

2D Simultaneous Localization and Mapping using an Enhanced Kalman Filter Estimator

Rohaan Ahmed

Masters in Engineering candidate at the University of Toronto Institute for Aerospace Studies

University of Toronto, Ontario, Canada. E-mail rohaan.ahmed@utoronto.ca

(December 31, 2011)

Abstract—A common desired trait in mobile robotics is the ability of a robotic vehicle to determine its own location as well as map an unknown environment using noisy sensors in real-time. Robotics engineers for many years have sought a solution to this problem, and one technique that has found widespread usage is known as the Extended Kalman Filter based Simultaneous Localization and Mapping algorithm (EKF-SLAM). This algorithm uses the non-linear Extended Kalman Filter to fuse data from interoceptive and exteroceptive sensors in order to estimate the vehicles position and orientation within its surroundings, while simultaneously mapping the previously unknown environment.

In this paper, the EKF-SLAM algorithm is demonstrated using a two-dimensional experiment in which a robotic vehicle, equipped with wheel odometers and a Hokuyo URG-04LX laser-rangefinder, drives within a forest of vertical tubes at a University of Toronto Institute for Aerospace Studies indoor robotics facility. The paper is intended to provide a beginners understanding of the EKF-SLAM algorithm, and assumes no prior understanding of the SLAM problem.

Index Terms—Simultaneous Localization and Mapping, Extended Kalman Filter, Mobile Robotics

I. INTRODUCTION TO SLAM

THE term SLAM stands for Simultaneous Localization and Mapping, and refers to the task of simultaneously locating a vehicle and logging information about its environment in mobile robotics. The objective is for the vehicle to recursively estimate its own location within its environment using noisy sensors and previously-detected environmental features, and then update the list of detected features with new information for use in future estimation. It represents a commonly encountered and much studied problem in the field of mobile robotics, and has seen application in planetary robotics, aerospace, underwater systems and so on [1].

This paper will demonstrate the utility and functionality of SLAM by applying it to an indoor-robotics experiment. This will be done using one of the most commonly used SLAM techniques, referred to as EKF based SLAM (EKF-SLAM), which uses the Extended Kalman Filter estimation algorithm to predict, estimate and update the state of the system.

II. EXPERIMENTAL SET-UP

The experiment designed to demonstrate SLAM is made up of three systems; a robotic vehicle equipped with interoceptive



Fig. 1. The Robot Vehicle and Landmarks at the University of Toronto Institute for Aerospace Studies

and exteroceptive sensors, multiple vertical tubes marked with reflective markers to serve as landmarks, and a ten-camera motion capture system to serve as the ground-truth data capturing system [2].

A. The Robotic Vehicle

The vehicle is a four-wheel remote-controlled robot equipped with an exteroceptive sensor package capable of providing range and bearing information to landmarks within its range of detection, and a set of interoceptive sensors used to provide the vehicles speed and angle through dead-reckoning. The vehicle is marked with a reflective marker.

Optical encoder based odometry sensors logged at 10Hz serve as the interoceptive sensors, whereas a Hokuyo URG-04LX laser rangefinder logged at 10Hz serves as the exteroceptive sensor. The rangefinder has a maximum range of 5m and a field-of-view of 240 degrees centered straight ahead of the vehicle. Both these sensors contain noise, which is exactly what makes an estimation algorithm, such as the EKF, necessary in the implementation of SLAM. This sensor noise will be studied later.

B. The Landmarks

Seventeen static vertical tubes marked with reflective markers serve as the landmarks in this experiment. These are the environmental features that the robot will be required to map and use to locate itself. Fig (1) shows the distribution of the landmarks and the robotic vehicle that will drive through it.

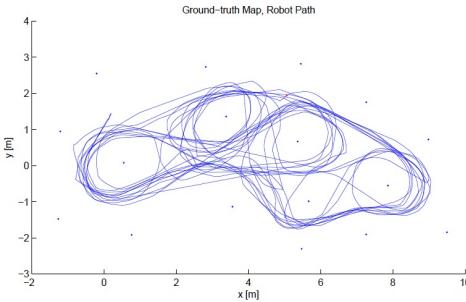


Fig. 2. Grond-truth robot and landmark positions

C. Ground-Truth Data

All experiments were conducted at the University of Toronto Institute for Aerospace Studies Vicon motion capture lab. A ten-camera motion captures system installed within this lab serves as the source for ground-truth data. The system uses reflective markers placed on landmarks and the robotic vehicle to capture their position at a rate of 40Hz. Data recorded using this system is considered to be highly accurate when compared with the expected estimation results, and thus will serve as the ground-truth information benchmark for comparison with the estimated map.

D. The Experimental Run

The robotic vehicle was driven within a forest of seventeen vertical tubular landmarks for 20 minutes, and data from the exteroceptive, interoceptive and ground-truth sensors was logged. Fig (2) displays the robotic vehicles path through the tubes as determined by ground-truth sensors. This map will serve as the benchmark for comparison with results of the EKF-SLAM.

E. Experimental Data Processing

Since the objective of this experiment is to demonstrate the EKF-SLAM algorithm and not data-acquisition or data-association, data logged from the sensors was cleaned-up to make subsequent calculations and implementation of EKF-SLAM cleaner. Firstly, asynchronous data from the three sensors was cleaned-up by interpolating the values at constant time intervals of 0.1s, thus allowing us to assume that all data arrives synchronously at 10Hz from each sensor. Secondly, the data from each of the wheel encoders is pre-calculated to provide speed and angle readings, thus eliminating the need for manually calculating this information at each time-step. Thirdly, laser-rangefinder information at each time-step was pre-processed to provide range and bearing data, eliminating the need for landmark-extraction at each time-step.

It is important to note that any real-world on-line application of SLAM will have to deal with these issues in real-time, however, by logging the data beforehand, we have afforded ourselves the luxury of pre-cleaning and pre-processing sensor data to compute the EKF-SLAM. It is vitally important to note, however, that sensor noise was not affected by any of these cleaning steps, as that would render the experiment void.

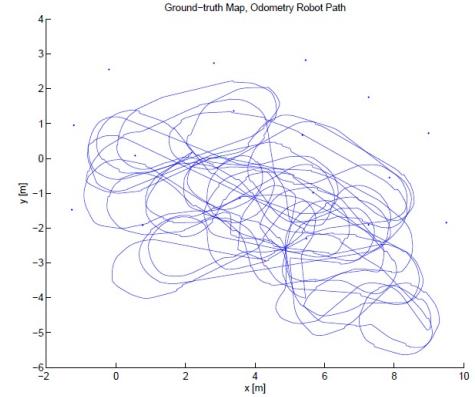


Fig. 3. Robot position calculated using only odometry data

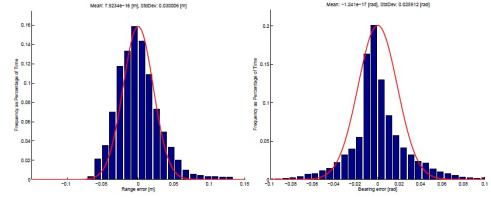


Fig. 4. Laser Rangefinder Error Graphs

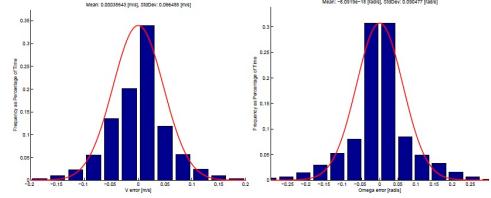


Fig. 5. Odometry Error Graphs

III. ALGORITHM SELECTION

Thus far in this paper, we have assumed that we will be using the Extended Kalman Filter to implement our SLAM algorithm without much discussion of the reasons or alternatives. In this section, we will try to show why SLAM and, in particular, EKF-SLAM, is a feasible approach in this particular experiment.

A. Why Simultaneous Localization and Mapping?

Fig (3) shows a map of the robots position calculated using only the interoceptive odometry data, with the landmarks fixed to their ground-truth position. Upon comparison with Fig (2), it is obvious that odometry data alone is not an accurate enough representation of the actual position of the robot, thus a more accurate way of locating the robot is required. Furthermore, Fig (4) and Fig (5) show plots of the errors of the laser-rangefinder and odometry measurements [2]. It can be seen that there is considerable noise in these readings. If we were to rely solely on the interoceptive sensor for location information, and then use the exteroceptive measurements only for mapping purposes, it is easy to see that our map would be highly inaccurate.

The Simultaneous Localization and Mapping approach is designed to solve this very problem. In SLAM, the robot is not reliant only on the odometry data for localization, but rather uses previously observed landmarks to correct its location error. In other words, it produces a more accurate estimate of the systems state by combining exteroceptive sensor information with interoceptive sensor information [3].

However, we have already seen that both sensors contain noise, and thus, readings from these sensors cannot be taken as 100% accurate. We must, therefore, use an estimator.

B. Why Extended Kalman Filter?

Revisiting Fig (4) and Fig (5), it can be seen that both sensors have noise that must be accounted for when calculating our system state. This points to the fact that an estimation algorithm will be required in order to compensate for both interoceptive and exteroceptive sensor noise. There are several well-established algorithms that may be applied to a problem for estimation purposes, but the selection of a particular algorithm can only be done through a careful examination of the system and the sensors.

Fig (4) and Fig (5) show that all four sensor parameters; range, bearing, speed, and orientation, have Gaussian-noise distribution curves, i.e., the noise is unimodal and has close to zero-mean. Furthermore, the system is non-linear due to the rotational movement of the robotic vehicle. Lastly, the system satisfies the Markov property, i.e., the future state of the system depends only on the current state of the system and the control inputs introduced into the system. Thus, any estimator used for this problem must be able to deal with non-linear systems, and may make use of its Gaussian noise and Markov characteristics [4].

The Extended Kalman Filter is an extension of the Kalman Filter estimation technique modified to work for non-linear systems through linearization about the mean and covariance of the system. For the EKF to be applicable, a system must satisfy the Markov property, and it should have a reasonably Gaussian noise distribution [4]. Hence, its clear that the EKF, at least theoretically, is a valid approach for estimating the state of this system.

Upon a review of literature, it is quickly seen that EKF-based-SLAM is, in fact, a common technique used in mobile robotics applications, and thus, should work for the given experiment [5] [6].

Note that there are other established estimation algorithms that have also been used to implement SLAM, such as the Rao-Blackwellized FastSLAM. The EKF-SLAM has been selected here due to its clarity, widespread usage, and ease of implementation [7] [3].

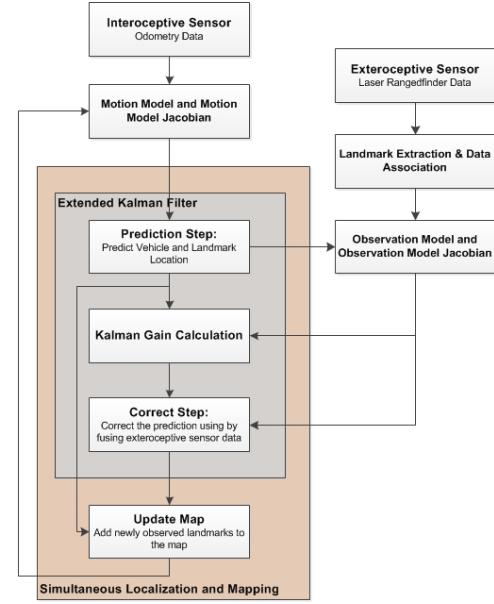


Fig. 6. Extended Kalman Filter based Simultaneous Localization and Mapping Control Flow

IV. EXTENDED KALMAN FILTER BASED SIMULTANEOUS LOCALIZATION AND MAPPING

The Extended Kalman Filter is a linear recursive algorithm that utilizes the analytical motion and observation models of the system to first predict the vehicles location, then correct the location using previously mapped data, and then update the environment map. The flowchart in Fig (6) displays the multiple steps required in the implementation of EKF-SLAM.

Note: In all the following equations, matrix dimensions are provided in terms of n and m . Here, n represent the number of landmarks detected by the vehicle *up-to* time-step k , and m represents the number of landmarks visible to the laser rangefinder *at* time-step k .

A. System Motion and Observation Models

The motion model of the vehicle is given in equation (1), where (x_k, y_k, θ_k) is the robots position and orientation, (v_k, ω_k) is the robots translational/rotational speed (derived from odometers), \mathbf{w}_k is the process noise, and T is the sampling period [2].

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}_{[3 \times 1]} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + T \underbrace{\begin{bmatrix} \cos\theta_{k-1} & 0 \\ \sin\theta_{k-1} & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{h}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, \mathbf{0})} \left(\begin{bmatrix} v_k \\ \omega_k \end{bmatrix} + \mathbf{w}_k \right) \quad (1)$$

The motion model of each landmark, l , is given in equation (2), where (x_l, y_l) is the landmarks position at time-step k . Note that this model assumes that the landmarks are static, and thus, the state of each landmark at time-step k is the same as its state at time-step $k - 1$.

$$\mathbf{x}_{l,k} = \begin{bmatrix} x_{l,k} \\ y_{l,k} \end{bmatrix}_{[2 \times 1]} = \begin{bmatrix} x_{l,k-1} \\ y_{l,k-1} \end{bmatrix} \quad (2)$$

Combining the two motion models, we get the system state update matrix, or the system motion model, as given by (3).

$$\mathbf{S} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ x_{l1,k} \\ y_{l1,k} \\ \vdots \\ x_{ln,k} \\ y_{ln,k} \end{bmatrix}_{[(3+2n) \times 1]} = \begin{bmatrix} \mathbf{h}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, \mathbf{0})_{3 \times 1} \\ x_{l1,(k-1)} \\ y_{l1,(k-1)} \\ \vdots \\ x_{ln,(k-1)} \\ y_{ln,(k-1)} \end{bmatrix} \quad (3)$$

The observation model from the robots position to landmark, l , at time-step k is given by (4), where $(r_{l,k}, \phi_{l,k})$ is the range/bearing to landmark l (derived from the laser rangefinder), (x_l, y_l) is the position of the center of landmark, and $\mathbf{n}_{l,k}$ is the exteroceptive sensor noise. The distance between the robots center and the laser rangefinder is d (the laser is at the front of the robot). Note that both the motion and observation models are nonlinear [2].

$$\begin{bmatrix} r_{k,l} \\ \phi_{k,l} \end{bmatrix}_{[2 \times 1]} = \underbrace{\begin{bmatrix} \sqrt{(x_l - x_k - d\cos\theta_k)^2 + (y_l - y_k - d\sin\theta_k)^2} \\ \text{atan2}((y_l - y_k - d\sin\theta_k), (x_l - x_k - d\cos\theta_k)) \end{bmatrix}}_{\mathbf{O}_l(\mathbf{x}_k, \mathbf{n}_k, l)} + \mathbf{n}_{l,k} \quad (4)$$

B. System Motion and Observation Model Jacobians

The EKF algorithm linearizes the system by taking the Jacobian of the motion and observation models evaluated at the predicted and previously estimated values [4].

Equation (5) represents the system update matrix differentiated with respect to the state, \mathbf{x} , and (6) represents the system update matrix differentiated with respect to the noise interoceptive, \mathbf{w}_k .

$$\mathbf{JS}_{\mathbf{x},k} = \begin{bmatrix} 1 & 0 & -Tv_k \sin(\hat{\theta}_{k-1}) & 0 & \dots & 0 \\ 0 & 1 & Tv_k \cos(\hat{\theta}_{k-1}) & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \mathbf{1}_{2 \times 2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathbf{1}_{2 \times 2} \end{bmatrix}_{[(3+2n) \times (3+2n)]} \quad (5)$$

$$\mathbf{JS}_{\mathbf{w},k} = \begin{bmatrix} T \cos(\hat{\theta}_{k-1}) & 0 \\ T \sin(\hat{\theta}_{k-1}) & 0 \\ 0 & T \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}_{[(3+2n) \times 2]} \quad (6)$$

The observation model differentiated with respect to the exteroceptive noise, $\mathbf{n}_{k,l}$, is given by equation (7).

$$\mathbf{JO}_{\mathbf{n},k} = \mathbf{1}_{[2m \times 2m]} \quad (7)$$

The observation model differentiated with respect to the state is given by equation (8),

$$\mathbf{JO}_{\mathbf{x},k} = \begin{bmatrix} [\mathbf{G}_{\mathbf{x},k}]_{2m \times 3} & \dots & \underbrace{[\mathbf{G}_{\mathbf{x}l,k}]_{2m \times 2}}_{\forall \text{ visible landmarks}} & \dots & 0 \end{bmatrix}_{[2m \times (3+2n)]} \quad (8)$$

Where the matrices $\mathbf{G}_{\mathbf{x},k}$ and $\mathbf{G}_{\mathbf{x}l,k}$, for each landmark, l , are given by equations (9) and (10), respectively.

$$\begin{aligned} \mathbf{G}_{\mathbf{x},k} &= \begin{bmatrix} g_{k,11} & g_{k,12} & g_{k,13} \\ g_{k,21} & g_{k,22} & g_{k,23} \\ -a & & \end{bmatrix}_{[2 \times 3]} \\ g_{k,11} &= \frac{-a}{\sqrt{(-a)^2 + (-b)^2}} \\ g_{k,12} &= \frac{-b}{\sqrt{(-a)^2 + (-b)^2}} \\ g_{k,13} &= \frac{2(ads\sin\theta_k - bdc\cos\theta_k)}{\sqrt{(-a)^2 + (-b)^2}} \\ g_{k,21} &= \frac{b \cdot \partial a_x}{1 + abs} \\ g_{k,22} &= \frac{\partial b_y}{a(1 + abs)} \\ g_{k,23} &= \frac{\partial b_\theta}{\partial a_\theta(1 + abs)} - 1 \end{aligned} \quad (9)$$

$$\begin{aligned} \mathbf{G}_{\mathbf{x}l,k} &= \begin{bmatrix} g_{l,11} & g_{l,12} \\ g_{l,11} & g_{l,22} \end{bmatrix}_{[2 \times 2]} \\ g_{l,11} &= \frac{a}{\sqrt{(-a)^2 + (-b)^2}} \\ g_{l,12} &= \frac{b}{\sqrt{(-a)^2 + (-b)^2}} \\ g_{l,21} &= \frac{-b \cdot \partial a_x}{1 + abs} \\ g_{l,22} &= \frac{-\partial b_y}{a(1 + abs)} \end{aligned} \quad (10)$$

Where,

$$\begin{aligned}
a &= x_l - x_k - d \cos \theta_k \\
b &= y_l - y_k - d \sin \theta_k \\
\partial a_x &= \frac{1}{(-a)^2} \\
\partial b_y &= -1 \\
\partial a_\theta &= d \sin \theta_k \\
\partial b_\theta &= -d \cos \theta_k \\
abs &= \frac{b^2}{a^2}
\end{aligned}$$

The noise covariance matrices, \mathbf{R}_k and \mathbf{Q}_k , are given by equations (11) and (12), respectively,

$$\mathbf{R}_k = \begin{bmatrix} \sigma_r^2 & 0 & 0 & 0 & \dots \\ 0 & \sigma_b^2 & 0 & 0 & \dots \\ 0 & 0 & \sigma_r^2 & 0 & \dots \\ 0 & 0 & 0 & \sigma_b^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}_{[2m \times 2m]} \quad (11)$$

$$\mathbf{Q}_k = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix}_{[2 \times 2]} \quad (12)$$

Where σ_r , σ_b , σ_ω and σ_v are pre-calculated sensor noise standard deviations, extracted from Fig (4) and Fig (5).

The *innovation* within an EKF is the difference between the calculated observation model and the actual range and bearing readings received from the sensors. \mathbf{y}_k is a column vector of actual laser rangefinder readings given by equation (13), and \mathbf{O}_k is a column vector of the calculated observation model for all landmarks visible at time-step k given by (14).

$$\mathbf{y}_k = \begin{bmatrix} r_1 \\ b_1 \\ r_2 \\ b_2 \\ \vdots \\ r_n \\ b_n \end{bmatrix}_{[2m \times 1]} \quad (13)$$

$$\mathbf{O}_k(\hat{\mathbf{x}}^-, \mathbf{0}) = \begin{bmatrix} \mathbf{O}_1(\hat{\mathbf{x}}^-, \mathbf{0}) \\ \mathbf{O}_2(\hat{\mathbf{x}}^-, \mathbf{0}) \\ \vdots \\ \mathbf{O}_n(\hat{\mathbf{x}}^-, \mathbf{0}) \end{bmatrix}_{[2m \times 1]} \quad (14)$$

C. Applying the Extended Kalman Filter

The first step in EKF is to calculate the predicted state of the system, as well as the predicted covariance matrix, using only the previously-known state from time-step ($k - 1$), the systems motion model, the interoceptive sensor noise covariance, and new interoceptive measurements received

at time-step k . Equations (15) and (16) show the two calculations involved in this step.

Prediction Stage

$$\hat{\mathbf{P}}_{[(3+2n) \times (3+2n)]}^- = \mathbf{J}\mathbf{S}_{x,k}\hat{\mathbf{P}}_{k-1}\mathbf{J}\mathbf{S}_{x,k}^T + \mathbf{J}\mathbf{S}_{w,k}\hat{\mathbf{Q}}_k\mathbf{J}\mathbf{S}_{w,k}^T \quad (15)$$

$$\hat{\mathbf{x}}_{k-1}^-_{[(3+2n) \times 1]} = \mathbf{S} \quad (16)$$

The second step in EKF involves the calculation of the Kalman Gain using the observation model for each visible landmark, the exteroceptive sensor noise covariance, and the predicted state and covariance matrices. Equation (17) displays the calculation of the Kalman Gain.

Kalman Gain Calculation

$$\mathbf{K}_{[(3+2n) \times 2m]} = \hat{\mathbf{P}}^- \mathbf{J}\mathbf{O}_{x,k}^T (\mathbf{J}\mathbf{O}_{x,k}\hat{\mathbf{P}}_{k-1}\mathbf{J}\mathbf{O}_{x,k}^T + \mathbf{J}\mathbf{O}_{n,k}\hat{\mathbf{R}}_k\mathbf{J}\mathbf{O}_{n,k}^T)^{-1} \quad (17)$$

The final step in the EKF is to calculate the corrected state and covariance matrices using the *innovation*, the predicted state and covariance, the Kalman Gain and the observation model for each landmark. Equations (18) and (19) show the calculations required in this step.

Correction Stage

$$\hat{\mathbf{P}}_k_{[(3+2n) \times (3+2n)]} = (\mathbf{I} - \mathbf{K}_k \mathbf{J}\mathbf{O}_{x,k}) \hat{\mathbf{P}}^- \quad (18)$$

$$\hat{\mathbf{x}}_k_{[(3+2n) \times 1]} = \hat{\mathbf{x}}^- + \mathbf{K}(\mathbf{y}_k - \mathbf{O}(\hat{\mathbf{x}}^-, \mathbf{0})) \quad (19)$$

D. Map Update

Once the location of the vehicle and all previously detected landmarks has been updated using the EKF, newly observed landmarks must be added to the map. This is done using the motion model for each new landmark to calculate its position (x_l, y_l), and updating the system state update matrix, \mathbf{S} , with this new information.

V. IMPLEMENTATION AND RESULTS

The data logged during the experiment was input into a MATLAB based program implementing an on-line EKF-SLAM algorithm. Various variations of the algorithm were then run on all the data-points collected during the 20 minute experiment for a total of 12,609 time-steps.

Since the EKF is a recursive algorithm, it requires an initial estimate of the robots position and orientation to serve as the base-case. For each of the following experiments, the base case was set to the robots true position and orientation. This is a valid assumption because, in most practical applications, the initial position and orientation of the vehicle is known. It is important to note, however, that the base-case plays a very important role in the convergence of the EKF algorithm, and must always be close to the true original state [4].

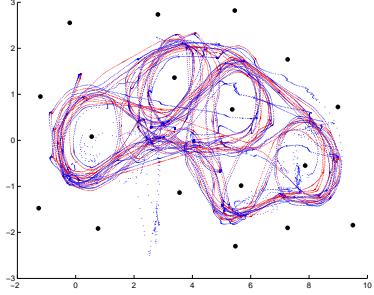


Fig. 7. EKF localization: true position vs estimated position. In this figure, the red dots represent the robots true position at each time-step, whereas the blue dots represent the estimated state. The black dots represent the true location of the static landmarks.

A. Localization Experiment

Initially, the algorithm was run to solve only for the position and orientation of the vehicle at each time-step by fixing all the landmarks to their true positions. This represents a classic Localization problem within a known environment. The results of this experiment are shown in Fig (7).

In this figure, by comparing the red and blue dots, it can be seen that despite the sensor noise, the localization algorithm performs remarkably better than relying purely on odometry data, as in Fig (3). It can also be seen that inaccuracies are greatest in areas where the density of landmarks is low. In these areas, the vehicle sees very few landmarks, and thus, is highly reliant on the highly-noisy odometry data for its localization. From this, it can be seen that one should expect that the SLAM algorithm to also struggle under conditions where landmark density is low.

B. EKF-SLAM Experiment

Next, the actual EKF-SLAM experiment was run. Fig (8) shows the final map acquired at the end of the complete EKF-SLAM algorithm, i.e., after time-step $k = 12,609$.

Figure (9) shows the difference between the estimated and true robot positions for all time-steps.

Comparing (9) with Fig (7), it is quite clear that the localization effort in the case of EKF-SLAM is not as accurate as when the landmark positions are known exactly, as should be expected. However, EKF-SLAM still does a much better job of localizing the robot when compared with

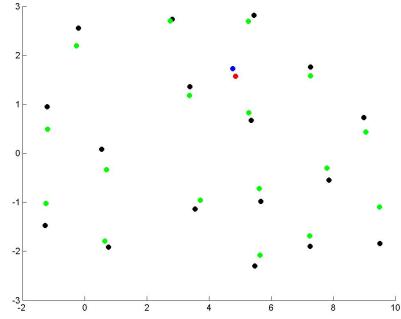


Fig. 8. EKF-SLAM: Final Map. In this figure, the red dot represents the true robot position, the blue dot represents the estimated robot position, the black dots represent true landmark positions and the green dots represent estimated landmark positions

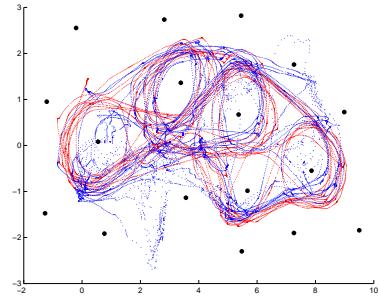


Fig. 9. EKF-SLAM: true position vs estimated position. In this figure, the red dots represent the true robot position at each time-step, whereas the blue dots represent the estimated state. The black dots represent the location of the landmarks.

only the odometry data, as in Fig (3).

In addition to the state of the system, the EKF-SLAM algorithm also updates the covariance matrix of the system at each time-step. The diagonal of the covariance matrix represents the uncertainty of an estimate, which can be displayed on a map in the form of uncertainty envelopes or uncertainty ellipses. Fig (10) shows the system map at time-step $k = 921$ with the presence of high uncertainty around a few landmarks. As the vehicle traverses through further time-steps, the uncertainty ellipses will shrink, representing the fact that the EKF-SLAM becomes more certain about the estimated positions of those landmarks as more information is received.

Fig (11), (12), and (13) are the robot's position error plots in

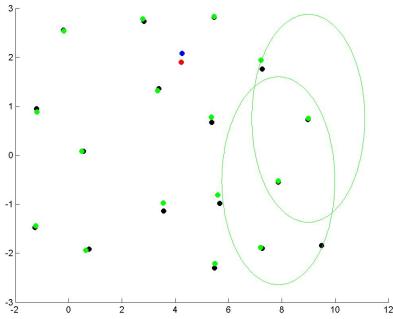


Fig. 10. EKF-SLAM: Covariance Update. In this map, the green uncertainty ellipses around two of the landmarks represent the uncertainty of the estimates for those recently encountered landmarks.

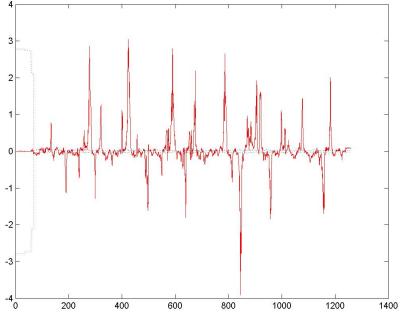


Fig. 11. EKF-SLAM: X Error

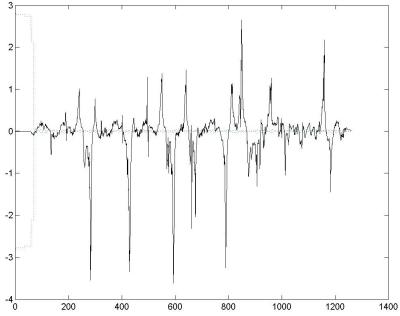


Fig. 12. EKF-SLAM: Y Error

x , y , and θ , respectively. The dotted lines above and below the horizontal axes represent the uncertainty envelope, extracted from the covariance matrix.

C. EKF-SLAM Experiment with Fixed Points

As can be seen from previous results, the robot performs a much better job of localizing itself when the position of all the landmarks is known (the Localization experiment), compared to when the robot has no information about the landmarks beforehand (EKF-SLAM experiment), as would be expected in the presence of sensor noise. However, an interesting experiment would be to see how the robot performs if only a few of the landmarks are fixed to their true positions.

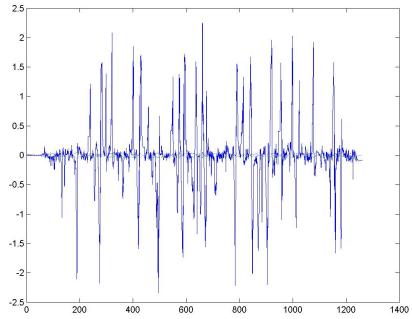


Fig. 13. EKF-SLAM: Theta Error

Fixing a few landmarks to the ground-truth serves a second purpose as well. The EKF-SLAM algorithm described above begins without any a priori knowledge about its environment. It calculates the position/orientation of the robot and the positions of landmarks with respect to the robots point of origin. This point of origin, however, is not associated with any ground-truth information, i.e., the robot will have absolutely no way of connecting the map to any real-world location. This means that the map would be considered valid regardless of its translation and/or rotation from its actual position.

Thus, by fixing one or more landmarks to their ground-truth position, we will provide the vehicle with a direct association to its environment's real-world location. Since this is a two-dimensional SLAM problem, fixing just one landmark to the ground-truth position will lock its x and y dimensions, but the map will still be valid under any orientation as the third dimension, θ is unlocked. Fixing two landmarks, however, will lock all three degrees-of-freedom in place, and the map will become locked to the current environment's position and orientation.

A third experiment was thus conducted using two fixed landmark locations. Fig (14) displays the resulting map at the end of the final time-step. As expected, the final map is more accurate than the one without any ground-truth landmark association, as in Fig (8)

D. Cramer-Rao Lower Bound solution to the EKF-SLAM

In estimation theory, the Cramer-Rao Lower Bound (CRLB) sets a lower-limit on the variance of any unbiased estimator, i.e., it defines a limit on the accuracy of a given estimator [8]. In case of the Extended Kalman Filter, a CRLB solution is reached when the motion and observation model Jacobian matrices are evaluated at the true state of the system, rather than the estimated state, in the Kalman Gain Calculation and Correction steps [2].

The purpose of the fourth experiment conducted was to evaluate the EKF-SLAM problem defined above at its CRLB limit, and compare the results with the results obtained using

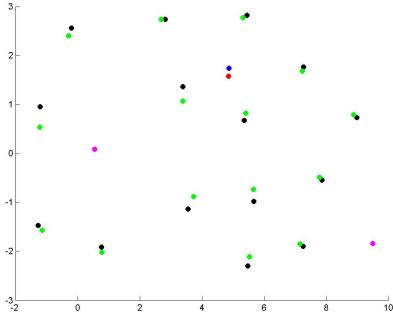


Fig. 14. EKF-SLAM with 2 fixed points: Final Map. In this figure, the red dot represents the true robot position, the blue dot represents the estimated robot position, the black dots represent the true landmark positions, the green dots represent the estimated landmark positions, and the magenta dots represented the landmarks that were fixed to the ground-truth.

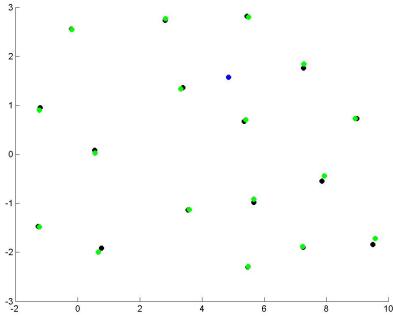


Fig. 15. EKF-SLAM-CRLB: Final Map. In this figure, the red dot represents the true robot position, the blue dot represents the estimated robot position, the black dots represent the true landmark positions and the green dots represent the estimated landmark positions.

the standard EKF-SLAM solution. For this experiment, all landmarks were, once again, un-locked from any ground-truth information.

Fig (15) shows the result of the CRLB EKF-SLAM map at the end of the final time-step, whereas Fig (16) shows the comparison between the robot's estimated and true position under the CRLB solution.

Fig(17), (18), and (19) are the robot position and orientation error plots in x , y , and θ , respectively. The dotted lines above and below the horizontal axes represent the uncertainty envelope, extracted from the covariance matrix.

It is quite clear that this CRLB-EKF-SLAM solution is quite a bit more accurate when compared with the standard EKF-SLAM, and the estimates always remain within the acceptable uncertainty limits. This is because the estimate is not reliant on its own predictions, but rather uses ground-truth data to compensate for noise and generate an estimate for the system. Of course, this solution is impractical in any on-line application as the ground-truth data would not be available. However, it represents the best possible solution our EKF-SLAM algorithm can provide, and thus serves as a

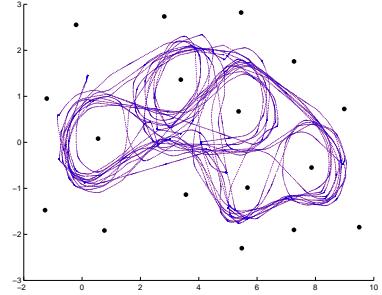


Fig. 16. EKF-SLAM-CRLB: true position vs estimated position. In this figure, the red dots represent the true robot position and the blue dots represent the estimated robot position at each time-step.

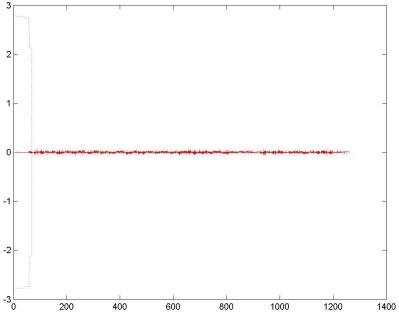


Fig. 17. EKF-SLAM-CRLB: X Error

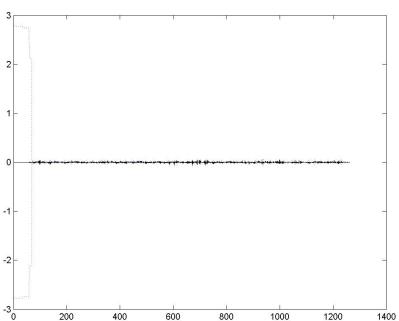


Fig. 18. EKF-SLAM-CRLB: Y Error

good benchmark to compare a particular implementation of the EKF-SLAM.

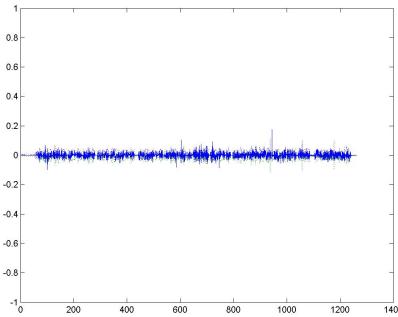


Fig. 19. EKF-SLAM-CRLB: Theta Error

VI. CONCLUSION

The objective of this paper was to demonstrate the implementation of a two-dimensional Simultaneous Localization and Mapping problem using the Extended Kalman Filter estimation technique. We first introduced the SLAM problem and set-up the experiment we would use to demonstrate the EKF-SLAM. Next, we studied why the EKF-SLAM algorithm was chosen for this particular application over a straightforward dead-reckoning and mapping approach. We then went on to define the equations required for the implementation of the EKF, and looked at the recursion cycle utilized to solve the problem. We finally ended with the implementation of the EKF-SLAM in MATLAB, and a discussion of the results obtained.

As pointed out previously, the EKF-SLAM is not the only established solution to the SLAM problem. One popular approach is the Rao-Blackwellized FastSLAM problem, which looks to reduce the time and space complexity required to compute a SLAM solution. As apparent from the System Motion and Observation models and their Jacobians, equations (1) to (14), the matrices in EKF-SLAM increase in two dimensions as the number of landmarks encountered increases. This means that for N landmarks, the worst-case complexity of an EKF based SLAM implementation would be $O(N^2)$. The FastSLAM approach reduces this complexity to $O(N \log N)$, which can be significantly more useful in many on-line applications. Further research and comparison between the EKF-SLAM and the FastSLAM is left to the reader.

REFERENCES

- [1] H. Durrant-Whyte and T. Bailey. June 2006. Tutorial - Simultaneous Localization and Mapping: Part I. *Robotics & Automation Magazine, IEEE* Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1638022
- [2] T.D. Barfoot, "State Estimation for Aerospace Vehicles - AER1513 Course Assignments Rev: 2.0", October 2009. [Online] Available: <http://asrl.utias.utoronto.ca/~tdb/aer1513/>
- [3] H. Durrant-Whyte. "Localisation, Mapping and the Simultaneous Localisation and Mapping (SLAM) Problem", Australian Centre for Field Robotics, The University of Sydney NSW, Australia. 2002. [Online] Available: <http://www.cas.kth.se/SLAM/Presentations/hdw-slides.pdf>
- [4] T.D. Barfoot, "State Estimation for Aerospace Vehicles - AER1513 Lecture 5-7: Nonlinear-Non-Gaussian Estimation Rev: 1.0", 2009. [Online] Available: <http://asrl.utias.utoronto.ca/~tdb/aer1513/>
- [5] E. Nebot. "Simultaneous Localization and Mapping 2002 Summer School Ver: 0.9", Australian Centre for Field Robotics, The University of Sydney NSW, Australia. 2006 [Online]
- [6] S. Riisgaard and M. R. Blas. "SLAM for Dummies - A Tutorial Approach to Simultaneous Localization and Mapping". 2004 [Online] Available: <http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j>
- [7] J Guivant, E Nebot and S Baker. October 2000. Autonomous Navigation and Map building Using Laser Range Sensors in Outdoor Applications. *Journal of Robotic Systems, Vol 17*. Available: http://www.cas.kth.se/SLAM/Papers/slam_robots.pdf
- [8] T. D. Barfoot. "State Estimation for Aerospace Vehicles - AER1513 Lecture 2: Primer on Probability Rev: 1.0", 2009. [Online] Available: <http://asrl.utias.utoronto.ca/~tdb/aer1513/>