Further Results with Localization and Mapping using Range from Radio

Joseph Djugash[1], Sanjiv Singh[1], and Peter Corke[2]

[1] Carnegie Mellon University. 5000 Forbes Ave., Pittsburgh, PA 15213, USA.
{robojoe@cmu.edu,ssingh@ri.cmu.edu}
[2] CSIRO ICT Centre. P.O. Box 883, Kenmore, Australia 4069.
{peter.corke@csiro.au}

Summary. In this paper, we present recent results with using range from radio for mobile robot localization. In previous work we have shown how range readings from radio tags placed in the environment can be used to localize a robot. We have extended previous work to consider robustness. Specifically, we are interested in the case where range readings are very noisy and available intermittently. Also, we consider the case where the location of the radio tags is not known at all ahead of time and must be solved for simultaneously along with the position of the moving robot. We present results from a mobile robot that is equipped with GPS for ground truth, operating over several km.

Key words: SLAM, range-based localization, kalman filter, particle filter

1 Introduction

Many tasks for which robots are well suited require a high level of precision in localization for the application to be successful. One solution to the problem of localization in which environmental structure can't be relied upon for use in localization is to obtain absolute position via GPS. This approach is limited, however, to environments in which a clear line of sight to GPS satellites orbiting the earth, is available. Robots navigating inside buildings or underground cannot receive GPS data, and in outdoor environments nearby structures and even foliage can affect the quality of localization. Another common localization technique is dead reckoning, in which the robot's position is estimated based on measurements of distance travelled and orientation taken from wheel encoders and gyros. Since the dead reckoning position estimate degrades over time, a robot must correct position error using landmarks detected by onboard sensors. A problem that frequently arises in these cases is that of data association: sensed data must be associated with the correct landmark, even though multiple landmarks may have similar features. Additionally, in many settings it is not possible to guarantee line of sight to the landmarks.

The method of sensing we have been using involves low-cost, low-power, radio frequency tags placed in the environment. Originally intended as a means to track assets and people in an environment equipped with special RF transponders, we invert the paradigm by fixing the tags in the environment and moving a transponder with a robot. As the robot moves, the transponder periodically sends out a query, and any tags within range respond by sending a reply. The robot can then estimate the distance to each responding tag by determining the time lapsed between sending the query and receiving the response. The advantage of such a method is that it does not require line of sight between tags and the mobile robot, making it useful in many environmental conditions that fail optical methods. Note that, since each tag transmits a unique ID number, distance readings are automatically associated with the appropriate tags, so the data association problem is solved trivially.

We would like to send a mobile robot into an environment containing these tags and have it navigate successfully while maintaining a reliable estimate of its location at all times. In this paper, we examine issues of robustness that result from noisy and infrequent range data. We also examine the issues of SLAM in this environment by presenting results from experiments in which the robot starts moving in the environment without apriori knowledge of the location of the tags.

2 Related Work

Most landmark-based localization systems use sensors that measure relative bearing or in some cases both range and bearing to distinct features in the environment. In the case that the location of these landmarks is unknown, the problem is more difficult and is generally known as Simulataneous Localization and Mapping (SLAM). Here we report on localization results with a modality in which only range to landmarks (RF tags) is measured. Some other researchers have used range to estimate position. In most cases, instead of using range, signal strength from a known transmitter is used to produce a “pseudorange” that is then used for triangulation. For instance, the Cricket System [9] uses fixed ultrasound emitters and embedded receivers in the object being located. Radio frequency signals are used to synchronize time measurements and to reject multipath readings. The localization technique is based on triangulation relative to the beacons. The RADAR system [1] uses 802.11b wireless networking for localization. This system uses the signal strength of each packet to localize a laptop. RADAR uses nearest neighbor heuristics to achieve localization accuracy of about 3 meters. The SpotOn system [4], uses radio signal attenuation to estimate distance between tags. The system localizes wireless devices relative to one another, rather than to fixed base stations, allowing for ad-hoc localization. Note that GPS also works by triangulating ranges to multiple satellites. In some cases, GPS localization is augmented with inertial measurement and/or dead reckoning. In almost all such systems, GPS triangulation generally develops an estimate of position as well as uncertainty that is merged with a position estimate from dead reckoning. Other methods choose to train on patterns of signal strength to localize. For example, Ladd et al propose a Bayesian formulation to localize based on signal strength patterns from fixed receivers [8].

In contrast, we use a single filter to combine range measurements with dead reckoning and inertial measurements. While the range measurements are noisy and exhibit biases, we find that treatment by an extended Kalman filter (necessary because the underlying system is non-linear) after preprocessing to remove outliers and to remove systematic biases, suffices as long as the estimate doesn't get too far from the true state. This might happen if the initial condition is too far from the true state or if the filter diverges due to missing range data over an extended period of time.The Kalman filter has the advantage that the representation of the distribution is compact; a Gaussian distribution can be represented by a mean and a covariance matrix. The robot's pose estimation is maintained as a Gaussian distribution and sensor data from dead reckoning and landmark observations is fused to obtain a new position distribution.

Recent extensions of Kalman filtering allow for non-Gaussian, multimodal probability distributions through multiple hypothesis tracking. The result is a more versatile estimation technique that still preserves many of the computational advantages of the Kalman filter. Monte Carlo localization, or particle filtering, provides a method of representing multimodal distributions for position estimation [2], [11], with the advantage that the computational requirements can be scaled. The main advantage of these methods is their ability to comverge from a poor initial condition. We show how a particle filter is able to recover from large offset errors that are large enough that the Kalman Filter fails. Also, we extend previous work [5], [7] in SLAM by treating the case in which the robot starts with no information about the location of the tags in the environment.

3 Approach

Our current emphasis is robustness. We would like to explicitly treat the case of noisy and missing range data in addition to requiring the robot to discover the location of the landmarks on its own. We assume only that the robot has some information about the accuracy of the range measurements as reported previously [7]. Here we use range data that is significantly less frequent and more noisy. For example, the range measurements can have a variance of upto 6 m and range measurements can be as spread out by as much as 15 seconds.

Below we discuss the use of particle filter as a method of being able to recover from large estimation errors. While the particle filter has weaker performance than the Kalman filter when all is well, it shines when there is a significant break in the range data or when there is a large initial offset. We show the ability of the robot to locate the radio tags in the case that their locations are not known ahead of time.

4 Localization

4.1 Localization with Kalman Filter

Formulation.

We have formulated a Kalman Filter that estimates position given measurements of odometry and heading change (from a gyro), and range measurements. Odometry and gyro measurements are used in the state propagation or the prediction step, while the range measurements are incorporated in the correction step.

Process Model. If the robot state at time k is $q_k = [x_k, y_k, \theta_k, \beta_k, \eta_k]^T$, where x_k, y_k, θ_k are the robot's position and orientation and β_k, η_k are the gyro output scale-factor error and bias at time k. The dynamics of the wheeled robot used in this experiment are well-modeled by the following set of non-linear equations:

$$q_{k+1} = \begin{bmatrix} x_k + \triangle D_k \cos(\theta_k) \\ y_k + \triangle D_k \sin(\theta_k) \\ \theta_k + (1+\beta_k)\triangle\theta_k + \eta_k \\ \beta_k \\ \eta_k \end{bmatrix} + \nu_k = f(\widehat{q}_k, u_k) + \nu_k, \tag{1}$$

where ν_k is a noise vector, $\triangle D_k$ is the odometric distance traveled, and $\triangle\theta_k$ is the orientation change. These dead reckoning measurements constitute the control input vector $u_k = [\triangle D_k, \triangle\theta_k]^T$. When a new control input vector $u(k) = [\triangle D_k, \triangle\theta_k]^T$ is received, the robot's state is updated according to the process model equation. Then we apply a standard Kalman filter, [7], to propagate the covariance matrix with the extension from our previous work to incorporate the gyro bias terms within the filter.

Measurement Model. The range measurement at time k is modeled by:

$$\begin{gathered} r_k = \sqrt{(x_k - x_b)^2 + (y_k - y_b)^2} \\ y(k) = \begin{bmatrix} r_k \\ \triangle D_k \\ \triangle\theta_k \end{bmatrix} + \omega(k) = h(q_{k+1}) + \omega(k) \end{gathered} \tag{2}$$

where, r_k is the range measurement received at time k and (x_b, y_b) is the location of the beacon from which a measurement was received. When a measurement is obtained, using the measurement model, we compute the expected range r_k to the beacon. Then the state can be updated using standard Kalman filtering equations.

Results.

In order to evaluate the performance of the filter we turn to the two commonly used error metrics, the Cross-Track Error (XTE) and the Along-Track Error (ATE). The XTE accounts for the position error that is orthogonal to the robot's path (i.e. orthogonal to the true robot's orientation), while the ATE accounts for the tangential component of the position error. As part of our error analysis of the path estimates, we observe the average of the absolute values of the XTE and ATE for each point in the path, as well as the maximum and standard deviation of these errors.

In the experiment illustrated here, the true initial robot position from GPS was used as the initial estimate. Furthermore, the location of each tag was known. Figure 1 shows the estimated path using the Kalman filter, along with the GPS ground truth (with 2 cm accuracy) for comparison.

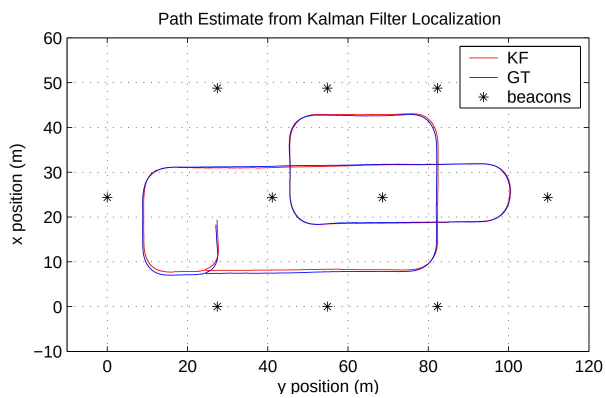

Fig. 1. The path estimate from localization (red), ground truth (blue) and beacon locations (*) are shown. The filter uses odometry and a gyro with range measurements from the RF beacons to localize itself. The path begins at (0,0) and ends at (33,0), travelling a total of 3.7 km and completing 11 identical loops, with the final loop (0.343 km) shown above. (Note the axes are flipped). Numerical results are given in Table 1.

Table 1. Cross-Track and Along-Track Errors for Kalman filter Localization estimate for the entire data set using the Kalman Filter with gyro bias compensation.

	XTE	**ATE**
Mean Abs.	0.3439 m	0.3309 m
Max.	1.7634 m	1.7350 m
Std. Dev.n	0.3032 m	0.2893 m

Failure.

Sensor Silence. An issue that requires attention while dealing with the Kalman filter is that of *extensive sensor silence*. When the system encounters a long period during which no range measurements are received from the beacons, it becomes heavily dependant on the odometry and its estimate diverges. Upon recovering from this period of sensor silence, the Kalman filter is misled into settling at a diverged solution. The Figure 2 shows the failure state of the Kalman filter when presented with a period of sensor silence. In this experiment, all range measurements received prior to a certain time were ignored so that the position estimate is derived through odometry alone. As can be seen in the figure, when the range data starts once again, the Kalman filter fails to converge to an accurate estimate.

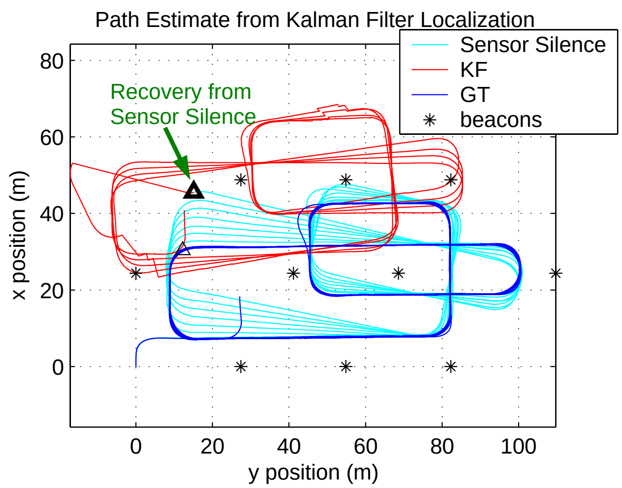

Fig. 2. The path estimate during the extended period of "simulated" sensor silence (cyan), Kalman filter's recovery from the diverged solution (red), ground truth (blue) and beacon locations are shown. (Note the axes are flipped). The filter is not able to properly recover from the diverged solution resultant of the initial period of sensor silence.

Although this is characteristic of all Kalman filters in general, this problem is especially critical while dealing with range-only sensors. Due to the extra level of ambiguity associated with each range measurement it becomes far easier for the estimate to converge at an incorrect solution.

4.2 Localization with Particle Filter

As we see above, the Kalman filter can fail when the assumptions of linearity can not be justified. In this case, it is useful to look at methods like Particle

Filters that can converge without an initial estimate. Particle Filters are a way of implementing multiple hypothesis tracking. Initially, the system starts with a uniform distribution of particles which then cluster based on measurements. As with the Kalman filter, we use the dead reckoning as a means of prediction (by drifting all particles by the amount measured by the odometry and gyro before a diffusion step to account for increased uncertainty). Correction comes from resampling based on probability associated with each particle. Position estimates are obtained from the centroid of the particle positions.

Formulation

The particle filter evaluated in this work estimates only position on the plane, not vehicle orientation. Each "particle" is a point in the state space (in this case the x, y plane) and represents a particular solution. The particle resampling method used is as described by Isard and Blake [3]. Drift is applied to all particles based on the displacement estimated by dead reckoning from the state at the previous measurment. Diffusion is achieved by applying a Gaussian distibuted displacement with a standard deviation of B m/s which scales according to intersample interval. Given a range measurement r from the beacon at location $X_b = (x_b, y_b)$ the probability for the i'th particle is

$$P(r, X_b, X_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(r-|X_b - X_i|)}{2\sigma^2}} + P_0 \tag{3}$$

which has a maximum in a circle of radius r about the beacon with a radial cross-section that is Gaussian. The minimum probability, P_0, helps reduce problems with particle extinction. σ is related to the variance in the received range measurements.

It was found to be important to gate range measurements through a normalized error and a range measurement band, [7]. In the event of a measurement outside the range gate an open-loop update is performed, the particles are displaced by the dead reckoning displacement without resampling or diffusion.

The location of the vehicle is taken as the probability weighted mean of all particles. There is no attempt made to cluster the particles so if there are, for example, two distinct particle clusters the mean would lie between them. Initially this estimate has a significantly different value to the vehicle's position but converges rapidly. Here we use 1000 particles, $\sigma = 0.37$, and $B = 0.03$.

Results

In the experiment illustrated here, the initial condition for the particles is based on **no** prior information, the particles are distributed uniformly over a large bounding rectangle that encloses all the beacons. The location of each tag was known apriori. Figure 3(a) contains the plot of the particle filter estimated path, along with the GPS ground truth.

It should be noted that the particle filter is a stochastic estimation tool and results vary from run to run using the same data. However it is consistently reliable in estimating the vehicle's location with no prior information.

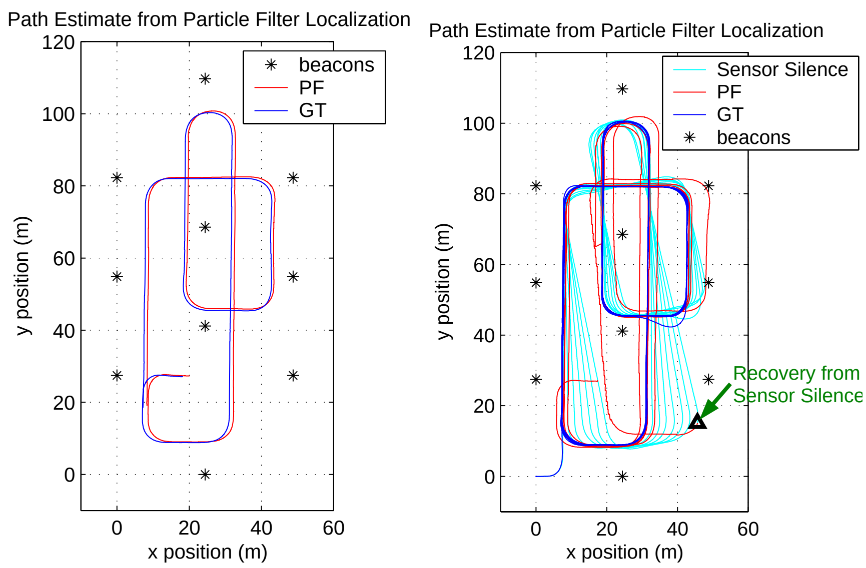

Fig. 3. (a) The path estimate from localization using a Particle Filter (red), ground truth (blue) and beacon locations (*) are shown. The filter uses the odometry and a gyro with absolute measurements from the RF beacons to produce this path estimate. The Particle Filter is not given any information regarding the initial location of the robot, hence it begins its estimate with a particle cloud uniformly distributed-with a mean at (-3.6 m, -2.5 m). The final loop (0.343 km) of the data set is shown here, where the Particle Filter converges to a solution. Numerical results are given in Table 2. (b) The path estimate during the extended period of "simulated" sensor silence (cyan), Particle filter's recovery from the diverged solution (red), ground truth (blue) and beacon locations are shown. The filter easily recovers from the diverged solution, exhibiting the true nature of the particle filter.

The next experiment addresses the problem of extensive sensor silence discussed in Section 4.1. When the Particle filter is presented with the same scenario that was given to the Kalman filter earlier we acquire the Figure 3(b). This figure reveals the ability of the Particle filter to recover from an initially diverged estimate. It can be observed that although in most cases the particle filter produces a locally non-stable solution (due to resampling of the

Table 2. Cross-Track and Along-Track Errors for Particle filter Localization estimate for the entire data set.

	XTE	**ATE**
Mean Abs.	0.4053 m	0.3623 m
Max.	1.6178 m	1.8096 m
Std. Dev.	0.2936 m	0.2908 m

particles), its ablity to recover from a diverged solution makes it an effective localization algorithm.

5 SLAM - Simultaneous Localization and Mapping

Here we deal with the case where the location of the radio tags is not known ahead of time. We consider an online (Kalman Filter) formulation that estimates the tag locations at the same time as estimating the robot position.

5.1 Formulation of Kalman Filter SLAM

The Kalman filter approach described in Section 4.1 can be reformulated for the SLAM problem.

Process Model: In order to extend the formulation from the localization case to perform SLAM, we need only to include position estimates of each beacon in the state vector. So,

$$q_k = \left[x_k \; y_k \; \theta_k \; x_{b1} \; y_{b1} \; ... \; x_{bn} \; y_{bn}\right]^T \tag{4}$$

where n is the number of initialized RF beacons at time k. The process used to initialize the beacons is described later in this section.

Measurement Model: To perform SLAM with a range measurement beacon b, located at (x_b, y_b), we modify the Jacobian H(k) (the measurement matrix) to include partials corresponding to each beacon within the current state vector. So,

$$H(k) = \frac{\partial h}{\partial q_k}|_{q=\hat{q}}|_{q=\hat{q}} = \left[\frac{\partial h}{\partial x_k} \; \frac{\partial h}{\partial y_k} \; \frac{\partial h}{\partial \theta_k} \; \frac{\partial h}{\partial x_{t1}} \; \frac{\partial h}{\partial y_{t1}} \; \cdots \; \frac{\partial h}{\partial x_b} \; \frac{\partial h}{\partial y_b}, \cdots \; \frac{\partial h}{\partial x_{tn}} \; \frac{\partial h}{\partial y_{tn}}\right] \tag{5}$$

where,

$$\begin{aligned} \frac{\partial h}{\partial x_{ti}} = \frac{\partial h}{\partial y_{ti}} &= 0, for \;\; ti \neq b, and 1 \leq i \leq n. \\ \frac{\partial h}{\partial x_b} &= \frac{-(x_k - x_b)}{\sqrt{(x_k - x_b)^2 + (y_k - y_b)^2}} \\ \frac{\partial h}{\partial x_b} &= \frac{-(y_k - y_b)}{\sqrt{(x_k - x_b)^2 + (y_k - y_b)^2}} \end{aligned} \tag{6}$$

Only the terms in H(k) directly related to the current range measurement (i.e., the partials with respect to the robot pose and the position of the beacon giving the current measurement) are non-zero. To complete the SLAM fomulation, P (the covariance matrix) is expanded to the correct dimentionality (i.e., 2n+3 square) when each new beacon is initialized.

Beacon Initialization: For perfect measurements, determining position from range information is a matter of simple geometry. Unfortunately, perfect measurements are difficult to achieve in the real world. The measurements are contaminated by noise, and three range measurements rarely intersect exactly. Furthermore, estimating the beacon location while estimating the robot's location introduces the further uncertainty associated with the robot location.

The approach that we employ, similar to the method proposed by Olson *et al* [10], considers pairs of measurements. A pair of measurements is not sufficient to constrain a beacon's location to a point, since each pair can provide up to two possible solutions. Each measurement pair "votes" for its two solutions (and all its neighbors) within a two dimensional probability grid to provide estimates of the beacon location. Ideally, solutions that are near each other in the world, share the same cell within the grid. In order to accomplish this requirement, the grid size is chosen such that it matches the total uncertainty in the solution: range measurement uncertainty plus Kalman filter estimate uncertainty. After all the votes have been cast, the cell with the greatest number of votes contains (with high probability) the true beacon location.

5.2 Results from Kalman Filter SLAM

In this experiment, the true initial robot position from GPS was used as an initial estimate. There was also no initial information, about the beacons, provided to the Kalman filter. Each beacon is initialized in an online method, as described in Section 5.1. Performing SLAM with Kalman filter produces a solution that is globally misaligned, primarily due to the dead reckoning that had accumulated prior to the initilization of a few beacons. Since, until the robot localizes a few beacons, it must rely on dead reckoning alone for navigation. Although this might cause the Kalman filter estimate to settle into an incorrect global solution, the relative structure of the path is still maintained.

In order to properly evaluate the performance of SLAM with Kalman filter, we must study the errors associated with the estimated path, after removing any global translational/rotation offsets that had accumulated prior to the initialization of a few beacons. Figure 4 shows the final 10% of the Kalman filter path estimate after a simple affine transform is performed based on the final positions of the beacons and their true positions. The plot also includes the corresponding ground truth path, affine transformed versions of the final beacon positions and the true beacon locations. Table 3 provides the XTE and ATE for the path shown in Figure 4.

Several experiments were performed, in order to study the convergence rate of SLAM with Kalman filter. The plot in Figure 5 displays the XTE and its 1 sigma bounds for varying amounts of the data used to perform SLAM (i.e., it shows the result of performing Localization after performing SLAM on different amounts of the data to initialize the beacons).

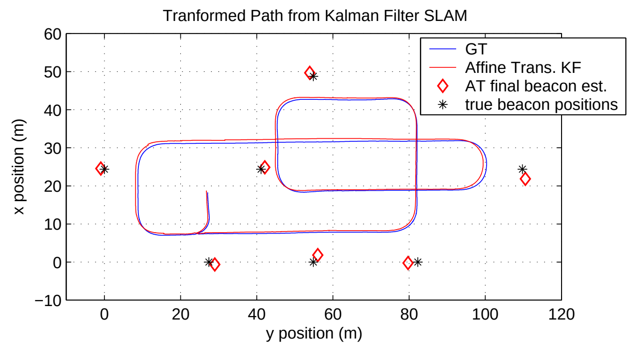

Fig. 4. The path estimate from SLAM using a Kalman Filter (green), the corresponding ground truth (blue), true beacon locations (black *) and Kalman Filter estimated beacon locations (green dimond) are shown. (Note the axes are flipped). A simple affine transform is performed on the final estimate beacon locations from the Kalman Filter in order to re-align the misaligned global solution. The path shown corresponds to the final loop (0.343 km) of the full data set after the affine transform. Numerical results are given in Table 3.

Table 3. Cross-Track and Along-Track Errors for the final loop (0.343 km) of the Data Set after the Affine Transform.

	XTE	**ATE**
Mean Abs.	0.5564 m	0.6342 m
Max.	1.3160 m	1.3841 m
Std. Dev.	0.3010 m	0.2908 m

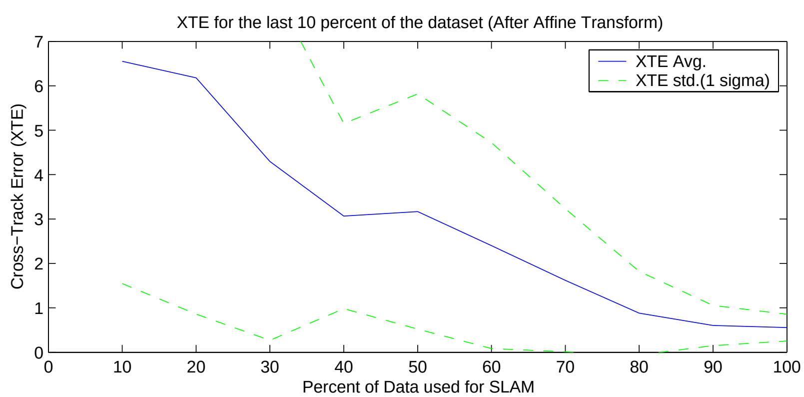

Fig. 5. Kalman Filter Covergence Graph. Varying amount of data is used to perform SLAM, after which the locations of the initialized beacons are fixed and simple Kalman filter localization is perform on the remaining data. The plot above shows the average absolute XTE and its 1 sigma bounds for various subsets of the data used for SLAM.

6 Summary

This paper has reported on extensions for increasing robustness in localization using range from radio. We have examined the use of a particle filter for recovering from large offsets in position that are possible in case of missing or highly noisy data from radio beacons. We have also examined the case of estimating the locations of the beacons when their location is not known ahead of time. Since practical use would dictate a first stage in which the locations of the beacons are mapped and then a second stage in which these locations are used, we have presented an online method to locate the beacons. The tags are localized well enough so that the localization error is equal to the error in the case where the tag locations are known exactly in advance.

References

1. P. Bahl and V. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *In Proc. of the IEEE Infocom 2000*, Tel Aviv, Israel, 2000.
2. D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localizatoin: Efficient position estimation for mobile robots. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1999.
3. M. Isard and A. Blake. Condensationconditional density propagation for visual tracking. In *International Journal of Computer Vision*, 1998.
4. R. Want J. Hightower and G. Borriello. Spoton: An indoor 3d location sensing technology based on rf signal strength. Technical report.
5. G. Kantor and S. Singh. Preliminary results in range-only localization and mapping. In *Proceedings of IEEE Conference on Robotics and Automation*, Washington D.C., USA, May 2002.
6. D. Kurth. Range-only robot localization and slam with radio. Master's thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2004. tech. report CMU-RI-TR-04-29.
7. D. Kurth, G. Kantor, and S. Singh. Experimental results in range-only localization with radio. In *Proceedings of IROS2003*, Las Vegas, USA, October 2003.
8. A.M. Ladd, K.E. Bekris, G. Marceau, A. Rudys, D.S. Wallach, and L.E. Kavraki. Robotics-based location sensing for wireless ethernet. In *Eighth ACM MobiCom*, Atlanta, GA, September 2002.
9. A. Chakraborty N. Priyantha and H. Balakrishman. The cricket location support system. In *In Proc. of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2000)*, Boston, MA, August 2000.
10. Edwin Olson, John Leonard, and Seth Teller. Robust range-only beacon localization. In *Proceedings of Autonomous Underwater Vehicles, 2004*, 2004.
11. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 101:99–141, 2000.