Special Problems Summer 2018

Planar Robot Localization with the Extended Kalman Filter

Sanandeesh Kamat *ssk93*

August 1, 2018

Contents

This final paper replicates a subset of procedures and findings described in *Probabilistic Robotics* by Sebastian Thrun et all (2005) which is referred to herein as **Thrun2005**. This author is deeply indebted to Dr. Thrun and his colleagues for producing this invaluable exposition into this fascinating area of research.

1 Introduction

This paper provides a limited introduction to the field of probabilistic robotics by demonstrating the performance of the Extended Kalman Filter (EKF) in localizing a planar robot. Localization is the process of estimating the pose of a robot relative to a global coordinate system. The planar robot pose consists of translational position (x, y) and angular orientation θ. Ideally, the robot would move precisely where it commanded itself to move and the sensor measurements would depict the environment precisely as it is. However, in reality, there exists randomness both in the evolution of state (i.e. the process) and the generation of measurements (i.e. observations). These uncertainties may arise due to imperfect robot actuators, dynamic environmental obstructions, and sensor noise. As such, the deterministic models of state transition and measurement produce expectations at best. The belief of a robot's environment and the outcome of given motion command can only be known with degrees of confidence.

When approached through probability theory, these degrees of confidence are represented explicitly as probability distribution functions (PDF). To meet the ever increasing complexities of modern robotics applications, roboticists rely heavily upon the algorithms which explicitly model situational uncertainty. The common thread unifying probabililistic robotics the *Bayes filter*, which is Bayes theorem exercised recursively over time. Bayes filters produce optimal estimates of state given noisy measurements, an imperfect measurement model, and an imperfect state transition model. These optimal estimates (or posterior beliefs) of the robot state are given as multi-dimensional probability density functions. The PDF is tightened by information gained by sensor measurements, and is loosened by information lost through state transition processes.

Bayesian filtering integrates imperfect process/measurement models, and sensor data in a way which overcomes their induvidual limiations. By accomodating these inherent imperfections, Bayesian filtering reduces the accuracy requirements of software models and hardware sensors. Of all approaches to robotics, none scale so naturally to increasingly uncertain environments as probabilistic robotics. Often cited drawbacks of probabilistic robotics algorithms are computational complexity and forced approximations. Traditional parameteric PDFs (e.g. Gaussian) may not sufficiently approximate complex multi-model beliefs. On the other hand, non-parametric PDFs defined over the entire space of possible robot/environment states can quickly become computationally intractable.

2 Belief as Recursive State Estimation

This section introduces the basic mathematical framework with which Bayesian filters are designed and state estimation is achieved. The core idea of state estimation is that a random state vector is sought, but is not directly observable. Rather, there is some measurement (ideally) correlated with the desired state vector which *is* directly observable, but is random as well. An optimal estimator is defined as that which minimizes the mean square error between the true state vector and estimated state vector. This criterion of optimality is met by an estimator which maximizes the posterior density of the state vector given the measurement vector. As such, *characeterizing* the posterior density of the state vector becomes of interest. Bayesian filters are designed to iteratively achieve optimal estimates of an evolving random state vector by combining on each iteration the prediction of the state based on state transition models with the inference of the state based on sensor measurement models. Quantities such as robot state, sensor measurement, and environment state are all modelled as *random vectors*, and so are all represented by probability density functions written as $p(\mathbf{x})$. The most commonly employed density function is the N-dimensional Gaussian written as

$$p(\mathbf{x}) = \frac{e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}}{(2\pi)^{\frac{N}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}}, \tag{1}$$

for which $\mathbf{x}$ is the $[N \times 1]$ state vector, μ is the $[N \times 1]$ expectation of the state vector $\mathbf{x}$, $\mathbf{\Sigma}$ is the $[N \times N]$ covariance of the state vector $\mathbf{x}$. A random vector under this distribution is abbreviated as $\mathbf{N}(\mathbf{x}; \mu, \mathbf{\Sigma})$.

Central to recursive state estimation in probabilistic inference is *Bayes Theorem* which is stated as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{\int p(\mathbf{y}|\mathbf{x}')p(\mathbf{x}')d\mathbf{x}'} \tag{2}$$

The state vector $\mathbf{x}$ is to be estimated from the measurement vector $\mathbf{y}$. $p(\mathbf{x}|\mathbf{y})$ is called the *posterior* density because it represents the conditional belief of state vector $\mathbf{x}$ after incorporating measurement vector $\mathbf{y}$. $p(\mathbf{x})$ is called the *prior* density because it represents the marginal belief of state vector $\mathbf{x}$ prior to incorprating measurement vector $\mathbf{y}$. $p(\mathbf{y}|\mathbf{x})$ is called the *likelihood* density becuase it represents the conditional likelihood of having seen measurement vector $\mathbf{y}$ given the state vector $\mathbf{x}$. Finally, the denominator $p(\mathbf{y})$ is called the normalizer, because is not a function of $\mathbf{x}$ and only forces the density to integrate to 1. Therefore, selecting state vector $\mathbf{x}$ to maximize the posterior density is achieved by maximizing instead the product of the prior and the likelihood densities.

The *state* of robots and environments is characterized by the following two state variables

- *Pose* is the location and orientation of the robot relative to a global coor-

dinate system. For planar robots, the pose vector is

$$\mathbf{x}_t = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix}. \tag{3}$$

Here, (x_t, y_t) are the cartesian location coordinates, and θ_t is the bearing.

- *Landmark Features* are the locations relative to the same global coordinate system and signatures of environmental objects which are assumed to be easily distinguishable and stationary. For planar landmarks, the *map* matrix is

$$\mathbf{m} = \begin{pmatrix} m_x^1 & & m_x^N \\ m_y^1 & \cdots & m_y^N \\ s^1 & & s^N \end{pmatrix}. \tag{4}$$

Here, m_x^j, m_y^j are the cartesian location coordinates, s^j is the landmark signature (e.g. color, RCS, temperature).

The *interaction* between robots and environments is characterized by the following two variables

- *Sensor Measurements* of the environment are collected by the robot. This paper assumes a *range sensor* with full angular coverage. The measurement matrix taken of the locally visible landmarks are

$$\mathbf{z}_t = \begin{pmatrix} r_t^1 & & r_t^N \\ \phi_t^1 & \cdots & \phi_t^N \\ s_t^1 & & s_t^N \end{pmatrix}. \tag{5}$$

Each landmark measurent consists of range r_t^j, bearing ϕ_t^j, and signature s_t^j in the local coordinate frame of robot.

- *Control Actions* taken by the robot impart change onto robot/enviroment state. This paper only considers *robot motion* from one pose to the next. The control vector applied by the robot is

$$\mathbf{u}_t = \begin{pmatrix} v_t \\ w_t \end{pmatrix}. \tag{6}$$

Each control action consists of an applied tranlational velocity v_t and angular velocity w_t.

2.1 The Bayesian Filter Framework

Sensor measurements $\mathbf{z}_t$ provide information about the environment and so reduce uncertainty. Control actions $\mathbf{u}_t$ introduce noisy outcomes to state and so increase uncertainty. Although sensor measurements and control actions occur simultaneously, they are conceptualized as occuring in sequence to accomodate the Bayes filter algorithm.

For localization, the map $\mathbf{m}$ is available to the robot and the pose $\mathbf{x}_t$ is to be estimated.

If the stochastic process $\mathbf{x}$ possesses the Markov property, the density can be expressed as

$$p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t), \tag{7}$$

This expression indicates that $\mathbf{x}_{t-1}$ is a *sufficient statistic* of all previous states $\mathbf{x}_{1:t-2}$, controls $\mathbf{u}_{1:t-1}$, and measurements $\mathbf{z}_{1:t-1}$. Markov processes are characterized by conditional independence between past and future states given the present state. This special property offers computational tractability to many Bayesian filtering algorithms such as the EKF presented in this paper. Similarly, the sufficiency of $\mathbf{x}_t$ causes the likelihood density to simplify as such,

$$p(\mathbf{z}_t|\mathbf{x}_{0:t}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) = p(\mathbf{z}_t|\mathbf{x}_t). \tag{8}$$

This expression indicates that the current state $\mathbf{x}_t$ is a sufficient statistic for characterizing the current measurement density, regardless of past events. The two densities simplified by the sufficiency of $\mathbf{x}$ are defined below.

- *State Transition Probability* $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$ represents the stochastic evolution of state $\mathbf{x}_t$ from one time step to the next.
- *Measurement Probability* $p(\mathbf{z}_t|\mathbf{x}_t)$ represents the stochastic generation of sensor measurements $\mathbf{z}_t$ from the conditioned state $\mathbf{x}_t$.

Together, these two densities model the stochastic robot/environment system (Sarkka, 2013).

Recursive state estimation techniques model the internal *belief* of state as posterior probability distributions conditioned measurements and control. Hence, in this paper, the term *belief*,

$$bel(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{z}_{1:t}, \mathbf{u}_{1:t}), \tag{9}$$

abbreviates posterior probability distributions.

Before computing $bel(\mathbf{x}_t)$, however, the Bayes filter requires the predicted belief prior to incorporation of the latest measurement $\mathbf{z}_t$,

$$\overline{bel}(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \tag{10}$$

1	**BayesFilter**$(bel(\mathbf{x}_{t-1}), \mathbf{u}_t, \mathbf{z}_t)$
2	for $i = 1 : N$
3	$\quad \overline{bel}(\mathbf{x}_t^i) = \int p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}) bel(\mathbf{x}_{t-1}) dx_{t-1}$
4	$\quad bel(\mathbf{x}_t^i) = p(\mathbf{z}_t|\mathbf{x}_t)\overline{bel}(\mathbf{x}_t)p(\mathbf{z}_t)^{-1}$
5	endfor
6	return $bel(\mathbf{x}_t)$

Table 1: Bayes Filter Algorithm

Table 1 describes the Bayes filter algorithm in terms of beliefs. The latest belief $bel(\mathbf{x}_t)$ is calculated from the previous belief $bel(\mathbf{x}_{t-1})$, the latest control $\mathbf{u}_t$, and the latest measurement $\mathbf{z}_t$. Internally, there are two essential stages.

- *Prediction Update* occurs in line 3 of Table 1. By invoking the *Law of Total Probability*, the predicted belief $\overline{bel}(\mathbf{x}_t^i)$ is the integral of the product of motion model and prior belief.
- *Measurement Update* occurs in line 4 of Table 1. By invoking Bayes Theorem, the updated posterior belief $bel(\mathbf{x}_t)$ is the product of the predicted belief $\overline{bel}(\mathbf{x}_t)$ and measurement probability $p(\mathbf{z}_t|\mathbf{x}_t)$. The denominator $p(\mathbf{z}_t)^{-1}$ ensures that $bel(\mathbf{x}_t)$ integrates to 1.

To summarize, three distributions are required to implement a Bayes filter for dynamic state estimation.

$\mathbf{x}_0 \sim p(\mathbf{x}_0)$	Initial State
$\mathbf{z}_t \sim p(\mathbf{z}_t\|\mathbf{x}_t)$	Measurement
$\mathbf{x}_t \sim p(\mathbf{x}_t\|\mathbf{x}_{t-1}, \mathbf{u}_t)$	State Transition

Table 2: Distributions for Recursive State Estimation

The Bayes filter, in this representation, is not tractable for implementation on a digital computer. Instead, there exist a variety of estimators derived from the Bayes filter algorithm for which the initial states, measurement models, state transition models, and posterior distributions take different forms.

2.2 The Kalman Filter and the Extended Kalman Filter

Gaussian filters are reccursive state estimators for which the posterior densities are represented as N-dimensional Gaussian random vectors,

$$p(\mathbf{x}) = \frac{e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}}{(2\pi)^{\frac{N}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}}. \tag{11}$$

$\mathbf{x}$ is the $[N \times 1]$ state vector, μ is the $[N \times 1]$ expectation of the state vector $\mathbf{x}$, $\mathbf{\Sigma}$ is the $[N \times N]$ covariance of the state vector $\mathbf{x}$. A random vector under this distribution is abbreviated as $\mathbf{N}(\mathbf{x}; \mu, \mathbf{\Sigma})$. This is known as *moments parametrization* because the mean μ and covariance Σ parameterize the density of $\mathbf{x}$.

The Gaussian assumption is only valid when the true state density is unimodal such as during local tracking. Hence, global estimation problems for which state density can be multimodal are handled with other representations such as Gaussian mixture models or Monte Carlo (particle) filters.

The *Kalman Filter* is a Gaussian filter for which state belief is specified with moments parametrization, $\mathbf{x} \sim \mathbf{N}(\mathbf{x}; \mu, \mathbf{\Sigma})$. In addition, the state transition $p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1})$ and measurement probabilities $p(\mathbf{z}_t|\mathbf{x}_t)$ are expected to be linear in the arguments with additive independent Gaussian noise. Such systems are referred to as *linear Gaussian systems*.

- *Distribution of Linear State Transition* $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$

  The trasition of state from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ is described by

$$\mathbf{x}_t = \mathbf{A}_t\mathbf{x}_{t-1} + \mathbf{B}_t\mathbf{u}_t + \epsilon_t. \tag{12}$$

  $\mathbf{A}_t$ is an nxn matrix, where n is the dimensionality of the state vector. It incorporates the previous state $\mathbf{x}_{t-1}$ into the state transition process. $\mathbf{B}_t$ is a nxm matrix, where m is the dimensionality of the control vector. It incorporates the robot control action $\mathbf{u}_t$ into the state transition process. ϵ_t is an nx1 zero-mean Gaussian random vector, $\epsilon_t \sim \mathbf{N}(\mathbf{0}, \mathbf{R}_t)$, which models the uncertainty introduced by stochastic state transition. The moments parameters of this linear Gaussian system are

$$\mu_t = E[\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}] = \mathbf{A}_t\mathbf{x}_{t-1} + \mathbf{B}_t\mathbf{u}_t,$$

  and

$$\mathbf{\Sigma}_t = E[\mathbf{x}_t\mathbf{x}_t^T|\mathbf{u}_t, \mathbf{x}_{t-1}] = \mathbf{R}_t.$$

  Inserting these parameters into the Gaussian distribution function (eq 11) yields

$$p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}) = \frac{e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{A}_t\mathbf{x}_{t-1} - \mathbf{B}_t\mathbf{u}_t)^T\mathbf{R}_t^{-1}(\mathbf{x}_t - \mathbf{A}_t\mathbf{x}_{t-1} - \mathbf{B}_t\mathbf{u}_t)}}{(2\pi)^{\frac{N}{2}}|\mathbf{R}_t|^{\frac{1}{2}}}. \tag{13}$$

- *Distribution of Linear Measurement* $p(\mathbf{z}_t|\mathbf{x}_t)$

  The generation of a measurement $\mathbf{z}_t$ from a state $\mathbf{x}_t$ is described by

$$\mathbf{z}_t = \mathbf{C}_t\mathbf{x}_t + \delta_t \tag{14}$$

  $\mathbf{C}_t$ is a k by n matrix, where k is the dimension of the measurement vector $\mathbf{z}_t$. It incorporates state $\mathbf{x}_t$ into the measurement generation process. δ_t is an kx1 zero-mean Gaussian random vector, $\delta_t \sim \mathbf{N}(\mathbf{0}, \mathbf{Q}_t)$, which models the uncertainty introduced by stochastic measurement generation. The moments parameters of this linear Gaussian system are

$$\mu_t = E[\mathbf{z}_t|\mathbf{x}_t] = \mathbf{C}_t\mathbf{x}_{t-1},$$

  and

$$\mathbf{\Sigma}_t = E[\mathbf{z}_t\mathbf{z}_t^T|\mathbf{x}_t] = \mathbf{Q}_t.$$

  Inserting these parameters into the Gaussian distribution function (eq 11) yields

$$p(\mathbf{z}_t|\mathbf{x}_t) = \frac{e^{-\frac{1}{2}(\mathbf{z}_t - \mathbf{C}_t\mathbf{x}_t)^T\mathbf{Q}_t^{-1}(\mathbf{z}_t - \mathbf{C}_t\mathbf{x}_t)}}{(2\pi)^{\frac{N}{2}}|\mathbf{Q}_t|^{\frac{1}{2}}} \tag{15}$$

- *Normally Distributed Initial State* $p(\mathbf{x}_0)$

  The initial belief of state is normally distributed with an expectaion of μ_0 and an uncertainty ellipse of $\mathbf{\Sigma}_0$,

$$p(\mathbf{x}_0) = \frac{e^{-\frac{1}{2}(\mathbf{x}_0 - \mu_0)^T\mathbf{\Sigma}_0^{-1}(\mathbf{x}_0 - \mu_0)}}{(2\pi)^{\frac{N}{2}}|\mathbf{\Sigma}_0|^{\frac{1}{2}}}. \tag{16}$$

The three bullet pointed conditions above enable the Kalman filter algorithm to operate as described in Table 3. The inputs are the previous state belief parameters ($\mu_{t-1}\mathbf{\Sigma}_{t-1}$), the current control action $\mathbf{u}_t$, and current measurement set $\mathbf{z}_t$. The outputs are the current state belief parameters ($\mu_t\mathbf{\Sigma}_t$).

The Kalman filter prediction update is described by lines 2 and 3. The predicted state (prior to incorporation of the measurement) $\overline{\mu}_t$ is the expectation of state transition evaluated from the previous state estimate μ_{t-1}. The covariance of the predicted state $\overline{\mathbf{\Sigma}}_t$ is the sum of the transition model uncertainty $\mathbf{R}_t$ and the previous state uncertainty projected across time from $t-1$ to t. The Kalman filter measurement update is described by lines 4 to 6. The Kalman Gain $\mathbf{K}_t$ controls the extent to which the measurement $\mathbf{z}_t$ influences the final state estimate $(\mu_t, \mathbf{\Sigma}_t)$. It is ratio of the cross-covariance between measurements and state, $\overline{\mathbf{\Sigma}}_t\mathbf{C}_t^T$, to the covariance of measurements, $(\mathbf{C}_t\overline{\mathbf{\Sigma}}_t\mathbf{C}_t^T + \mathbf{Q}_t)$. The innovation factor, $(\mathbf{z}_t - \mathbf{C}_t\overline{\mu}_t)$, is the difference between the actual measurement $\mathbf{z}_t$ and the expected measurement $E[\mathbf{z}_t|\overline{\mu}_t]$. Lines 5 and 6 can be recognized as the *linear minimum mean squared error* estimator equations. Because the the state $\mathbf{x}_t$ and measurements $\mathbf{u}_t$ are jointly Gaussian, the Kalman Filter yields the *non-linear MMSE* and is, hence, an optimal estimator. Kalman filters require the transition and measurement models to be linear because Gaussianity is maintained only under linear transformations. This enables the posterior beliefs computed recursively to remain Gaussian, and the resulting closed form representations to remain optimal.

1	**KalmanFilter**$(\mu_{t-1}, \mathbf{\Sigma}_{t-1}, \mathbf{u}_t, \mathbf{z}_t)$
2	$\overline{\mu}_t = \mathbf{A}_t\mu_{t-1} + \mathbf{B}_t\mathbf{u}_t$
3	$\overline{\mathbf{\Sigma}}_t = \mathbf{A}_t\mathbf{\Sigma}_{t-1}\mathbf{A}_t^T + \mathbf{R}_t$
4	$\mathbf{K}_t = \overline{\mathbf{\Sigma}}_t\mathbf{C}_t^T(\mathbf{C}_t\overline{\mathbf{\Sigma}}_t\mathbf{C}_t^T + \mathbf{Q}_t)^{-1}$
5	$\mu_t = \overline{\mu}_t + \mathbf{K}_t(\mathbf{z}_t - \mathbf{C}_t\overline{\mu}_t)$
6	$\mathbf{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{C}_t)\overline{\mathbf{\Sigma}}_t$
7	return $\mu_t, \mathbf{\Sigma}_t$

Table 3: The Kalman Filter Algorithm

Unfortunately, the restriction to *linear* Gaussian systems significantly reduces the Kalman filter's applicability to most robotics problems. The motion and measurement models employed in this paper, for example, are non-linear transformations. Therefore, filter designers resort to Extended Kalman Filter (EKF), which linearize nonlinear state transition and measurement models with Taylor series expansions. Linearization approximates the non-linear function by a linear function that is tangent to the function at the mean of the input parameter. This matrix representation of multiple multi-variable non-linear functions which have all been linearized with respect each variable is a *Jacobian* matrix. Because these linearized representations of non-linear functions are only approximations, so, too, are the estimated posteriors. The quality of the approximated posteriors is worsened by (1) high input uncertainty and (2) high local-nonlinearity of the function.

- *Distribution of Non-Linear State Transition* $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$

  The non-linear state transition function g replaces linear transformation matrices $\mathbf{A}_t$ and $\mathbf{B}_t$ in eq 12.

$$\mathbf{x}_t = g(\mathbf{u}_t, \mathbf{x}_{t-1}) + \epsilon_t \tag{17}$$

  The Jacobian of the state transition model is the given as

$$\mathbf{G}_t = \frac{\partial g(\mathbf{u}_t, \mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}}.$$

  Therefore, the linearized (approximated) state transition model is given as

$$g(\mathbf{u}_t, \mathbf{x}_{t-1}) \approx g(\mathbf{u}_t, \mu_{t-1}) + \mathbf{G}_t(\mathbf{x}_{t-1} - \mu_{t-1}).$$

  The moments parameters of this linearized Gaussian system are

$$\mu_t = E[\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}] = g(\mathbf{u}_t, \mu_{t-1}) + \mathbf{G}_t(\mathbf{x}_{t-1} - \mu_{t-1}),$$

  and

$$\mathbf{\Sigma}_t = E[\mathbf{x}_t\mathbf{x}_t^T|\mathbf{u}_t, \mathbf{x}_{t-1}] = \mathbf{R}_t.$$

  Inserting these parameters into the Gaussian distribution function (eq 11) yields

$$p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}) \approx \frac{e^{-\frac{1}{2}(\mathbf{x}_t - g(\mathbf{u}_t, \mathbf{x}_{t-1}) - \mathbf{G}_t(\mathbf{x}_{t-1} - \mu_{t-1}))^T \mathbf{R}_t^{-1} (\mathbf{x}_t - g(\mathbf{u}_t, \mathbf{x}_{t-1}) - \mathbf{G}_t(\mathbf{x}_{t-1} - \mu_{t-1}))}}{(2\pi)^{\frac{N}{2}} |\mathbf{R}_t|^{\frac{1}{2}}} \tag{18}$$

- *Distribution of Non-Linear Measurement* $p(\mathbf{z}_t|\mathbf{x}_t)$

  The non-linear measurement function h replaces linear transformation matrix $\mathbf{C}_t$ in eq 14.

$$\mathbf{z}_t = h(\mathbf{x}_t) + \delta_t \tag{19}$$

  The Jacobian of the measurement model is the given as

$$\mathbf{H}_t = \frac{\partial h(\mathbf{x}_t)}{\partial \mathbf{x}_t}$$

  Therefore, the linearized (approximated) measurement model is given as

$$h(\mathbf{x}_t) \approx h(\mu_t) + \mathbf{H}_t(\mathbf{x}_t - \overline{\mu}_t)$$

  The moments parameters of this linearized Gaussian system are

$$\mu_t = E[\mathbf{z}_t|\mathbf{x}_t] = h(\mu_t) + \mathbf{H}_t(\mathbf{x}_t - \overline{\mu}_t),$$

  and

$$\mathbf{\Sigma}_t = E[\mathbf{z}_t\mathbf{z}_t^T|\mathbf{x}_t] = \mathbf{Q}_t.$$

  Inserting these parameters into the Gaussian distribution function (eq 11) yields

$$p(\mathbf{z}_t|\mathbf{x}_t) = \frac{e^{-\frac{1}{2}(\mathbf{z}_t - h(\mu_t) - \mathbf{H}_t(\mathbf{x}_t - \overline{\mu}_t))^T \mathbf{Q}_t^{-1} (\mathbf{z}_t - h(\mu_t) - \mathbf{H}_t(\mathbf{x}_t - \overline{\mu}_t))}}{(2\pi)^{\frac{N}{2}} |\mathbf{Q}_t|^{\frac{1}{2}}} \tag{20}$$

The EKF algorithm is described in Table 4. This algorithm is identical to the Kalman filter algorithm in Table 3, except in a few ways. In line 2, the predicted state $\overline{\mu}_t$ is the direct output of the non-linear state transition model. Similarly, in line 5, the predicted measurement is $\hat{\mathbf{z}}_t$ is the direct output of the non-linear measurement model. Finally, the matrices $\mathbf{G}_t$ and $\mathbf{H}_t$ are the Jacobian matrices computed via Taylor series expansion as described above.

1	**ExtendedKalmanFilter**$(\mu_{t-1}, \mathbf{\Sigma}_{t-1}, \mathbf{u}_t, \mathbf{z}_t)$
2	$\overline{\mu}_t = g(\mathbf{u}_t, \mu_{t-1})$
3	$\overline{\mathbf{\Sigma}}_t = \mathbf{G}_t \mathbf{\Sigma}_{t-1} \mathbf{G}_t^T + \mathbf{R}_t$
4	$\mathbf{K}_t = \overline{\mathbf{\Sigma}}_t \mathbf{H}_t^T (\mathbf{H}_t \overline{\mathbf{\Sigma}}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1}$
5	$\mu_t = \overline{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - h(\overline{\mu}_t))$
6	$\mathbf{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \overline{\mathbf{\Sigma}}_t$
7	return $\mu_t, \mathbf{\Sigma}_t$

Table 4: The Extended Kalman Filter Algorithm

Jacobian matrices represent the amount of *sensitivity* nonlinear multi-variable functions have *to error* at any given point and with respect to any given variable. Hence, the coefficients of the Jacobian matrices are used to scale covariance values of random vectors like so, $\mathbf{C}_{Out} = \mathbf{J}\mathbf{C}_{In}\mathbf{J}^T$. This operation is crucial for projecting covariance (i.e. uncertainty) from one space to another. Examples of spaces covariance can belong to include measurement space $\mathbf{Q}_t$, motion space $\mathbf{R}_t$, and state space $\mathbf{\Sigma}_t$.

3 Planar Robot Motion & Measurement Models

This section presents the probabilistic process models driving the planar robot simulation. These are models which have their origins in classical robotics, but have been *probabilified* by adding random noise parameters which represent the uncertainty in robotic action and perception. Intuitively, these noise parameters would result from rigorous modelling of the true system dynamics. However, in practice, the mere inclusion of random noise itself affords succesful state estimation. Furthermore, deliberate overestimation of the random noise fortifies the performance of the filter against the violations of the Markov assumption, such as unmodelled state parameters. As such, the simple motion and measurement models described herien have enjoyed succesful results in real robotic applications.

3.1 Velocity Motion Model

The motion model dictates the state transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$. The outcome of a control action is uncertain, and so is described a posterior distribution. This representation enables it to be incorporated into Bayes filters such as the EKF.

The velocity motion model assumes that the control action $\mathbf{u}_t = (v_t, w_t)^T$ applies independant translational velocity v_t and rotational velocity w_t commands to the robot motors. The deterministic result of this control action is specified as

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} -\frac{v}{w}\sin(\theta) + \frac{v}{w}\sin(\theta + w\Delta t) \\ \frac{v}{w}\cos(\theta) - \frac{v}{w}\cos(\theta + w\Delta t) \\ w\Delta t \end{pmatrix} \tag{21}$$

and results in circular motion along an arc of radius

$$r = \left|\frac{v}{w}\right|.$$

Notice how a motion along a straight line ($w_t = 0$) would result in an effectively infinite radius. The interpretation of this model is that after Δt seconds, the robot has moved $v_t\Delta t$ meters along the arc of the path and turned by $w_t\Delta t$ radians.

In reality, this deterministic model is subject to noise. The noisy model mimics the deterministic model, but adds noise to the commanded velocities at each time step.

$$\begin{pmatrix} \hat{v} \\ \hat{w} \\ \hat{\gamma} \end{pmatrix} = \begin{pmatrix} v \\ w \\ 0 \end{pmatrix} + \begin{pmatrix} \epsilon_{\alpha_1 v^2 + \alpha_2 w^2} \\ \epsilon_{\alpha_3 v^2 + \alpha_4 w^2} \\ \epsilon_{\alpha_5 v^2 + \alpha_6 w^2} \end{pmatrix}. \tag{22}$$

This additional noise parameter ϵ_{b^2} is a zero-mean Normal random variable, $\epsilon_{b^2} \sim \mathbf{N}(0, b^2)$. The parameters $\alpha_1 \cdots \alpha_6$ are adjusted model the amount of

variance contributed by v_t, w_t. Simply adding noise to v_t, w_t will result in random motion that *is still circular*. In order to achieve truly random motion which breaks the circular trajectory, an additional parameter $\hat{\gamma}$ is inserted to randomly adjust the final robot bearing θ. The revised velocity motion model with randomized outcomes is

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} -\frac{\hat{v}}{\hat{w}}\sin(\theta) + \frac{\hat{v}}{\hat{w}}\sin(\theta + \hat{w}\Delta t) \\ \frac{\hat{v}}{\hat{w}}\cos(\theta) - \frac{\hat{v}}{\hat{w}}\cos(\theta + \hat{w}\Delta t) \\ \hat{w}\Delta t + \hat{\gamma}\Delta t \end{pmatrix}. \tag{23}$$

The probability of any motion outcome is the product of the probabilities of the underlying motion commands (due to their independence).

$$p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}) = p(\epsilon_{\alpha_1 v^2 + \alpha_2 w^2}) p(\epsilon_{\alpha_3 v^2 + \alpha_4 w^2}) p(\epsilon_{\alpha_5 v^2 + \alpha_6 w^2}) \tag{24}$$

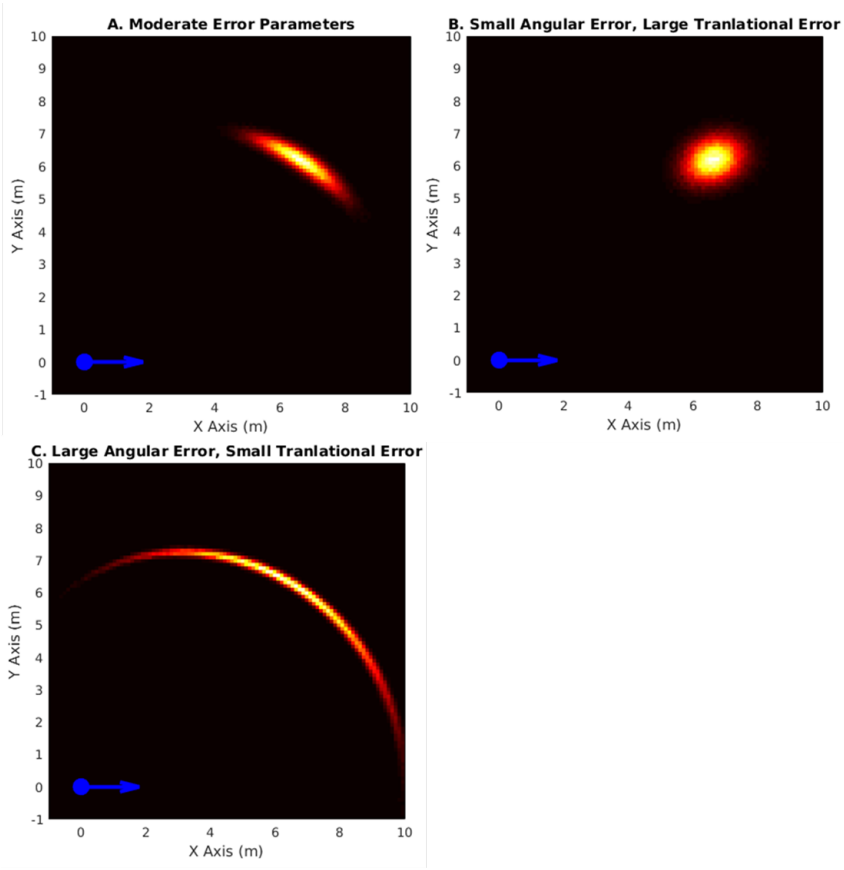

Figure 1: Histograms of Noisy Velocity Motion Model Outputs

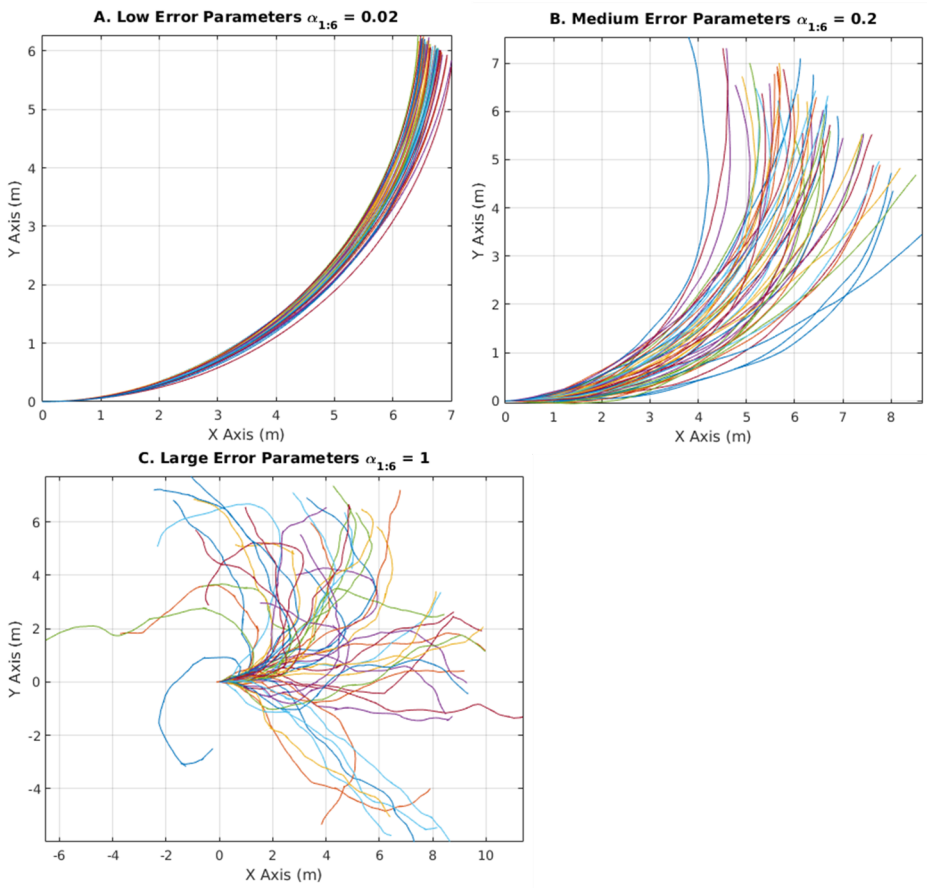

Figure 2: Random Walks from Noisy Velocity Motion Model

3.2 Feature Measurement Model

This paper assumes the usage of a range sensor with full angular coverage. Furthermore, this paper assumes that *features* are extracted from the measurements of *landmarks*.

The measurement model dictates the measurement probability $p(\mathbf{z}_t|\mathbf{x}_t)$. This describes the generative process by which sensor measurements of the environment arise. This paper assumes a *range sensor* with full angular coverage. The measurement matrix taken of the locally visible landmarks are

$$\mathbf{z}_t = \begin{pmatrix} r_t^1 & & r_t^N \\ \phi_t^1 & \cdots & \phi_t^N \\ s_t^1 & & s_t^N \end{pmatrix}. \tag{25}$$

Each landmark measurent consists of range r_t^j, bearing ϕ_t^j, and signature s_t^j in the local coordinate frame of robot.

Assuming conditional independence between measurements of induvidual landmarks, the likelihood of a measurement set $\mathbf{z}_t$ is

$$p(\mathbf{z}_t|\mathbf{x}_t, m) = \prod_{k=1}^{K} p(z_t^k|x_t, m) = \prod_{k=1}^{K} p(r_t^k, \phi_t^k, s_t^k|x_t, m). \tag{26}$$

$$\begin{pmatrix} r_r^i \\ \phi_t^i \\ s_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x_t)^2 + (m_{j,y} - y_t)^2} \\ atan2(m_{j,y} - y_t, m_{j,x} - x_t) - \theta \\ s_j \end{pmatrix} + \begin{pmatrix} \epsilon_{\sigma_r^2} \\ \epsilon_{\sigma_\phi^2} \\ \epsilon_{\sigma_s^2} \end{pmatrix} \tag{27}$$

given that $\epsilon_{b^2} \sim \mathbf{N}(0, b^2)$. In robotics, the problem of associating a measured landmark with a known map landmark is known as the *correspondence problem*. For simplicity, this paper assumes that the robot can perform perfect correspondence.

$$p(\mathbf{z}_t|\mathbf{x}_t, m) = \prod_{k=1}^{K} p(r_t^k - \hat{r}; 0, \sigma_r) p(\phi_t^k - \hat{\phi}; 0, \sigma_\phi) p(s_t^k - \hat{s}; 0, \sigma_s) \tag{28}$$

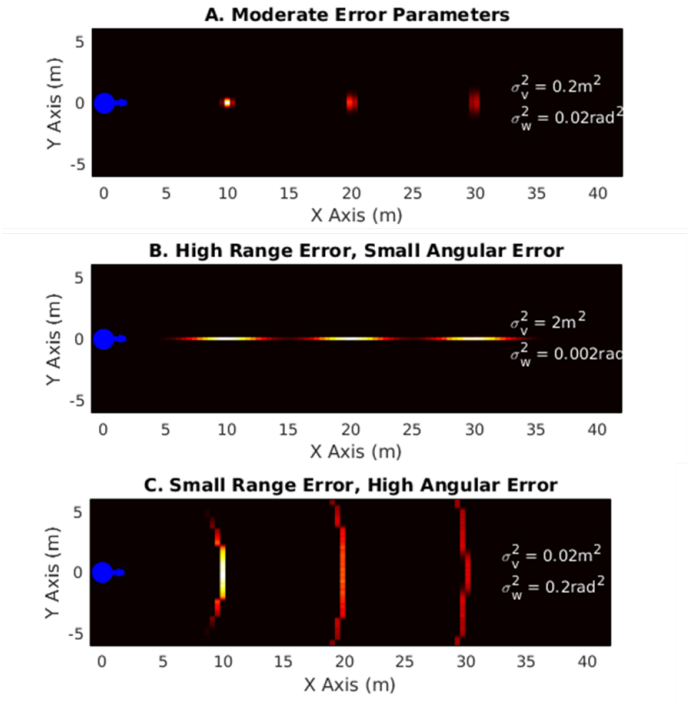

Figure 3: Histograms of Noisy Feature Measurement Model Outputs

4 Markov Localization with the Extended Kalman Filter

The application of Bayes filter to the localization problem is known as *Markov Localization*. Robot localization is the operation of estimating the robot pose $\mathbf{x}_t = (x, y, \theta)^T$ relative to a reference coordinate frame. Hence, Markov Localization propogates posterior distributions over pose $\mathbf{x}_t$ from one time step to the next. The external inputs to this operation consist of measurements of landmarks in the local coordinate frame, each of which correspond to elements of a known map of landmarks in the reference coordinate frame. This operation can be conceptualized as a hidden markov model. Table 5 summarizes the key features of the particular localization algorithm implemented here.

Local Localization	Initial position is assumed known with small error.
Static Environment	Landmarks within the environment are assumed stationary.
Passive Localization	Measurements and pose estimates do not affect the contral actions.
Single Robot	Pose is determined solely from one robot.

Table 5: Taxonomy of the Localization Algorithm

When EKF localization is employed, the belief $bel(\mathbf{x}_t)$ is entirely characterized by the mean μ_t and covariance $\mathbf{\Sigma}_t$. The algorithm is entirely described within Table 6. The algorithm accepts as input the mean μ_{t-1} and covariance $\mathbf{\Sigma}_{t-1}$ of the previous robot pose belief, the velocity motion commands $\mathbf{u}_t$, the set of landmark feature measurements $\mathbf{z}_t$, the map $\mathbf{m}$, and the correspondence variables $\mathbf{c}_t$. The algorithm outputs the mean μ_t and covariance $\mathbf{\Sigma}_t$ of the current robot pose belief, and the probability of the measurement set $p_{\mathbf{z}_t}$.

1	**EKFLocalization**$(\mu_{t-1}, \Sigma_{t-1}, \mathbf{u}_t, \mathbf{z}_t, \mathbf{m}, \mathbf{c}_t)$
	Prediction Step Section 4.1
	Jacobians of Linearized Motion Model $\mathbf{G}_t, \mathbf{V}_t$
2	$\mathbf{G}_t = \begin{pmatrix} 1 & 0 & -\frac{v_t}{w_t}\cos(\theta) + \frac{v_t}{w_t}\cos(\theta + w_t\Delta t) \\ 0 & 1 & -\frac{v_t}{w_t}\sin(\theta) + \frac{v_t}{w_t}\sin(\theta + w_t\Delta t) \\ 0 & 1 & 1 \end{pmatrix}$
3	$\mathbf{V}_t = \begin{pmatrix} -\frac{\sin(\theta)+\sin(\theta+w_t\Delta t)}{w_t} & \frac{v_t(\sin(\theta)-\sin(\theta+w_t\Delta t))}{w_t^2} + \frac{v_t\cos(\theta+w_t\Delta t)\Delta t}{w_t} \\ \frac{\cos(\theta)-\cos(\theta+w_t\Delta t)}{w_t} & -\frac{v_t(\cos(\theta)-\cos(\theta+w_t\Delta t))}{w_t^2} + \frac{v_t\sin(\theta+w_t\Delta t)\Delta t}{w_t} \\ 0 & \Delta t \end{pmatrix}$
	Motion Noise Covariance Matrix $\mathbf{M}_t$
4	$\mathbf{M}_t = \begin{pmatrix} \alpha_1 v_t^2 + \alpha_2 w_t^2 & 0 \\ 0 & \alpha_1 v_t^2 + \alpha_2 w_t^2 \end{pmatrix}$
	Prediction Update from Motion Model $\overline{\mu}_t, \overline{\Sigma}_t$
5	$\overline{\mu}_t = \mu_{t-1} + \begin{pmatrix} -\frac{v}{w}\sin(\theta) + \frac{v}{w}\sin(\theta + w\Delta t) \\ \frac{v}{w}\cos(\theta) - \frac{v}{w}\cos(\theta + w\Delta t) \\ w\Delta t \end{pmatrix}$
6	$\overline{\Sigma}_t = \mathbf{G}_t\Sigma_{t-1}\mathbf{G}_t^T + \mathbf{V}_t\mathbf{M}_t\mathbf{V}_t^T$
	Measurement Step Section 4.2
	Measurement Noise Covariance Matrix $\mathbf{Q}_t$
7	$\mathbf{Q}_t = \begin{pmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_s^2 \end{pmatrix}$
8	for $i = (1 \cdots N)$
	Predicted Measurement of ith Landmark $\hat{z}_t^i$
9	$q = (m_{j,x} - \overline{\mu}_{t,x})^2 + (m_{j,y} - \overline{\mu}_{t,y})^2$
10	$\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ atan2(m_{j,y} - \overline{\mu}_{t,y}, m_{j,x} - \overline{\mu}_{t,x}) - \overline{\mu}_{t,\theta} \\ m_{j,s} \end{pmatrix}$
	Jacobians of Linearized Measurement Model $\mathbf{H}_t^i$
11	$\mathbf{H}_t^i = \begin{pmatrix} -\frac{m_{j,x}-\overline{\mu}_{t,x}}{\sqrt{q}} & -\frac{m_{j,y}-\overline{\mu}_{t,y}}{\sqrt{q}} & 0 \\ \frac{m_{j,y}-\overline{\mu}_{t,y}}{q} & -\frac{m_{j,x}-\overline{\mu}_{t,x}}{q} & -1 \\ 0 & 0 & 0 \end{pmatrix}$
	Covariance of the Measurement $\mathbf{S}_t^i$
12	$\mathbf{S}_t^i = \mathbf{H}_t^i\overline{\Sigma}_t[\mathbf{H}_t^i]^T + \mathbf{Q}_t$
	Kalman Gain $\mathbf{K}_t^i$
13	$\mathbf{K}_t^i = \overline{\Sigma}_t[\mathbf{H}_t^i]^T[\mathbf{S}_t^i]^{-1}$
	Measurement Update from Measurement Model (LMMSE) μ_t, Σ_t
14	$\overline{\mu}_t = \overline{\mu}_t + \mathbf{K}_t^i(z_t^i - \hat{z}_t^i)$
15	$\overline{\Sigma}_t = (I - \mathbf{K}_t^i\mathbf{H}_t^i)\overline{\Sigma}_t$
16	endfor
17	$\mu_t = \overline{\mu_t}$
18	$\Sigma_t = \overline{\Sigma_t}$
19	$p_{\mathbf{z}_t} = \prod_{i=1}^{N} \frac{e^{-\frac{1}{2}(\mathbf{z}_t^i - \hat{\mathbf{z}}_t^i)^T[\mathbf{S}_t^i]^{-1}(\mathbf{z}_t^i - \hat{\mathbf{z}}_t^i)}}{(2\pi)^{\frac{N}{2}}\lvert\mathbf{S}_t^i\rvert^{\frac{1}{2}}}$
20	return $\mu_t, \Sigma_t, p_{\mathbf{z}_t}$

Table 6: The EKF Localization Algorithm with Known Landmark Correspondences

4.1 Prediction Step

Lines 2 to 6 of Table 3 encompass the prediction step of the EKF localization algorithm. The object of this step is to compute the parameters of our predicted (prior) belief $\overline{bel}(\mathbf{x}_t)$, the mean $\overline{\mu}_t$ and covariance $\overline{\mathbf{\Sigma}}_t$. The predicted mean $\overline{\mu}_t$ is computed in Line 5 according to the deterministic (noise-less) velocity motion model,

$$\overline{\mu}_t = \mu_{t-1} + \begin{pmatrix} -\frac{v}{w}\sin(\theta) + \frac{v}{w}\sin(\theta + w\Delta t) \\ \frac{v}{w}\cos(\theta) - \frac{v}{w}\cos(\theta + w\Delta t) \\ w\Delta t \end{pmatrix} \tag{29}$$

The predicted covariance $\overline{\mathbf{\Sigma}}_t$ is tricker to compute. $\overline{\mathbf{\Sigma}}_t$ is the sum of the two independent contributing sources of covariance, $\mathbf{\Sigma}_{t-1}$ and $\mathbf{M}_t$, after they have been transformed by their respective Jacobian matrices, $\mathbf{G}_t$, and $\mathbf{V}_t$.

$\mathbf{G}_t$ linearizes the velocity motion model $g(\mathbf{u}_t, \mu_{t-1})$ by deriving it with respect to the previous pose $\mathbf{x}_{t-1}$ and evaluating it at $\mathbf{u}_t$ and μ_{t-1}.

$$\begin{aligned} \mathbf{G}_t &= \frac{\partial g(\mathbf{u}_t, \mu_{t-1})}{\partial \mathbf{x}_{t-1}} \\ &= \begin{pmatrix} \frac{\partial x_t}{\partial \mu_{t-1,x}} & \frac{\partial x_t}{\partial \mu_{t-1,y}} & \frac{\partial x_t}{\partial \mu_{t-1,\theta}} \\ \frac{\partial y_t}{\partial \mu_{t-1,x}} & \frac{\partial y_t}{\partial \mu_{t-1,y}} & \frac{\partial y_t}{\partial \mu_{t-1,\theta}} \\ \frac{\partial \theta_t}{\partial \mu_{t-1,x}} & \frac{\partial \theta_t}{\partial \mu_{t-1,y}} & \frac{\partial \theta_t}{\partial \mu_{t-1,\theta}} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & -\frac{v_t}{w_t}\cos(\theta) + \frac{v_t}{w_t}\cos(\theta + w_t\Delta t) \\ 0 & 1 & -\frac{v_t}{w_t}\sin(\theta) + \frac{v_t}{w_t}\sin(\theta + w_t\Delta t) \\ 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

The previous posterior covariance $\mathbf{\Sigma}_{t-1}$ contributes $\mathbf{G}_t\mathbf{\Sigma}_{t-1}\mathbf{G}_t^T$ to the total predicted current belief covariance $\overline{\mathbf{\Sigma}}_t$. This represents a propagation of uncertainty across time from the *previous state space* to the *current state space*. The remaining covariance in $\overline{\mathbf{\Sigma}}_t$ originates from the covariance of the motion command parameters

$$\mathbf{M}_t = \begin{pmatrix} \alpha_1 v_t^2 + \alpha_2 w_t^2 & 0 \\ 0 & \alpha_1 v_t^2 + \alpha_2 w_t^2 \end{pmatrix}. \tag{30}$$

As with $\mathbf{\Sigma}_{t-1}$, $\mathbf{M}_t$ needs to be projected from from one space to another by a unique Jacobian matrix. $\mathbf{V}_t$ linearizes the velocity motion model $g(\mathbf{u}_t, \mu_{t-1})$ by deriving it with respect to the current motion command $\mathbf{u}_t$ and evaluating it at

$\mathbf{u}_t$ and μ_{t-1}.

$$\begin{aligned}
\mathbf{V}_t &= \frac{\partial g(\mathbf{u}_t, \mu_{t-1})}{\partial \mathbf{u}_t} \\
&= \begin{pmatrix} \frac{\partial x_t}{\partial v_t} & \frac{\partial x_t}{\partial w_t} \\ \frac{\partial y_t}{\partial v_t} & \frac{\partial y_t}{\partial w_t} \\ \frac{\partial \theta_t}{\partial v_t} & \frac{\partial \theta_t}{\partial w_t} \end{pmatrix} \\
&= \begin{pmatrix} -\frac{\sin(\theta)+\sin(\theta+w_t\Delta t)}{w_t} & \frac{v_t(\sin(\theta)-\sin(\theta+w_t\Delta t))}{w_t^2} + \frac{v_t\cos(\theta+w_t\Delta t)\Delta t}{w_t} \\ \frac{\cos(\theta)-\cos(\theta+w_t\Delta t)}{w_t} & -\frac{v_t(\cos(\theta)-\cos(\theta+w_t\Delta t))}{w_t^2} + \frac{v_t\sin(\theta+w_t\Delta t)\Delta t}{w_t} \\ 0 & \Delta t \end{pmatrix}
\end{aligned}$$

The motion covariance $\mathbf{M}_t$ contributes $\mathbf{V}_t\mathbf{M}_t\mathbf{V}_t^T$ to the total predicted current belief covariance $\overline{\mathbf{\Sigma}}_t$. This represents a propagation of uncertainty from *control space* to the *state space*.

Finally, in line 6, the total amount of covariance in $\overline{\mathbf{\Sigma}}_t$ is

$$\overline{\mathbf{\Sigma}}_t = \mathbf{G}_t\mathbf{\Sigma}_{t-1}\mathbf{G}_t^T + \mathbf{V}_t\mathbf{M}_t\mathbf{V}_t^T \tag{31}$$

That the total predicted covariance $\overline{\mathbf{\Sigma}}_t$ is the sum of the two (transformed) contributing sources is due to the independence between the current motion uncertainty and the previous pose uncertainty.

4.2 Measurement Correction Step

Lines 7 to 16 of Table 3 encompass the measurement step of the EKF localization algorithm. The object of this step is to compute the parameters of our posterior belief $bel(\mathbf{x}_t)$, the mean μ_t and covariance $\mathbf{\Sigma}_t$. Exploiting the conditional independence between each landmark measurement ($p(\mathbf{z}_t^i|\mathbf{x}_t) \perp p(\mathbf{z}_t^j|\mathbf{x}_t)$), the algorithm iteratively incorporates each measurement into the updated the posterior belief parameters. Regardless of the order in which the measurements are incorporated, the belief parameters μ_t and $\mathbf{\Sigma}_t$ will still converge to the optimal solution. This can be seen as a sequence of N uninterupted measurement updates before the next prediction update. Essentially, μ_t and $\mathbf{\Sigma}_t$ are generated via the with the Linear Minimum Mean Square Estimator (LMMSE). The LMMSE yields the optimal $bel(\mathbf{x}_t)$ because the robot pose $\mathbf{x}_t$ and measurement $\mathbf{z}_t$ are taken as jointly Gaussian.

Within each LMMSE iteration, the new necessary elements are the measurement covariance $\mathbf{Q}_t$, expected measurements $\hat{\mathbf{z}}_t$, and the measurement model Jacobian $\mathbf{H}_t^i$. The measurement covariance is

$$\mathbf{Q}_t = \begin{pmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & \sigma_s^2 \end{pmatrix} \tag{32}$$

The expected measurement of a landmark m_j (with known correspondence

to measurement z_t^i) is the output of the deterministic measurement model evaluated at $\overline{\mu}_t$,

$$\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ atan2(m_{j,y} - \overline{\mu}_{t,y}, m_{j,x} - \overline{\mu}_{t,x}) - \overline{\mu}_{t,\theta} \\ m_{j,s} \end{pmatrix}, \tag{33}$$

where $q = (m_{j,x} - \overline{\mu}_{t,x})^2 + (m_{j,y} - \overline{\mu}_{t,y})^2$.

$\mathbf{H}_t$ linearizes the feature measurement model $h(\overline{\mu}_t, j, \mathbf{m})$ by deriving it with respect to robot pose $\mathbf{x}_t$ and evaluating it at the predicted mean $\overline{\mu}_t$.

$$\begin{aligned} \mathbf{H}_t &= \frac{\partial h(\overline{\mu}_t, j, \mathbf{m})}{\partial \mathbf{x}_t} \\ &= \begin{pmatrix} \frac{\partial r_t^i}{\partial \overline{\mu}_{t,x}} & \frac{\partial r_t^i}{\partial \overline{\mu}_{t,y}} & \frac{\partial r_t^i}{\partial \overline{\mu}_{t,\theta}} \\ \frac{\partial \phi_t^i}{\partial \overline{\mu}_{t,x}} & \frac{\partial \phi_t^i}{\partial \overline{\mu}_{t,y}} & \frac{\partial \phi_t^i}{\partial \overline{\mu}_{t,\theta}} \\ \frac{\partial s_t^i}{\partial \overline{\mu}_{t,x}} & \frac{\partial s_t^i}{\partial \overline{\mu}_{t,y}} & \frac{\partial s_t^i}{\partial \overline{\mu}_{t,\theta}} \end{pmatrix} \\ &= \begin{pmatrix} -\frac{m_{j,x} - \overline{\mu}_{t,x}}{\sqrt{q}} & -\frac{m_{j,y} - \overline{\mu}_{t,y}}{\sqrt{q}} & 0 \\ \frac{m_{j,y} - \overline{\mu}_{t,y}}{q} & -\frac{m_{j,x} - \overline{\mu}_{t,x}}{q} & -1 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

The predicted pose covariance $\overline{\mathbf{\Sigma}}_t$ contributes $\mathbf{H}_t^i \overline{\mathbf{\Sigma}}_t [\mathbf{H}_t^i]^T$ to the total measurement covariance $\mathbf{S}_t^i$. This represents a propagation of uncertainty from *state space* to the *measurement space*. The measurement covariance $\mathbf{S}_t^i$ is

$$\mathbf{S}_t^i = \mathbf{H}_t^i \overline{\mathbf{\Sigma}}_t [\mathbf{H}_t^i]^T + \mathbf{Q}_t. \tag{34}$$

The Kalman Gain $\mathbf{K}_t^i$ is the ratio of the pose/measurement cross-covariance $\overline{\mathbf{\Sigma}}_t [\mathbf{H}_t^i]^T$ to the measurement covariance $\mathbf{S}_t^i$.

$$\mathbf{K}_t^i = \overline{\mathbf{\Sigma}}_t [\mathbf{H}_t^i]^T [\mathbf{S}_t^i]^{-1}. \tag{35}$$

The culminating measurement update equations are

$$\mu_t = \overline{\mu}_t + \mathbf{K}_t^i (z_t^i - \hat{z}_t^i) \tag{36}$$

and

$$\mathbf{\Sigma}_t = (I - \mathbf{K}_t^i \mathbf{H}_t^i) \overline{\mathbf{\Sigma}}_t, \tag{37}$$

where μ_t and $\mathbf{\Sigma}_t$ are the defining parameters of the optimal posterior distribution $bel(\mathbf{x}_t)$. Here, the mean μ_t is the sum of the predicted mean $\overline{\mu}_t$ and *measurement correction* $\mathbf{K}_t^i (z_t^i - \hat{z}_t^i)$. The measurment correction is the distance between the measured and predicted measurement (the innovation factor) transformed by the Kalman Gain from measurement space to state space. As a thought experiment, consider that the measurement $\mathbf{z}_t$ and pose $\mathbf{x}_t$ were entirely *un*correlated. Then one would glean nothing from the difference between the

acquired measurement and the predicted measurement. Consequently, the numerator of the Kalman Gain would be zero and the EKF would simply produce the prediction step outputs $\overline{\mu}_t$ and $\overline{\mathbf{\Sigma}}_t$. Conversely, if the measurement and state were *highly* correlated, and the measurement covariance relatively low, innovation factors $(z_t^i - \hat{z}_t^i)$ would be critically informative. The Kalman Gain then increases, and $\mu_t, \mathbf{\Sigma}_t$ are heavily influenced by the innovation factors.

Finally, line 19 computes the joint probability of the measurement set $\mathbf{z}_t$, which is used to reject statistical outliers. The conditional independence between each landmark measurement ensures that the joint probability $p_{\mathbf{z}_t}$ is the product of the probability of each induvidual measurement.

$$p_{\mathbf{z}_t} = \prod_{i=1}^{N} \frac{e^{-\frac{1}{2}(\mathbf{z}_t^i - \hat{\mathbf{z}}_t^i)^T [\mathbf{S}_t^i]^{-1} (\mathbf{z}_t^i - \hat{\mathbf{z}}_t^i)}}{(2\pi)^{\frac{N}{2}} |\mathbf{S}_t^i|^{\frac{1}{2}}}. \tag{38}$$

Rejection of outlier measurement sets can substantially improve Bayesian filter performance.

5 Simulation Results

5.1 Scenario Design

Given the definitions of the belief $bel(\mathbf{x}_t)$, measurement model $p(\mathbf{z}_t|\mathbf{x}_t)$, and motion model $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$, we have to specify a multitude of parameters governing the robot/environment system. These include the number and locations of landmarks $\mathbf{M}$, the measurement error parameters $\sigma_r^2, \sigma_\phi^2$, the applied motion commands v, w, and the motion error parameters $\alpha_1 \cdots \alpha 6$.

The goal of these simulations is to learn how these parameters influence the EKF localization performance. As such, the simulations were broken into three categories.

- Simulations which modulate the *motion error parameters* $\alpha_1 \cdots \alpha 4$, and leave the remaining parameters constant.
- Simulations which modulate the *measurement error parameters* $\sigma_r^2, \sigma_\phi^2$, and leave the remaining parameters constant.
- Simulations which modulate the *map configuration* $\mathbf{M}$, and leave the remaining parameters constant.

Table 7 describes all of the parameters represented in the simulations, including those which were not probed for analysis (no range of values).

	Constant	**Range**
Time Step	$Ts = 0.1s$	
Motion Commands	$v = 2\text{m/s}$ $w = 0.2\text{rad/s}$	
Motion Error	$\alpha_{1,2} = 0.5$ $\alpha_{3,4} = 0.5$ $\alpha_{5,6} = 0.5$	$\alpha_{1,2} = 0.1 \cdots 5.0$ $\alpha_{3,4} = 0.1 \cdots 5.0$
Measurement Error	$\sigma_R^2 = 0.5$ $\sigma_\phi^2 = 0.05$	$\sigma_R^2 = 0.1 \cdots 5.0$ $\sigma_\phi^2 = 0.01 \cdots 1.0$
Map Configuration	$N = 10$ $R = 50\text{m}$	$N = 3 \cdots 15$ $R = 5 \cdots 100\text{m}$

Table 7: Selected Simulation Parameters

The environment consisted of N landmarks uniformly spaced along the perimeter of a circle of radius R centered on the origin. Figure 4 illustrates three example maps of different N, R configurations.

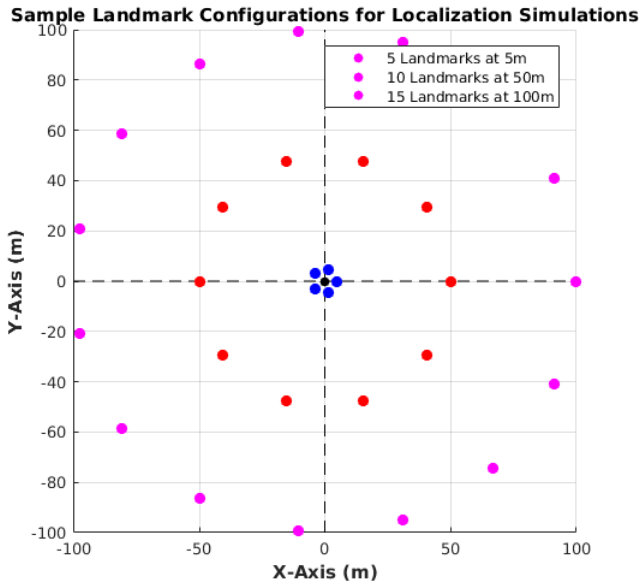

Figure 4: Sample Landmark Configurations for Localization Simulations

Within the map, the robot initializes at the global origin $\mathbf{x}_0 = (0, 0, 0)^T$ with the following initial belief distribution

$$\begin{aligned} \mu_0 &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\ \mathbf{\Sigma}_0 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

The performance of the localization was measured according to the position error

$$\epsilon_{x,y} = \sqrt{(\hat{x} - x)^2 + (\hat{y} - y)^2} \tag{39}$$

and angular error

$$\epsilon_\theta = \left| \hat{\theta} - \theta \right|. \tag{40}$$

5.2 Effects of Motion Error on Performance

This section describes how localization performance was impacted by various amounts of tranlational error $\alpha_{1,2}$ and rotational error $\alpha_{3,4}$ parameters. As shown in Figure 2, motion error parameters $\alpha_{0...6}$ cause the robot trajectory to deviate from a deterministically circular path. It was expected that localization performance would deteriorate as these parameters increased. In particular, it was expected that position error $\epsilon_{x,y}$ would be influenced by tranlational error parameters $\alpha_{1,2}$ and angular error ϵ_θ would be influenced by rotational error paramters $\alpha_{3,4}$. The two pairs of motion error parameteters $\alpha_{1,2}$ and $\alpha_{3,4}$ were re-sampled across the range of values specified in the Table 7 *Range* column. The remaining simulation parameters were held constant at the values specified in the Table 7 *Constant* column.

The following four figures show the results of 10 trials from four motion error configurations on the extreme ends as described in Table 8.

	Low Rotation Error $\alpha_{3,4} = 0.1$	**High Rotation Error** $\alpha_{3,4} = 0.5$
Low Translation Error $\alpha_{1,2} = 0.1$	Figure 5	Figure 7
High Translation Error $\alpha_{1,2} = 5.0$	Figure 6	Figure 8

Table 8: Overview of Motion Error Parameters for Figures 5 to 8

The lefthand panes show the true positions (dashed black) and estimated positions (solid green) of each trial. The upper-righthand panes show each $\epsilon_{x,y}$ versus time, and the lower-righthand panes show each ϵ_θ versus time.

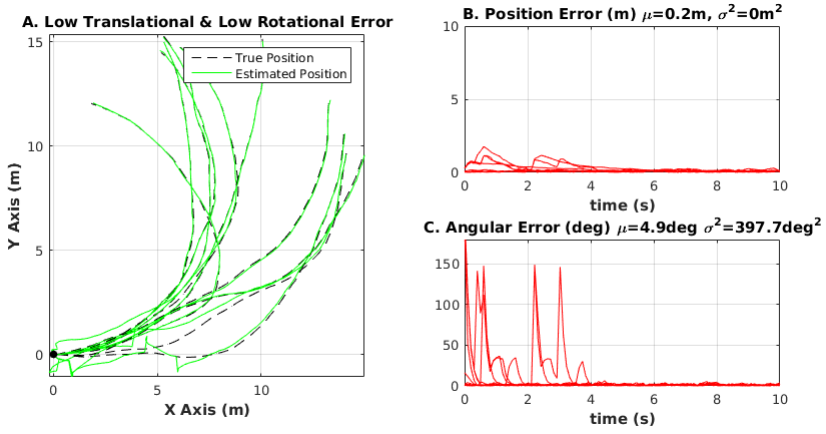

Figure 5: Localization Performance with Low Translational and Low Angular Motion Error

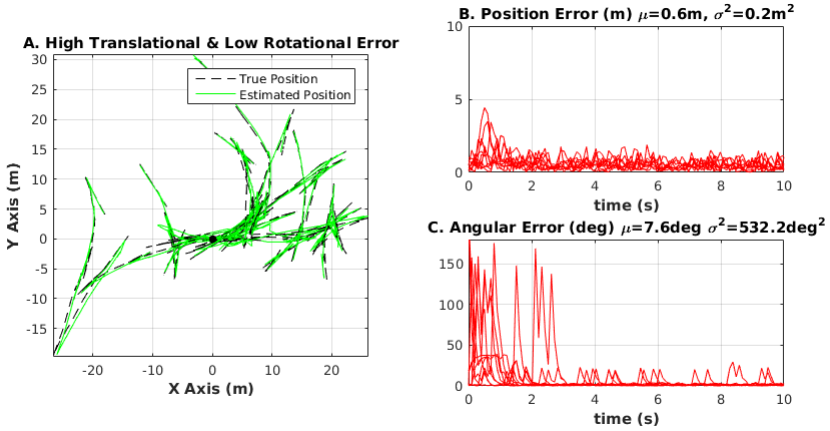

Figure 6: Localization Performance with High Translational and Low Angular Motion Error

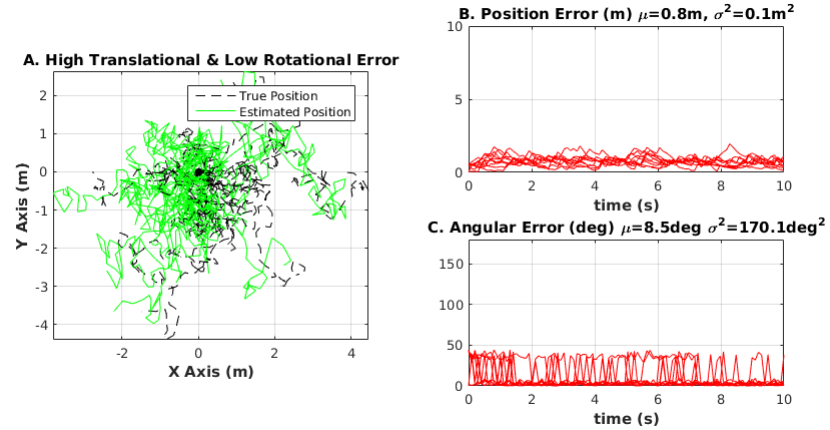

Figure 7: Localization Performance with Low Translational and High Angular Motion Error

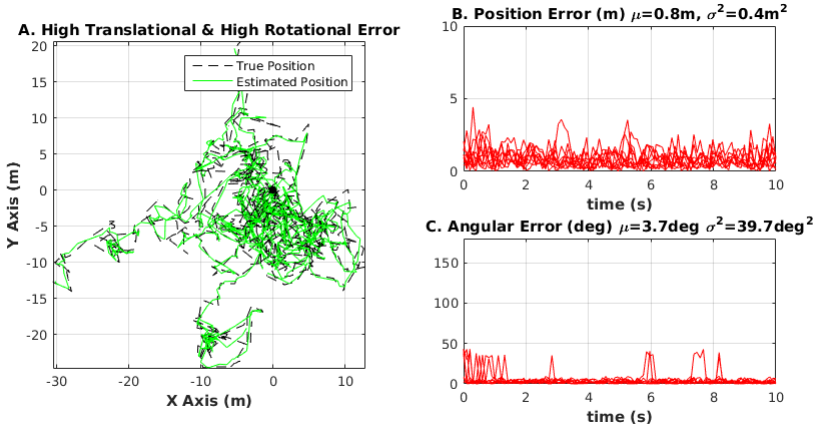

Figure 8: Localization Performance with High Translational and High Angular Motion Error

The statistical nature of the paths in each figure are unique because the motion error parameters for each figure are unique. Figure 5A shows nearly identically circular paths. Figure 6A shows straight paths which turn abruptly into other straight paths. Figure 7A shows paths which are only turning, but traveling only few meters. Figure 8A shows paths which are identical in turning behavior to those of Figure 7A, but travel a greater number of meters (resembling the paths of Figures 5A, 6A). All of these observed path characteristics are consistent with our intuition.

However, it is the performance of the localization operations that is of primary interest. As expected, $\epsilon_{x,y}$ was the least in Figure 5, with the lowest motion error $alpha_{1\ldots4}$. Unexpectedly, when there is low rotational motion error (Figure 5A, 6A), there is clear increase in ϵ_θ during the first three seconds of most trials.

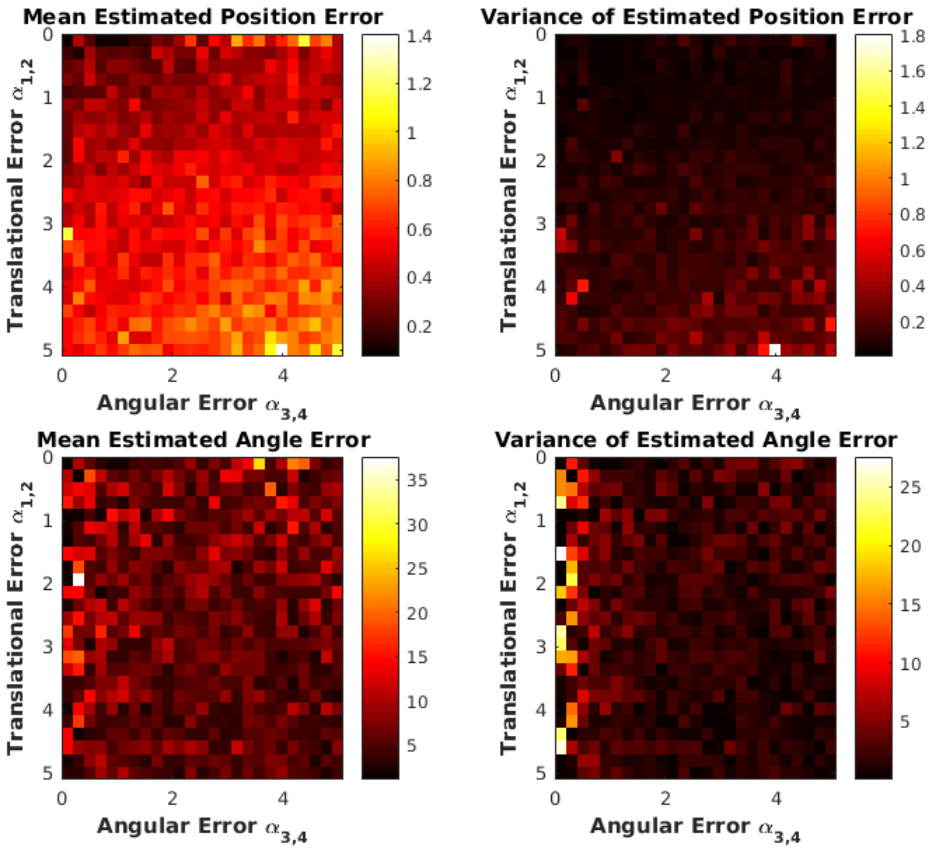

Figure 9: Mean and Variance of Localization Errors over Motion Error Parameters

Next, the localization performance parameters $\epsilon_{x,y}$ and ϵ_θ were iteratively sampled from simulations under $\alpha_{1,2} = (0.1 \cdots 5.0)$ and $\alpha_{3,4} = (0.1 \cdots 5.0)$. The mean and variance of both errors from each simulation are shown in Figure 9. As expected the mean and variance of $\epsilon_{x,y}$ grow steadily as a function of $\alpha_{1,2}$. The counterintuitive results of Figures 5C and 6C are reenforced by Figure 9; the expected ϵ_θ *increases* for lower rotational error $\alpha_{3,4}$. This behavior appears across the full range of $\alpha_{1,2}$.

5.3 Effects of Measurement Error on Performance

This section describes how localization performance was impacted by various amounts of measurement error, σ_R^2 and $\alpha_{3,4}$. As shown in Figure 3, sensor measurement error causes landmark range and bearing estimates to deviate from the true values. It was expected that localization performance would deteriorate as these parameters increased. In particular, it was expected that position error $\epsilon_{x,y}$ would be influenced by range variance σ_R^2 and angular error ϵ_θ would be influenced by bearing variance σ_ϕ^2. The two variances σ_R^2 and $\alpha_{3,4}$ were sampled across the range of values specified in the Table 7 *Range* column. The remaining simulation parameters were held constant at the values specified in the Table 7 *Constant* column.

The following four figures show the results of 10 trials from four measurement error configurations on the extreme ends as described in Table 9.

	Low Angle Error $\sigma_\phi^2 = 0.01$	**High Angle Error** $\sigma_\phi^2 = 1.0$
Low Range Error $\sigma_R^2 = 0.1$	Figure 10	Figure 12
High Range Error $\sigma_R^2 = 5.0$	Figure 11	Figure 13

Table 9: Overview of Measurement Error Parameters for Figures 10 to 14

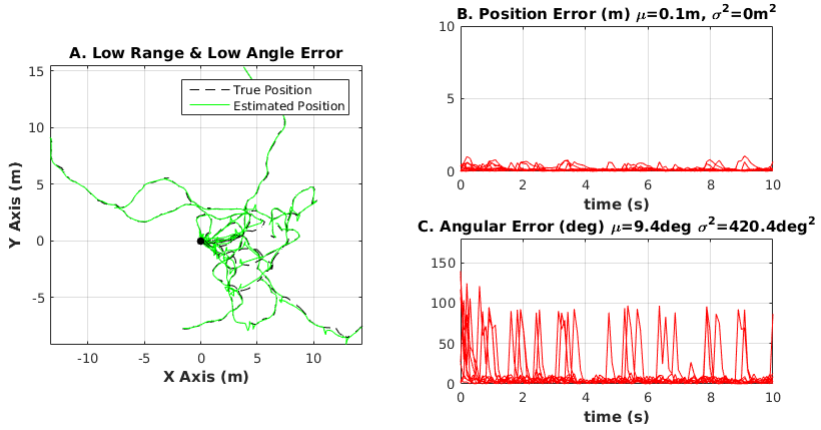

Figure 10: Localization Performance with Low Range and Low Angular Measurement Error

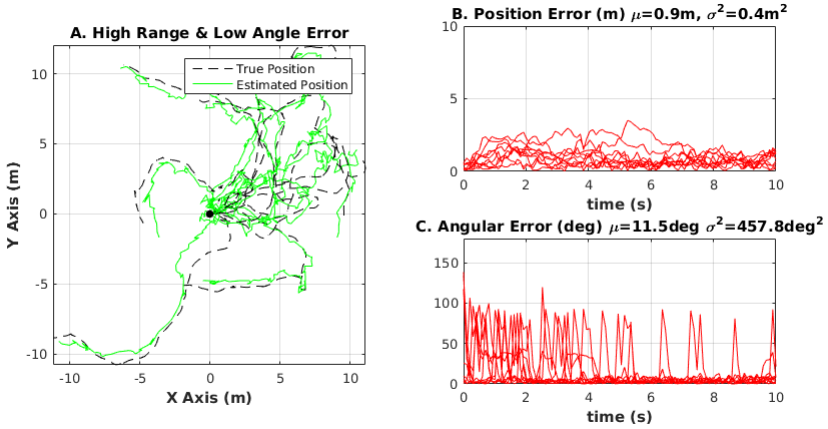

Figure 11: Localization Performance with High Range and Low Angular Measurement Error

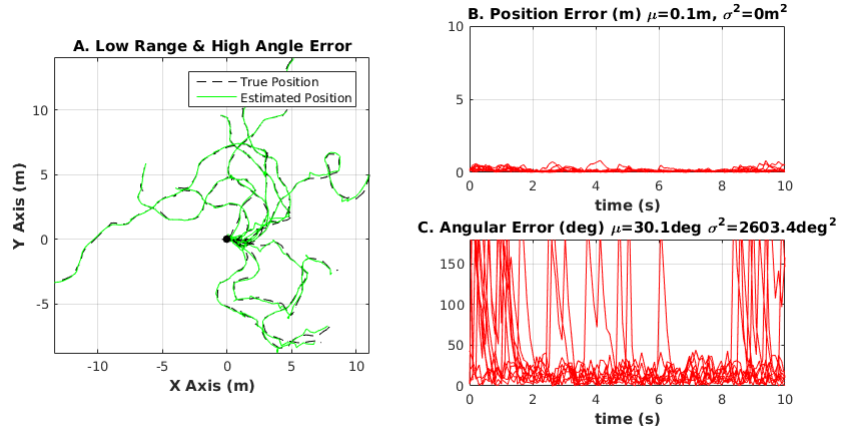

Figure 12: Localization Performance with Low Range and High Angular Measurement Error

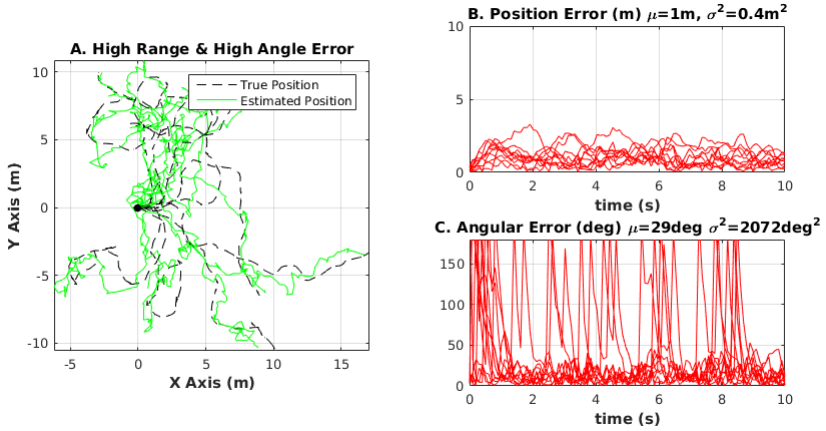

Figure 13: Localization Performance with High Range and High Angular Measurement Error

The statistical nature of the paths in each figure are identical because the motion error paramters $\alpha_{1\ldots6}$ were held constant throughout. As expected, $\epsilon_{x,y}$ were small in Figures 10 and 12, which had the low σ_R^2, and were high in Figures 11 and 13, which had high σ_R^2. Similarly, ϵ_θ were (relatively) small in Figures 10 and 11, which had the low σ_ϕ^2, and were high in Figures 12 and 13, which had high σ_ϕ^2. By comparing these figures, it appears that $\epsilon_{x,y}$ and ϵ_θ can change with σ_R^2 and σ_ϕ^2, respectively, quite independently of each other.

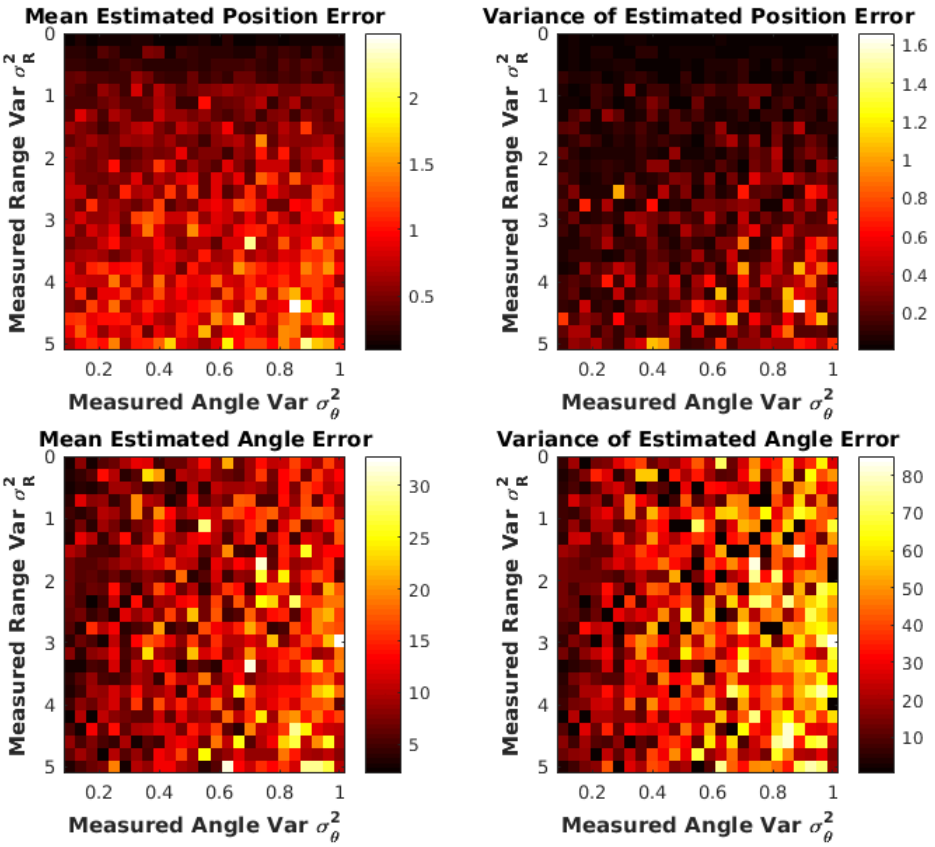

Figure 14: Mean and Variance of Localization Errors over Measurement Error Parameters

Next, $\epsilon_{x,y}$ and ϵ_θ were iteratively sampled from simulations under $\sigma_R^2 = (0.1 \cdots 5.0)$ and $\sigma_\phi^2 = (0.01 \cdots 1.0)$. The mean and variance of both errors from each simulation are shown in Figure 9. As expected the mean and variance of $\epsilon_{x,y}$ grow steadily as a function of σ_R^2 (and σ_ϕ^2 to a lesser extent). Similarly, the mean and variance of ϵ_θ grow steadily as a function of σ_ϕ^2.

5.4 Effects of Map Configuration on Performance

This section describes how localization performance was impacted by various map configurations. As described in Section 5.1, a map is defined as N landmarks uniformly spaced along the perimeter of a circle of radius R centered on the origin. It was expected that localization performance would deteriorate as the number of landmarks reduced and their ranges to the origin increased (having seen that measured landmark bearing worsens with higher ranges). The two parameters, number of landmarks N and radial range R were sampled across the range of values specified in the Table 7 *Range* column. The remaining simulation parameters were held constant at the values specified in the Table 7 *Constant* column.

The following four figures show the results of 10 trials from four map configurations on the extreme ends as described in Table 10.

	Short Range $R = 5m$	**Long Range** $R = 100m$
Few Landmarks $N = 3$	Figure 15	Figure 17
Many Landmarks $N = 15$	Figure 16	Figure 18

Table 10: Overview of Map Configurations for Figures 15 to 18

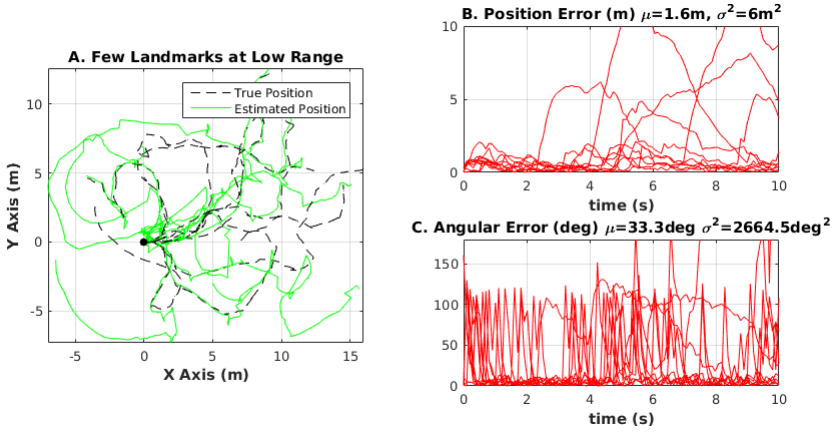

Figure 15: Localization Performance with Few Landmarks (3) at Low Range (5m)

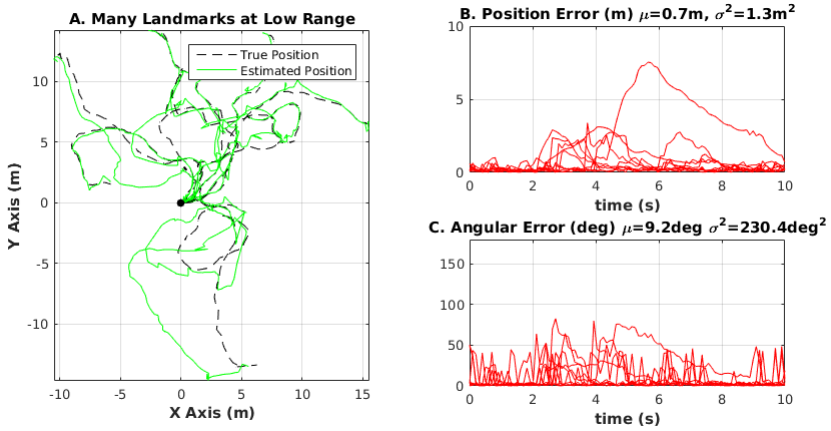

Figure 16: Localization Performance with Many Landmarks (15) at Low Range (5m)

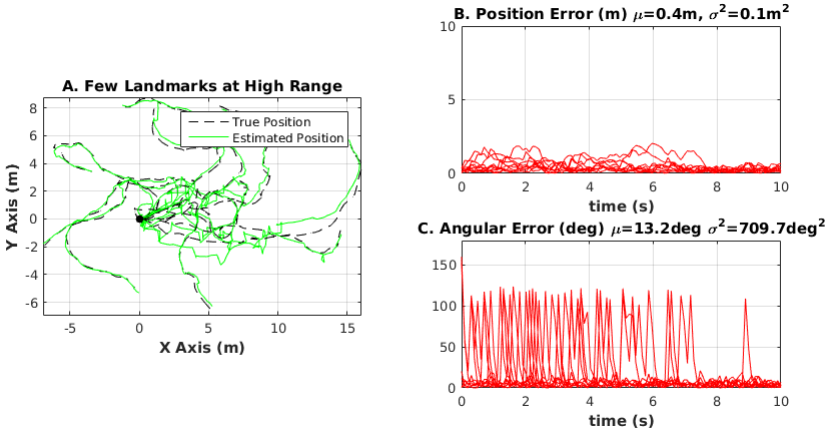

Figure 17: Localization Performance with Few Landmarks (15) at High Range (5m)

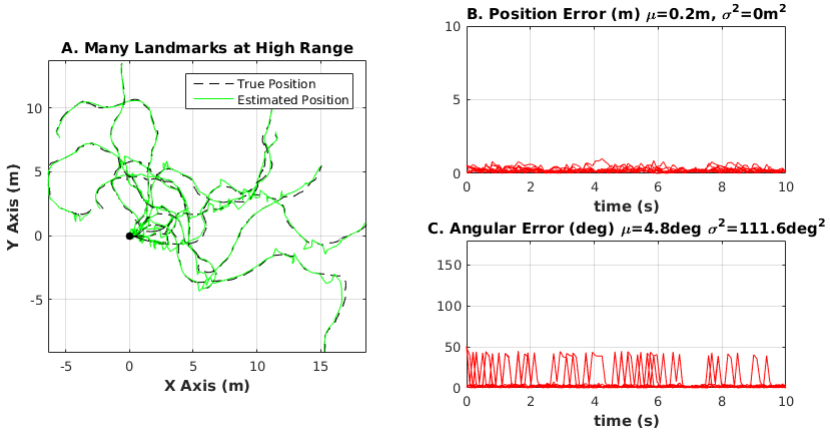

Figure 18: Localization Performance with Many Landmarks (15) at High Range (100m)

The $\epsilon_{x,y}$ were small in Figures 17 and 18, which had the high landmark ranges R, and were high in Figures 15 and 16, which had low landmark ranges R. Similarly, ϵ_θ were (relatively) small in Figures 16 and 18, which had the many landmarks N, and were high in Figures 15 and 17, which had few landmarks N. As such, the best performance was achieved in Figure 18, with many landmarks N at high range R.

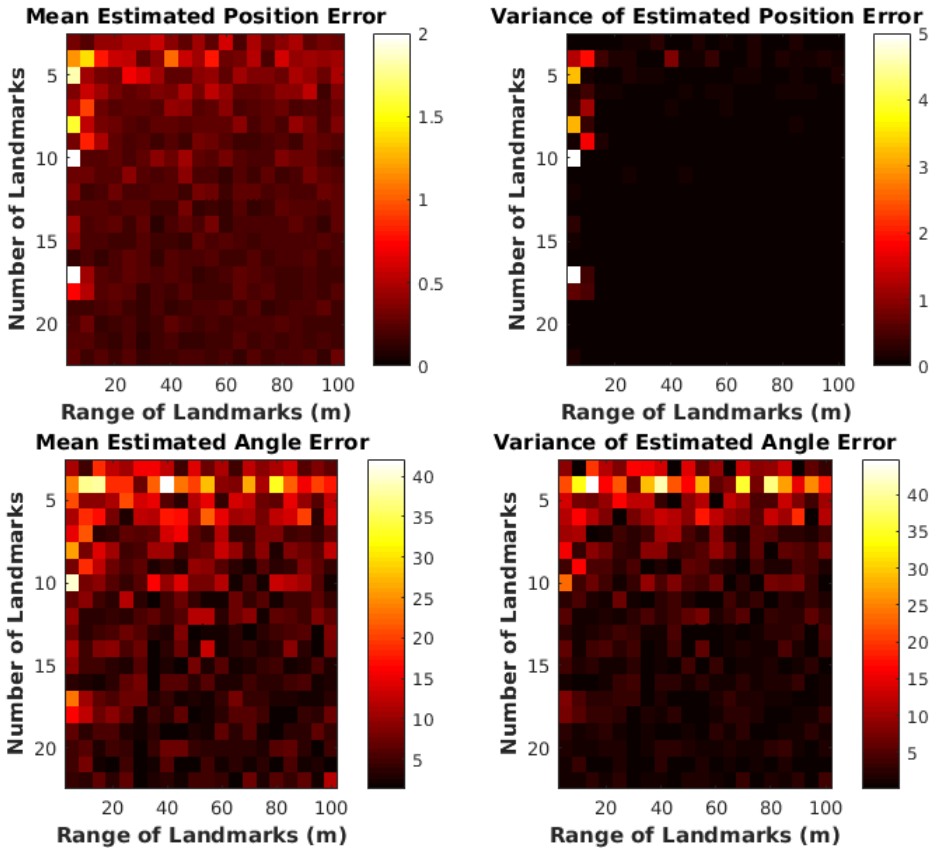

Figure 19: Mean and Variance of Localization Errors over Map Configurations

Next, $\epsilon_{x,y}$ and ϵ_θ were iteratively sampled from simulations under $N = (3 \cdots 15)$ and $R = (5.0 \cdots 100.0)$m. The mean and variance of both errors from each simulation are shown in Figure 19. As suggested by Figure 15, $\epsilon_{x,y}$ and ϵ_θ are the greatest for few landmarks at close range. The range of the landmarks R heavily influence position error $\epsilon_{x,y}$ (for $N < 10$). The number of the landmarks N heavily influence angular error ϵ_θ. Moreover, the *thresholds* for what constitutes too few landmarks and too close a range appear clear. Landmark ranges R must be greater than 10m, but stop yielding improvements after 20m. The number of landmarks must be greater than 6, but stop yielding improvements after 6.

6 Conclusion

The purpose of this paper was to develop and demonstrate the capability of planar robot localization by leveraging the Extended Kalman Filter (EKF) algorithm. The paper began by expressing the robot perception problem in terms of *bayesian* (i.e. optimal) *estimation*. The framework of recursive bayesian estimation made clear from the start that beliefs are represented with probability distribution functions (PDF). Therefore, the viability of a particular estimation algorithm rested on the feasibility of representing and transforming the adopted belief PDFs. The earliest implementations of Bayesian filters assumed that state and measurements are jointly *Gaussian*. This is because properties of Gaussian random variables, such as the preservation of Gaussianity under linear transformation and the conformity of real-world phenomenon to Gaussianity, justify it's usagage.

A Bayesian filter for which the prior belief is a moments-parametrized Gaussian, and the process models are linear Gaussian systems is known as the *Kalman Filter*. The fact that most real-world state transition or measurement models would be non-linear in their arguments necessitated an augmentation of the Kalman filter. The EKF replaces the linear operations for computing predicted state and predicted measurements with the corresponding non-linear operations. In addition, the transformation matrices with which covariances (uncertainty) are estimated are the Jacobians of the original non-linear transformations. The applicability linearized transformations is limited by the uncertainty of inputs and the local non-linearity of the transformations. Nonetheless, the EKF is one of the most popular stochastic state estimation algorithms employed in robotics.

The performance of localization was measured according to the position error $\epsilon_{x,y}$ and angular error ϵ_θ. The purpose of the simulations was to learn how the multitude of input parameters relatively influence the overall localization performance. For thoroughness, the simulations sampled across the translational and rotational motion error parameters ($\alpha_{1\ldots4}$), the range and bearing error parameters ($\sigma_R^2, \sigma_\phi^2$), and the map configuration (N, R). This large volume of results, for the most part, supported the hypothesized inuitive outcomes. The motion error test results, in Figure 9, showed that $\epsilon_{x,y}$ increased with $\alpha_{1\ldots4}$. However, ϵ_θ did not increase with with $\alpha_{1\ldots4}$ and was actually highest for low $\alpha_{3,4}$. The measurement test results, in Figure 14, showed that both $\epsilon_{x,y}$ increased with σ_R^2 and ϵ_θ increased with σ_ϕ^2. The map configuration test results, in Figur 19, showed that $\epsilon_{x,y}$ were high when landmarks were closer to the origin than 10m ($R < 10$m), but settled down immediatly for $R > 10$m. Conversely, ϵ_θ were high when the number of landmars were lower than 6 ($N < 6$) but settled down immediately for $N > 6$.

A significant assumption made by this paper was that the map of landmarks $\mathbf{M}$ was precisely known to the robot and that the correspondences to the locally observed landmarks were preciesly known. The anticipated follow on work for this project is to remove this a priori knowledge entirely and have the robot both localize itself *and* map the environment. The problem of simultaneous localization and mapping (SLAM) is critical robotics problem which also has an

EKF solution. Other possible follow on work could include the implementation of non-parameteric bayesian filters such as particl filters.

7 References

1. Mathworks. (2011). *Statistics and Machine Learning Toolbox (r2018a).* Retrieved April 15, 2018 from https://www.mathworks.com/help/stats/

2. Sarkka S (2013). *Bayesian Filtering and Smoothing.* Cambridge University Press.

3. Thrun S, Burgard W, Fox D. (2005). *Probabilistic Robotics.* Intelligent Robotics and Autonomous Agents. MIT Press.