covariance matrix is defined as $P = E\left\{(\hat{X}_w(k) - X_w(k)) \cdot (\hat{X}_w(k) - X_w(k))^T\right\}$ and its time propagation is governed by the following matrix Ricatti equation

$$\dot{P}(t) = F(\hat{X}_w(t))P(t) + P(t)F^T(\hat{X}_w(t)) + Q \tag{14}$$

In order to improve the accuracy of the error covariance propagation, we decided to use a Taylor series second order approximation to the propagation equation by using the 2^{nd} order time derivative of the error covariance matrix

$$\ddot{P}(t) = \frac{\partial \dot{P}(t)}{\partial \hat{X}_w} \cdot \dot{X}_w \tag{15}$$

where F is defined as

$$F(\hat{X}_w(t), t) = \frac{\partial f(X(t))}{\partial X(t)}|_{X(t)=\hat{X}_w(t)} \tag{16}$$

3.5 Filter Equations

The nonholonomic model of the mobile robot and the measurement function h_k that is of trigonometric form, are both inherently nonlinear. Therefore, an extended version of the Kalman filter is necessary. Listed below are the Filter Equations :

$$\begin{aligned} \hat{X}_k(+) &= \hat{X}_k(-) + K_k[z_k - h_k(X_k(-))] \\ P_k(+) &= [I - K_k H_k^T(\hat{X}_k(-))]P_k(-) \\ K_k &= P_k(+)H_k^T R_k^{-1} \end{aligned} \tag{17}$$

with H_k defined as

$$H_k(\hat{X}_k(-) = \frac{\partial h_k(X(t_k)}{\partial X(t)}|_{X(t_k)=\hat{X}_k(-)} \tag{18}$$

where $\hat{X}_k(-)$ is the expected state just before the arrival of the k-th measurement (it is obtained by state equation integration, as described in section 3.3), $\hat{X}_k(+)$ is the expected state just after the arrival of the k-th measurement, $P_k(-)$ is the error covariance matrix just before the arrival of the k-th measurement (it is obtained by error covariance matrix equation integration, as described in section 3.4) and $P_k(+)$ is the error covariance matrix just after the arrival of the k-th measurement.

There are two different methods in order to calculate the Kalman gain matrix K_k. [Gel80] We chose the form shown above in order that the largest matrix inversion would be of a 3x3 matrix.

3.6 Unexpected Obstacles

During robot operation, various unexpected obstacles may interfere with robot localization. Suppose, an unexpected obstacle detected by sensor j is located in the environment as shown in figure 6. In addition, suppose that sensor "i" was predicted to catch the corner at A as its distance measurement, missed corner A.

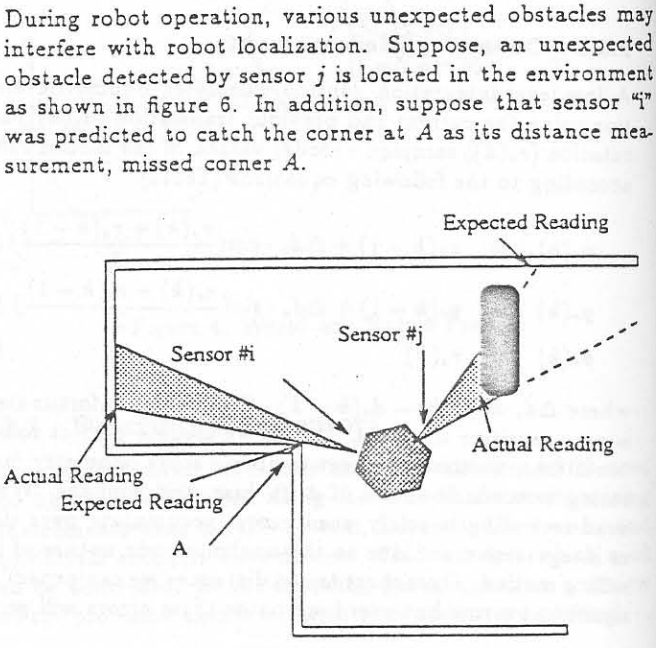

Figure 6: Sensing Discrepancies

Consider now the profile of the difference value (i.e actually received range signals - predicted range signals) from the 16 sensors as presented in figure 7. If we examine the

difference between predicted and actual sensor information for this example, we would notice a large discrepancy between valid and incorrect data for those two cases.

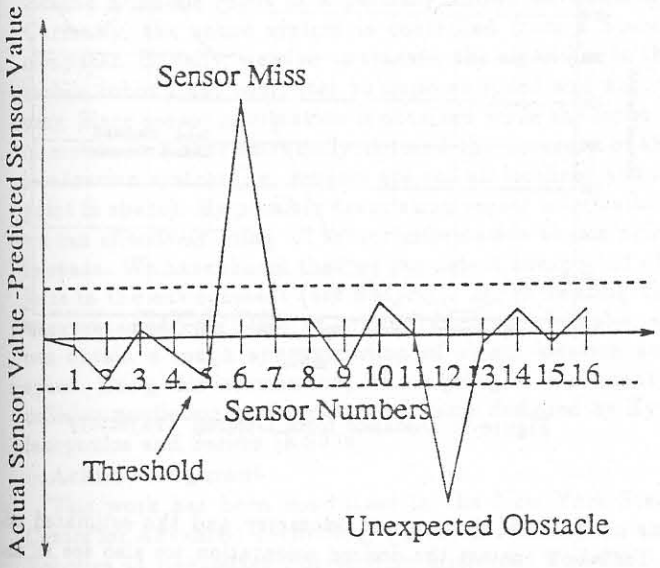

Figure 7: Sensor Comparison

Therefore, an alteration is introduced so that the Kalman filter will not be incorrectly biased by sensor readings i and j. In normal operation, each sensor has a variance associated with it, indicating the uncertainty that we have about the incoming data from the sensor. If we decrease this uncertainty factor for a given sensor, we decrease the bias that the robot has from that sensor. By significantly increasing this uncertainty factor we essentially remove its contribution to the Kalman filter. If a threshold is created, as shown in 7, then all values outside the threshold are considered incorrect and ignored.

In addition to unbiasing the filter due to incorrect measurements, this method also indicates possible unexpected obstacles which a collision avoidance scheme could avoid. In the event that an unmodelled object is in the robot's sensors field of view, the sensors detecting said object will return values considerably lower than the predicted values. When comparing the actual to predicted sensors value, we will obtain a large negative discrepancy. Now if consider the problem of missed edges, we will obtain the inverse. In the missed edge case, we will obtain actual values dramatically larger than the predicted. Therefore, through this thresholding method we can identify the type of sensor discrepancy, missed edge or unmodelled feature.

4 Implementation

Implementation of the localization system required compromises based on numerous experiments. In the following paragraphs, problems and solutions associated with each area are highlighted.

4.1 Definition of the Apriori Map

In order to make the system stable, we needed to minimize the number of calculations per Kalman Filter cycle. We decided to simplify the apriori map definition so that the calculations involved in $h(X_w)$ are minimized. In ignoring any fine detail of the hallway (i.e. The 49 doors which are depressed from the hallway by approximately 5 inches) and selecting motion control gains accordingly, we decrease the robot's sensitivity to a non-ideal map. By ignoring these doors, we unfortunately increase the uncertainty in position and orientation.

4.2 Sensor Variances

The accuracy of ultrasonic sensors is highly dependent on the texture of the reflecting surface. Rough surfaces will return the wavefront produced by the sensor at any incident angle as opposed to smooth surfaces which will return only wavefronts which are incident on the surface at near to perpendicular angles. Through experimentation, the variance dependency on distance was found. Generic ultrasonic sensors do not contain any orientation information of the reflecting surface, though it may be possible to obtain feature information as proposed by [BK91]. If we assume we can accurately predict robot location and orientation, we may place a orientation variation to improve the system response. Through additional experimentation with the entire system an orientational variance was determined. The resulting variance equation for sensor "i" is :

$$R_{ii} = d_k^i \cdot (0.04 + 0.5\,|\sin(\theta_{inc})|) \tag{19}$$

where d_k^i is defined in eq. 11 and θ_{inc} is the angle from perpendicular that the predicted sensor ray encounters the wall. Both parameters are depicted in figure 5.

4.3 Odometric Variances

Odometric variances are dependent on the type of floor surface. In our experiments, we have a carpeted floor. Through experimentation we obtained the variances $diag(Q) = [0.001\ 0.001\ 0.0001]^T$, where Q was defined in section 3.1.

4.4 Motion Controller

The selected motion control algorithm was of the form

$$u_1 = -k(x_e \cos\theta_w + y_e \sin\theta_w) + \dot{x}_d \cos\theta_w + \dot{y}_d \sin\theta_w \tag{20}$$

$$u_2 = \frac{1}{\epsilon}[-k(y_e \cos\theta_w - x_e \sin\theta_w) - \dot{x}_d \sin\theta_w + \dot{y}_d \cos\theta_w] \tag{21}$$

where $x_e(t) = x_w(t) - x_d(t)$, $y_e(t) = y_w(t) - y_d(t)$ are the position errors and $x_d(t)$, $y_d(t)$ are the reference trajectories. It was implemented in order to track a desired trajectory and designed by Pappas and Kyriakopoulos [PK93]. This algorithm was chosen because of it is highly robust in correcting any type of error in robot position.

5 Experiemental Results

The proposed Extended Kalman Filter localization algorithm was tested using a Nomadic Technologies' Nomad 200 mobile robot. The Nomad 200 is comprised of a 486 33Mhz Computer, with a radio link of 19200 baud rate to a Unix Sparc2 Computer. Low level motion control is achieved through a Gahlil motion controller. Currently, we are controlling motion through a rotational and translational velocity command. The Sensing capabilities of the Nomad 200 are 16 ultrasonic, infrared and tactile sensors. Currently, the localization algorithm is running on a Sparc2 computer. This was done only to facilitate corrections to the localization code. The total cycle time for localization and Motion control, including modem communication to the robot, is approximately 150-200 milliseconds.

In order to test the localization method, we selected to localize and control the robot in our floor hallways. In these experiments, some of the laboratory doors were left open to test if erroneous reading are rejected. Seventy Percent of the walls in our lab are smooth plasterboard which results in poor ultrasonic measurement for sensors not parallel to the wall. The remaining thirty percent of the walls are 9 inch square bricks. The depressions in the brick usually give a fairly accurate reading up to 50 degrees from perpendicular. The entire path is a carpeted floor. A rectangular path was selected with a total distance of 360 feet. Robot Velocity was set at 5 inches/second because of the slow time to update all sonic sensors.

We can see in figure 8 that we can successfully localize the mobile robot, even though odometry has dramatically failed. The slight deviations we see from a path directly down the center of the hallway is due to room and elevator doors that we did not model.

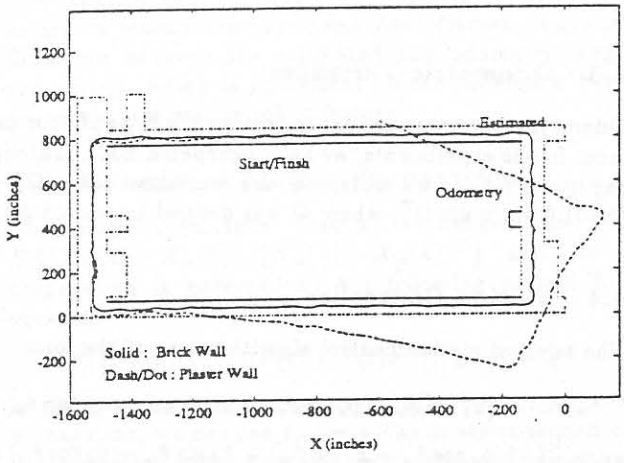

Figure 8: Mobile Robot Navigation

If we compare the difference between odometry and the Kalman Filter based estimated position against the desired trajectory (see fig. 9) we can see that the localization algorithm presented successfully removes the integration error existent in the odometry.

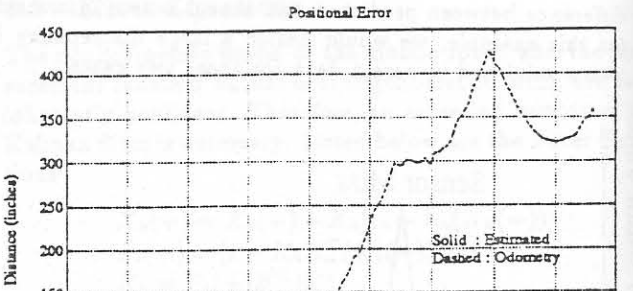

Figure 9: Distance from Desired Trajectory

Finally, if we compare odometry and the estimated orientation against the desired orientation we also see an improvement (see fig. 10). The 10 degree oscillations in the routine are due to the oscillatory converging behavior of the control algorithm and unmodelled doors.

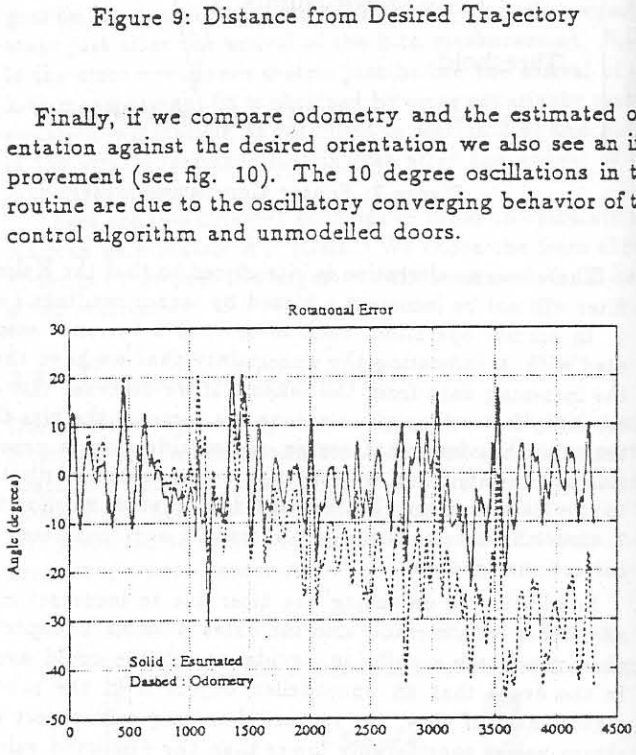

Figure 10: Angle from Desired Rotation

In tabular form we obtained the following results from 10 successive runs.

Maximum X error (Navigation) 18.1" Maximum Y error (Navigation) 12.5" Maximum θ (Navigation) 25.8° Maximum Normed Error (Navigation) 18.1"

Final X error 30.0" Final Y error 3.0" Final θ error 10°

The errors in Y and θ during navigation can mostly be attributed to the unmodelled door jams. The X error is attibuted to the featureless in X information obtained from the long hallways. The larger errors at the final position are partly due to the nonholonomic controller.

6 Discussion - Future Work

Using this method, we have shown that we can successfully localize a mobile robot in a partially known environment. Currently, the entire system is controlled from a Sparc2 computer. Initially, we plan to transfer the algorithm to the mobile robot's 486 computer to improve speed and autonomy. Since sensor information is obtained while the robot is in motion, we have essentially reduced the accuracy of the localization system (i.e. sensors are not all localized to one point in space). By possibly translating sensor information, we can effectively bring all sensor information to one point in space. We have shown that we can detect unexpected objects in the environment (see figure 7). By correlating the unexpected objects detected between successive samples, we can obtain a rough approximation of object location and speed. Using this information, we are going to implement a collision prediction and avoidance scheme designed by Kyriakopoulos and Saridis [KS92a].

Acknowledgment

This work has been conducted in the New York State Center for Advanced Technology (CAT) in Automation and Robotics at Rensselaer Polytechnic Institute. The CAT is partially funded by a block grant from the New York State Science and Technology Foundation. The first author has been additionally supported by a scholarship from Raytheon Company. The authors would also like to thank Nomadic Technologies for their assistance.

References

[AM91] Predciado A. Meizel D. Segovia A. and Rombaut M. Fusion of mutl-sensor data: A geometric approach. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 2806–2811, April 1991.

[BK88] J. Borenstein and Y. Koren. Obstacle avoidance with ultrasonic sensors. *IEEE Journ. Robotics and Autom.*, pages 213–218, April 1988.

[BK91] O. Bozma and R. Kuc. Building a sonar map in a specular environmen using a single mobile sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1260–1269, December 1991.

[Cox91] I.J Cox. Blanche - an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, pages 193–204, April 1991.

[Cro89] J.L Crowley. Asynchronous control of orientation and displacement in a robot vehicle. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pages 1277–1282, May 1989.

[Gel80] Arthur Gelb. *Applied Optimal Estimation*. M.I.T. Press, Cambridge, Ma., 1980.

[HMB92] A. Holenstein, M. Muller, and E Badreddin. Mobile robot localization in a structured environement cluttered with obstacles. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2576–2581, May 1992.

[KK92] A. Kosaka and A.C. Kak. Fast vision-guided mobile robot navigation using model- based reasoning and prediction of uncertainties. *Computer Vision Graphics and Image Processing, Image Understanding*, November 1992.

[Kle92] L. Kleeman. Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2582–2587, 1992.

[KS92a] K. Kyriakopoulos and G. Saridis. An integrated collision prediction and avoidance scheme for mobile robots in non-stationary environments. In *Proceedings of 1992 International Conference on Robotics and Automation, and to Appear in AUTOMATICA*, 1992.

[KS92b] K. Kyriakopoulos and G. Saridis. Optimal motion planning for collision avoidance of mobile robots in non-stationary environments. In *1992 American Control Conf. and submitted to Transactions in Automatic Control*, June 1992.

[PK93] G.J Pappas and K.J. Kyriakopoulos. Dynamic modelling and tracking control of nonholonomic wheeled vehicles. In *Submitted to IFAC '93 World Congress*, 1993.

[Tec12] Nomadic Technologies. Nomad 200 motion control software. Technical report, Nomadic Technologies, Inc., 19912.