Dual Kalman Filters for Autonomous Terrain Aided Navigation in Unknown Environments

Anindya Sankar Paul

Adaptive Systems Lab, Department of Computer Science and Electrical Engineering
OGI School of Science and Engineering, Oregon Health & Science University
20000 NW Walker Rd., Beaverton, OR 97006

ABSTRACT

In this paper, we address a method for terrain aided navigation of unmanned vehicles in unknown environments. The task is to simultaneously estimate the state of the vehicle (position and attitude) and a map of the surrounding environment given limited sensing capabilities. Possible available sensors include an on-board inertial measurement unit (IMU) and other simple “terrain sensors” such as altitude, temperature, or vision based features. Explicit positioning sensors such as GPS or a prior land map are not available. This problem is widely known as Simultaneous Localization and Mapping (SLAM). As no priori terrain information and initial knowledge of the vehicle location is assumed, the task of obtaining a solution to the SLAM problem is considered as extremely challenging. A dual Kalman filter framework is proposed that works by alternating between using one Kalman filter to localize the vehicle given the current estimated map, and a second Kalman filter to update the estimate of the map given the position of the vehicle. Simulation results are generated for a two dimensional environment comparing the Extended Kalman filter (EKF) and Sigma Point Kalman filter (SPKF) based implementations. The results show that it is possible for an autonomous vehicle to start in an unknown location in an unknown environment and using relative observations only, incrementally build a perfect map of the terrain and to compute simultaneously a bounded estimate of it’s location and direction. It is also shown that the SPKF based approach converges faster and to a lower Mean Square Error (MSE) than that of the EKF counterpart.

Keywords: Simultaneous localization and mapping, terrain aided navigation, kalman filter, dual framework.
I. INTRODUCTION

The primary objective of any SLAM algorithm is to build a map of the environment and based on the map simultaneously estimate the state of the agent. The ability to place a vehicle at an unknown location in an unknown environment and then have it simultaneously build a map using only relative observations and based on this map estimate the vehicle’s state makes a vehicle truly autonomous. For unmanned vehicle navigation, SLAM attempts to localize the vehicle without the need for GPS or other direct positioning information. Although GPS-aided navigation has long term stability with high accuracy and it has worldwide coverage in any weather condition, the main drawback is its dependency on external satellite signals which can be easily blocked or jammed by intentional interference. Thus, one of the major advantages of SLAM is that it eliminates the need for artificial infrastructures or a priori topological knowledge of the environment. A solution to the SLAM problem would be of inestimable value in a range of applications where absolute position and precise map information is unobtainable, such as autonomous planetary exploration, subsea deployment of autonomous vehicle and unmanned air vehicle navigation. In the last few years, the prospect of SLAM algorithms has attracted a great deal of attention in the mobile robotics and autonomous vehicle navigation communities [5,12,21,27,28,36]. SLAM was first addressed in the paper by Smith and Cheeseman [5] and has evolved from the work of indoor robotics research community to explore unknown environments [9,19,20,34,38]. The Australian Center for Field Robotics group headed by M.W.M. Gamini Dissanayake [8] showed that theoretically a solution to the SLAM problem is indeed possible.

There are two predominant approaches to the SLAM problem as seen in literature. One approach models the environment by estimating the location of the landmarks scattered throughout and based on this model estimates the vehicle’s state [7-8]. Dissanayake et al. [8] described a substantial implementation of the SLAM algorithm on a vehicle operating in an outdoor unknown environment using millimeter-wave (MMW) radar to provide relative range and bearing observations from certain landmarks. The other approach, called a Terrain Aided navigation system (TANS), builds an environmental map from the concurrent terrain information provided by its sensors and then generates state estimate based on it [16-17]. Key aspects of this approach include the choice of representation
for the map and the algorithms to perform the estimation or map building. The terrain contour matching (TERCOM) system, employed in cruise missile [2], combines onboard radar-altimeter readings with a pre-stored digitized terrain elevation (DTE) map in order to guide low-flying missiles maintain a fixed height above the ground. For both approaches, the prevailing method is to use the EKF to fuse the estimate of the state and the map [8,9,16]. The popularity of this approach is due to two main factors. First, it directly provides a recursive solution to the navigation problem and second, it computes consistent estimates for the uncertainty in vehicle and map based on vehicle kinematic model and relative sensor observations. Ayache and Faugeras [1] and Chatila and Laumond [4] used Kalman filter algorithms in visual navigation of mobile robots. Dissanayake et al. [8] used EKF to fuse vehicle and landmark location estimates using relative measurement between the vehicle and individual landmarks. J. Kim and S. Sukkarieh [20] implemented EKF based SLAM algorithm on unmanned aerial vehicle (UAV) navigation. Beside EKF, a lot of research has been performed on comparing alternative stochastic estimation techniques. Notable are the work by Thrun et al. [35-36] and Yamauchi et al. [39]. Thrun et al. used a bayesian approach to map building that does not assume Gaussian probability distributions as required by Kalman filter. Yamauchi et al. used an evidence grid approach where environment was decomposed to a number of cells. Kosecka et al. [17] developed a vision based SLAM algorithm based on constrained hidden markov model (HMM). These techniques while very effective for localization with respect to maps, do not lend themselves in providing incremental solution to SLAM, where a map is built gradually as information is received from sensors. A number of research teams have also tackled the SLAM problem using batch estimation techniques [22]. However, the major disadvantage in these techniques is that they cannot be operated in an online mode. The use of the Unscented Kalman filter (UKF) [14-15] to replace the EKF has also been proposed for unmanned aerial vehicle navigation [18]. Langelaan et al. [18] developed an UKF SLAM algorithm to navigate a small UAV through an unsurveyed environment, but the comparison between the UKF and EKF was performed only for the pure state estimation case (the map was assumed to be known almost perfectly).

This paper presents both an EKF and Sigma Point Kalman filter (SPKF) [23-25] based solution to the problem of simultaneous estimation of the state and map in terrain aided navigation. A dual Kalman filter framework [37] is used to learn both the hidden state and parameters of a terrain map. In this framework, two filters are run
concurrently. One filter estimates the state by taking as input the current estimate of the map parameters and the 
local observations from simple terrain sensors. The second filter estimates the map parameters by using the 
current estimate of the state and the local observations. In this study [30], we simulated a vehicle equipped with a 
low cost inertial measurement unit (IMU) and three simple terrain sensors. Each terrain sensor provides a noisy 
measurement of some “characteristic” of the environment at the current vehicle location. GPS position and 
velocity measurements were not incorporated in our algorithms. Simulation performances between EKF and 
SPKF based algorithms were compared for the following three cases: pure state estimation (terrain maps are 
known exactly), pure map estimation (vehicle position is known exactly), and finally dual estimation of the state 
and map. In all cases, the SPKF outperformed the EKF in both convergence time and lower MSE. This was 
expected as the vehicle kinematic model and the observation models are highly nonlinear in our simulation.

We begin in Section II with details of the dual Kalman Filter framework of state and parameter estimation. 
Section III underlines the principle of optimal recursive estimation and section IV highlights the EKF process and 
it’s flaws. In section V, the SPKF is introduced. Section VI-VII shows the EKF and SPKF estimators design and 
summarizes the simulation results. Finally, section VIII presents conclusion and future work.

II. DUAL KALMAN FRAMEWORK

The general state and parameter sequential dual estimation framework is illustrated in Fig. 1. As applied to the 
SLAM problem, the vehicle state and terrain model is estimated sequentially at each time step. State and model 
estimation are dependent and complimentary to each other. The terrain map parameters, \( \hat{w}_k \), are estimated at each 
time step given the current estimated state \( \hat{x}_k \) and also available noisy sensor measurements \( y_k \).

![Fig. 1: Sequential Dual Estimation Framework](image)
Similarly, the state estimate, \( \hat{x}_k \), is also performed by taking into account the estimated model parameters, \( \hat{w}_k \), and noisy measurements. For better understanding, the state and parameter estimation frameworks are briefly reviewed next before going into the details of dual framework.

### a. State Estimation

For a nonlinear state-space system with known map, the state estimation framework can be expressed as:

\[
x_{k+1} = f(x_k, u_k, v_k)
\]

\[
y_k = h(x_k, w, n_k)
\]

where \( x_k \) represent the unobserved state and \( y_k \) is the observed signal. \( f(\cdot) \) corresponds to known system dynamics (e.g., vehicle kinematics) and \( h(\cdot) \) corresponds to the observation function (e.g., terrain map) with known parameters \( w \). The state process noise \( v_k \) drives the dynamical system. Observation noise is given by \( n_k \) and \( u_k \) corresponds to any exogenous inputs (e.g., control commands). Note that, here noise sources are not considered as additive only. In state estimation, recursive filter, as EKF is the standard method of choice to achieve a recursive (approximate) maximum likelihood estimation of the state \( x_k \).

### b. Parameter Estimation

The parameter estimation framework with known state can be shown as:

\[
w_{k+1} = w_k + v_{w_k}
\]

\[
y_k = h(x_k, w_k) + n_k
\]

where input \( x_k \) and output \( y_k \) corresponds to the unobserved state and nonlinear observation. \( w_k \) corresponds to a stationary process with identity state transition matrix, driven by process noise \( v_{w_k} \) and \( h \) is the terrain map parameterized by \( w \). The nonlinear map, for example may be a gaussian mixture model (\( w \) are the mean and variances of each gaussian) or a feedforward neural network (\( w \) are the weights). The objective of the parameter estimation corresponds to learning the weights \( w \) in order to minimize the expected squared error between desired (\( d_k \)) and generated (\( y_k \)) outputs. While a number of optimization approaches exist (e.g. gradient descent using back propagation), the EKF can be applied as an efficient “second-order” technique for learning the parameters.
\( \textbf{w}_k \). In the linear case, the relationship between the Kalman filter (KF) and recursive least squares (RLS) is given in [33]. In the nonlinear case, the EKF training corresponds to a modified-Newton optimization method [13]. The use of the EKF for training neural networks has been developed by Singhal and Wu [34] and Feldkamp [31].

c. Dual Estimation

When both the states and map is unknown, a dual Kalman framework that is the combination of both state and parameter estimation frameworks is required. In this framework, two Kalman filters are run concurrently. At every time step, the state filter estimates the state using the current map parameter estimate, \( \hat{\textbf{w}}_k \) as shown in eqs (1-2), while the weight filter estimates the weights using the current state estimate, \( \hat{\textbf{x}}_k \), as in eqs (3-4). The discrete time non-linear dynamical system for dual estimation problem can be expressed as:

\[
\begin{align*}
\textbf{x}_{k+1} &= f(\textbf{x}_k, \textbf{u}_k, \textbf{v}_{x_k}) \quad (5) \\
\textbf{w}_{k+1} &= \textbf{w}_k + \textbf{v}_{w_k} \quad (6) \\
\textbf{y}_k &= h(\textbf{x}_k, \textbf{w}_k, \textbf{n}_k) \quad (7)
\end{align*}
\]

where both the system state \( \textbf{x}_k \) and the set of model parameters \( \textbf{w}_k \) for the dynamical system must be simultaneously estimated from only the observed noisy signal \( \textbf{y}_k \). A general theoretical and algorithmic framework for dual kalman based estimation has been presented in ch. 5 of [13] (though not for the specific SLAM application). In the next sections the optimal estimation framework and flaws of EKF is reviewed in state estimation context to help motivate the Sigma Point Kalman Filter (SPKF). Most of the background material covered in the next sections was the contribution of Wan and Van der Merwe [23-25,37].

III. OPTIMAL RECURSIVE ESTIMATION

The goal is to provide an optimal estimate of the system state \( \textbf{x}_k \) (or weight \( \textbf{w}_k \)) given a sequence of observations, \( \textbf{Y}_0^k = \{\textbf{y}_0, \textbf{y}_1, ..., \textbf{y}_k\} \). The optimal estimate of the state \( \textbf{x}_k \) in the minimum mean-squared error (MMSE) sense is given by the conditional mean:

\[
\hat{\textbf{x}}_k = \mathbf{E} [\textbf{x}_k | \textbf{Y}_0^k] = \int \textbf{x}_k p(\textbf{x}_k | \textbf{Y}_0^k) d\textbf{x}_k \quad (8)
\]
Evaluation of this expectation requires knowledge of a posteriori density \( p(x_k | Y^k_0) \). The problem of determining a posteriori density can be evaluated recursively using the Bayesian approach according to the following relations:

\[
p(x_k | Y^k_0) = \frac{p(y_k | x_k)p(x_k | Y^{k-1}_0)}{p(y_k | Y^{k-1}_0)} \tag{9}
\]

where the prior is defined as:

\[
p(x_k | Y^{k-1}_0) = \int p(x_k | x_{k-1})p(x_{k-1} | Y^{k-1}_0)dx_{k-1} \tag{10}
\]

and the normalizing constant is given by:

\[
p(y_k | Y^{k-1}_0) = \int p(y_k | x_k)p(x_k | Y^{k-1}_0)dx_k \tag{11}
\]

This recursion specifies the current state density as a function of the previous density and the most recent measurement data. The kinematics or dynamics of the vehicle comes into play by specifying the state-transition probability \( p(x_k | x_{k-1}) \). The observation density \( p(y_k | x_k) \) represents the “observation likelihood”. Knowledge of these densities and the initial condition \( p(x_0 | y_0) = p(y_0 | x_0)p(x_0) / p(y_0) \) determines \( p(x_k | Y^k_0) \) for all \( k \).

Unfortunately, the multi-dimensional integration indicated by Eqs. (9-11) makes a closed form solution intractable for most systems. However, under the assumption that state and noise distributions are gaussian and system process and measurement models are linear, the Kalman Filter (KF) is optimal for the recursive propagation of the estimate of the system state \( x_k \). For nonlinear system dynamics, a first-order approximation of the state space models to account for nonlinearities leads to the Extended Kalman Filter (EKF), which is the current industry standard and most widely used algorithm.

**IV. THE EKF AND ITS FLAWS**

Extended Kalman Filter (EKF) has been exhaustively used as a standard technique for performing recursive nonlinear estimation. The basic state space estimation framework of a discrete-time nonlinear dynamic system can be represented as:
\[ x_{k+1} = f(x_k, u_k, v_k) \]  
\[ y_k = h(x_k, n_k) \]

Given the noisy observation \( y_k \), the recursive estimation for \( x_k \) can be expressed in the form:

\[ \hat{x}_k = (\text{optimal prediction of } x_k) + \kappa_k \{ y_k - (\text{optimal prediction of } y_k) \} \]  \hspace{1cm} (14)

This recursion provides the optimal minimum mean-squared error estimate for \( x_k \) assuming all relevant random variables in the system can be efficiently and consistently modeled by maintaining their first and second order moments, i.e., they can be accurately modeled as Gaussian random variables (GRVs). We need not to assume linearity of the model. The optimal terms in this recursion are given by

\[ \hat{x}_k = \mathbb{E}[f(\hat{x}_{k-1}, v_{k-1})] \]  \hspace{1cm} (15)
\[ \kappa_k = P_{x,y_k} P_{y_k}^{-1} \]  \hspace{1cm} (16)
\[ \hat{y}_k = \mathbb{E}[h(\hat{x}_k, n_k)] \]  \hspace{1cm} (17)

where the optimal prediction of \( x_k \) is written as \( \hat{x}_k^{-} \), and corresponds to the expectation of a nonlinear function \( f(.) \) of the random variables \( \hat{x}_{k-1} \) and \( v_{k-1} \). Similarly, the optimal prediction of \( y_k \) is denoted as \( \hat{y}_k^{-} \), and corresponds to the expectation of a nonlinear function \( h(.) \) of the random variables \( \hat{x}_k^{-} \) and \( n_k \). The optimal gain term \( \kappa_k \) is expressed as a function of posterior covariance matrices (with \( \hat{y}_k = y_k - \hat{y}_k \)). Note these terms also require taking expectations of a nonlinear function of the prior state estimates. The Kalman filter calculates these quantities exactly in the linear case, and can be viewed as an optimal method for analytically propagating a GRV through linear system dynamics. For nonlinear models, however, the EKF approximates the optimal terms as:

\[ \hat{x}_k^{-} = f(\hat{x}_{k-1}, u_{k-1}, v) \]  \hspace{1cm} (18)
\[ \kappa_k = \tilde{P}_{x,y_k}^{\text{lin}} \left( \tilde{P}_{y_k}^{\text{lin}} \right)^{-1} \]  \hspace{1cm} (19)
\[ \hat{y}_k^{-} = h(\hat{x}_k^{-}, \bar{n}) \]  \hspace{1cm} (20)

where predictions are approximated as simply the function of the prior mean value for estimates (no expectation taken). Furthermore, the covariances \( \tilde{P}_{x,y_k}^{\text{lin}} \) and \( \tilde{P}_{y_k}^{\text{lin}} \) are determined by linearizing the system model around the current estimate of the state and then determining (approximating) the posterior covariance matrices analytically for the linear system (see [23] for exact equations).
Table 1: EKF equations

Initialization:

\[
\hat{x}_0 = E[x_0], \quad P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]
\]

For \( k = 1, \ldots, \infty \)

1. Time-update equations:

\[
\begin{align*}
\hat{x}_k^- &= f(\hat{x}_{k-1}, u_{k-1}, \bar{v}) \\
P_{k^-} &= A_{k-1} P_{x_{k-1}} A_{k-1}^T + B_{k} R^{k} B_{k}^T
\end{align*}
\]

2. Measurement update equations:

\[
\begin{align*}
K_k &= P_{k^-} C_k^T (C_k P_{k^-} C_k^T + D_k R^k D_k^T)^{-1} \\
\hat{x}_k &= \hat{x}_k^- + K_k [y_k - h(\hat{x}_k^-, \bar{n})] \\
P_{x_k} &= (I - K_k C_k) P_{k^-}
\end{align*}
\]

where

\[
\begin{align*}
A_k &= \left. \frac{\partial f(x, u, \bar{v})}{\partial x} \right|_{x = x_k^-} \\
B_k &= \left. \frac{\partial f(x, u, \bar{v})}{\partial v} \right|_{x = x_k^-} \\
C_k &= \left. \frac{\partial h(x, \bar{n})}{\partial x} \right|_{x = x_k^-} \\
D_k &= \left. \frac{\partial h(x, \bar{n})}{\partial n} \right|_{x = x_k^-}
\end{align*}
\]

and where \( R^k \) = process noise covariance and \( R^n \) = measurement noise covariance

Assumption:

The noise means denoted by \( \bar{n} = E[n] \) and \( \bar{v} = E[v] \) are assumed to equal to zero

In other words, in the EKF the state distribution is approximated by a GRV which is then propagated analytically through the first order linearization of the nonlinear system. The equations for EKF are shown in table 1 for state filter. As such, the EKF can be viewed as providing “first-order” approximations to the optimal terms in Eq. (14). These approximations used in the EKF can result in large errors in the true posterior mean and covariance of the transformed (Gaussian) random variable, which may lead suboptimal performance and sometimes even divergence of the filter. It is these flaws which was amended in the design of Sigma Point Kalman Filter (SPKF).

V. THE SIGMA POINT KALMAN FILTER (SPKF) APPROACH

Julier & Uhlmann [14-15] derived a novel, more accurate and theoretically better grounded alternative to the EKF called the Unscented Kalman Filter (UKF) for state estimation within the application domain of nonlinear control. The UKF, and another closely related algorithm, the Central Difference Kalman Filter (CDKF) [29] has been
extended within the general field of probabilistic inference to parameter and dual estimation [13,37]. Computationally efficient *square-root* versions of these algorithms that are ideally suited for numerically robust real-time implementation [23,25] has also been researched. All these algorithms are collectively referred as *Sigma-Point Kalman Filters* (SPKF) and have shown that they consistently have equal or better performance than the EKF on the full range of estimation problems.

The *Sigma-Point Kalman filter* (SPKF) addresses the approximation issues of the EKF through a fundamentally different approach for calculating the posterior 1st and 2nd order statistics of a random variable that undergoes a nonlinear transformation. The state distribution is again represented by a GRV, but is now specified using a minimal set of deterministically chosen weighted sample points, called *sigma-points*. These sigma points completely capture the true mean and covariance of the GRV, and when propagated through the true nonlinear system, capture the posterior mean and covariance accurately to the 3rd order (Taylor series expansion) for any nonlinearity. In the next subsections, a detailed description of Unscented Transform (UT) and the corresponding SPKF is provided.

### a. Unscented Transformation (UT)

The Unscented Transformation (UT) is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation [15]. To be more specific, let us consider propagating a random variable \( x \in \mathbb{R}^l \) through an arbitrary nonlinear function \( y = g(x) \). Assume \( x \) has mean \( \bar{x} \) and covariance \( P_x \). To calculate the statistics of \( y \) we form a matrix \( \chi \) of \( 2L + 1 \) sigma-points \( \{\chi_i : i = 1, \ldots, 2L\} \) where \( \chi_i \in \mathbb{R}^l \). The sigma-point selection scheme for a simple 2-dimensional GRV is shown in Fig. 2. The sigma-points are calculated (with corresponding weights \( W_i \)) according to the following general selection scheme:

\[
\begin{align*}
\chi_0 &= \bar{x} \\
\chi_i &= \bar{x} + \left( \sqrt{(L+\lambda)P_x} \right) i = 1, \ldots, L \\
\chi_i &= \bar{x} - \left( \sqrt{(L+\lambda)P_x} \right) i = L + 1, \ldots, 2L
\end{align*}
\]
\[
W_0^{(m)} = \frac{\lambda}{(L + \lambda)} \\
W_0^{(c)} = \frac{\lambda}{(L + \lambda)} + \left(1 - \alpha^2 + \beta\right) \\
W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L + \lambda)} \quad i = 1, \ldots, 2L
\]  

where \( \lambda = \alpha^2(L + \kappa) - L \) is a scaling parameter. The constant \( \alpha \) determines the spread of the sigma points around \( \bar{x} \), and is usually set to a small positive value (e.g. \( 1 \leq \alpha \leq 10^{-4} \)). The constant \( \kappa \) is a secondary scaling parameter, which is usually set to \( 3 - L \) (see [14] for more details). \( \beta \) is used to incorporate prior knowledge of the distribution of \( x \) (for gaussian distribution \( \beta = 2 \)). \( \sqrt{(L + \lambda)P_x} \) is the \( i \)th column/row of the matrix square root (e.g. lower-triangular Cholesky factorization). These sigma vectors are propagated through the nonlinear function,

\[
Y_i = g(\chi_i) \quad i = 0, \ldots, 2L
\]

The mean and covariance for \( y \) are approximated using a weighted sample mean and covariance of the posterior sigma points,

\[
\bar{y} = \sum_{i=0}^{2L} W_i^{(m)} Y_i \\
P_y = \sum_{i=0}^{2L} \sum_{j=0}^{2L} W_i^{(c)} Y_i Y_j^T \\
P_{xy} = \sum_{i=0}^{2L} \sum_{j=0}^{2L} W_i^{(c)} \chi_i^T Y_i^T
\]

where \( \chi \) denotes the transformed sigma points.
where $W_{i}^{m}$ and $W_{i}^{c}$ are scalar weights. Note, all weights need not to be positive valued. In fact, depending on the specific sigma-point approach at hand, certain weights on the cross-terms are set equal to zero, i.e., $W_{ij}^{c}$ for certain $\{i,j\}$. The specific values of the weights $(W)$ and the scaling factors $(\gamma)$ depend on the type of sigma-point approach used: These include the unscented transformation [15] and the Stirling-interpolation based central difference transformation [29] to name but two.

The sigma-point approach differs substantially from general stochastic sampling techniques such as Monte-Carlo integration which require orders of magnitude more sample points in an attempt to propagate an accurate (possibly non-Gaussian) distribution of the state. The deceptively simple approach results in posterior approximations for the mean and covariance that are accurate to the 2nd order (Taylor series expansion) for any nonlinearity (3rd order accuracy is achieved if the prior random variable has a symmetric distribution, such as the exponential family of pdfs) [23]. Furthermore, no analytical Jacobians of the system equations need to be calculated as is the case for the EKF. This makes the sigma-point approach very attractive for use in “black box” systems where analytical expressions of the system dynamics are either not available or not in a form which allows for easy linearization. A simple example is illustrated in Fig. 3 [24] for a 2-dimensional system: the left plot displays the true mean and covariance propagation using Monte-Carlo sampling; the center plot shows the same using a linearization approach as done in EKF and the right one shows the performance of UT (only 5 sigma points are used). The superior performance of UT compared to linearization approach is clear from Fig. 3.

b. Sigma Point Kalman Filter (SPKF)

The sigma point kalman filter (spkf) is a straightforward extension of the UT to the recursive estimation as defined in Eq. 10. The state random variable (RV) is redefined as the concatenation of the original state and noise variables: $x_{k}^{a} = [x_{k}^{T} \, v_{k}^{T} \, n_{k}^{T}]$. The sigma point selection scheme, defined in subsection a, is applied to this new augmented state RV to calculate the corresponding sigma point set, $\{\chi_{k,i}^{a}; \, i=0,\ldots,2L\}$ where $\chi_{k,i}^{a} \in \mathbb{R}^{l_{x}+l_{v}+l_{n}}$. The UKF equations are provided in table 2. The specific type of resulting SPKF is determined by the choice of sigma-
point selection scheme (weights & scaling factors) as well as the specific method by which the propagated sigma-points are combined in order to calculate the posterior covariance matrices (see [23]). Note that the overall computational complexity of the SPKF is the same as that of the EKF. The SPKF has been successfully applied to numerous application, including time series prediction, state, parameter and dual estimation for automatic control, machine learning, and econometrics to name but a few [3,23,26]. In all cases, the superiority over the EKF has been well documented.

Table 2: SPKF equations

Initialization:

\[ \begin{align*}
\dot{x}_0 &= E[x_0], \quad P_{x_0} = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \\
\hat{x}_0 &= E[x_0^*] = \begin{bmatrix} \hat{x}_0^T & \hat{v}_0^T & \hat{n}_0^T \end{bmatrix}^T \\
P_0^* &= E\left[(x_0^* - \hat{x}_0^*)(x_0^* - \hat{x}_0^*)^T\right] = \begin{bmatrix} P_{x_0} & 0 & 0 \\
0 & P_v & 0 \\
0 & 0 & P_n \end{bmatrix}
\end{align*} \]

For \( k = 1 \ldots \infty \)

1. Calculate sigma-points:

\[ \chi_{i,k-1}^k = \left[ \hat{x}_{i-1}^k + \sqrt{(L + \lambda)P_{i,k-1}^*} \hat{x}_{i-1}^k - \sqrt{(L + \lambda)P_{i,k-1}^*} \right] \]

2. Time-update equations:

\[ \begin{align*}
\hat{x}_k &= \sum_{i=0}^{2L} W^i \chi_{i,k-1}^k \\
P_{i,k}^* &= \sum_{i=0}^{2L} W^i \left( \chi_{i,k-1}^k - \hat{x}_k \right) \left( \chi_{i,k-1}^k - \hat{x}_k \right)^T \\
\Upsilon_{i,k-1} &= h(\chi_{i,k-1}^k, \chi_{i,k-1}^k)
\end{align*} \]

3. Measurement update equations:

\[ \begin{align*}
P_{y_i,k} &= \sum_{i=0}^{2L} W^i \left( Y_{i,k-1} - \hat{y}_k \right) \left( Y_{i,k-1} - \hat{y}_k \right)^T \\
P_{x_i,k} &= \sum_{i=0}^{2L} W^i \left( \chi_{i,k-1}^k - \hat{x}_k \right) \left( Y_{i,k-1} - \hat{y}_k \right)^T \\
K_k &= P_{x_i,k} P_{y_i,k}^{-1} \\
\hat{x}_k &= \hat{x}_k + K_k (y_k - \hat{y}_k) \\
P_{x_i,k} &= P_{x_i,k} - K_k P_{y_i,k} K_k^T
\end{align*} \]

where \( P_v = \) process noise cov., \( P_n = \) measurement noise cov., \( \chi^* = \) \[ \begin{bmatrix} \chi^T & (\chi^*)^T & (\chi^*)^T \end{bmatrix}^T \]
VI. EKF/SPKF BASED SLAM

In this section, the application of the EKF and SPKF is described to the problem of simultaneous estimation of the state and map parameters for localizing an unmanned vehicle. The experimental scenario considered here consists of an unmanned vehicle maneuvering through an unsurveyed environment within a 10 m × 10 m bounded region. Though the terrain aided navigation algorithm presented in this paper can be extended to a full three dimensional environment (e.g., for Unmanned Aerial Vehicles), for ease of visualization, simulations were conducted in two dimensions. Three onboard sensors obtain terrain information and an IMU provides acceleration and angular rate in the body fixed frame. We compared our SPKF based system performance to the baseline EKF with specific focus on: 1) vehicle state estimation accuracy given the known maps, 2) map parameter estimation accuracy given the true vehicle location, and 3) dual state and parameter estimation accuracy. The next subsections will discuss the specific system process and observation (map) models used inside the EKF/SPKF based estimators.

a. Vehicle Process Model

In the vehicle system process model, the standard IMU driven kinematic process model formulation [11,32] that comprises an inertial navigation system (INS) mechanization component and sensor bias error components was followed. In two dimensions, the IMU includes two accelerometers and one rate gyros. Errors in the sensors include both bias and additive noise. The vehicle state vector was given by:

\[
\hat{x}_t = \begin{bmatrix} p^T & v^T & \psi & b_a^T & b_a \end{bmatrix}
\]

(31)

where \( p = [x \ y]^T \), \( v = [v_x \ v_y]^T \), are the position and velocity vectors of the vehicle in the 2D body frame, \( \psi \) is the Euler angle (yaw), \( b_a = [b_a, b_a] \) is the accelerometer bias vector, and \( b_a \) is the IMU gyro rate bias. There is no separate scale error term in the state vector, as it was found to be sufficient to model both the sensor bias and scale error as a time varying bias term. The continuous time kinematic equations of the vehicle followed in this paper are [23]:
\[
\dot{x} = \cos \psi \nu_x - \sin \psi \nu_y \tag{32}
\]
\[
\dot{y} = \sin \psi \nu_x + \cos \psi \nu_y \tag{33}
\]
\[
\psi = z_{\omega} - b_{\omega} \tag{34}
\]
\[
\dot{v}_x = z_x - b_x + (z_{\omega} - b_{\omega})v_y \tag{35}
\]
\[
\dot{v}_y = z_y - b_y - (z_{\omega} - b_{\omega})v_x \tag{36}
\]
\[
b_x = w_{v_x} \tag{37}
\]
\[
b_y = w_{v_y} \tag{38}
\]
\[
b_{\omega} = w_{\omega} \tag{39}
\]

Accelerometer and rate gyro sensor measurements are denoted as \( z_a = \begin{bmatrix} z_x & z_y \end{bmatrix} \) and \( z_{\omega} \). They are defined as:

\[
z_a = a + b_a + n_a \tag{40}
\]
\[
z_{\omega} = \omega + b_{\omega} + n_{\omega} \tag{41}
\]

where \( a = [a_x \ a_y] \) is the vehicle true acceleration and \( \omega \) is the true angular rate, \( n_a \) and \( n_{\omega} \) are IMU acceleration and gyro rate additive noise terms. It was assumed that the accelerometer is situated exactly at the vehicle center of gravity (CG). The IMU sensor bias error terms were modeled as slowly varying random walk model as shown in eqs (37-39), where \( w_{v_x} = \begin{bmatrix} w_{v_x} & w_{v_y} \end{bmatrix} \) and \( w_{\omega} \) are zero mean gaussian random variables.

### b. Terrain Map (Observation Model)

In this simulation, three terrain maps are used. Each map can be thought of as providing complimentary terrain information such as altitude, pressure, or visual characteristics of the environment. For this preliminary study, each map was modeled as a mixture of 3, 5 and 2 Gaussians respectively. Clearly, this is not a complex realistic scenario, but serves the purpose to investigate the ability of EKF/SPKF estimators to generate consistent estimates of the vehicle state and map parameters after convergence and to provide a performance comparison between the SPKF and EKF implementation. Fig. 4 shows visual contour plots of these maps.

The measurements from the 3 onboard sensors correspond to each of the map values at the current true vehicle location plus additive noise. The role of the Kalman filter state-estimator is thus to fuse these 3 sensor readings with the process model to estimate the vehicle location and heading. The signal to noise ratio (SNR) at the output of each terrain sensor was kept as 30 db. A typical sensor trace from a random trajectory is shown in Fig. 5. In the
Fig. 4 displays the true terrain maps used in our simulation.

Fig. 5: Trace of 3 noisy sensor (S) observation

dual Kalman filter setup for SLAM, the maps are assumed unknown and some parametric representation of the maps must be simultaneously learned from these sensor traces.

VII. EXPERIMENTAL RESULTS

The experimental result section is subdivided into three parts: a) vehicle state estimation with the assumption that the perfect terrain maps are known, b) map parameter estimation with assumption that we know exactly the position of the vehicle, and c) dual estimation when both the trajectory and maps are unknown.

<table>
<thead>
<tr>
<th>MSE</th>
<th>EKF</th>
<th>SPKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position (m)</td>
<td>0.0089</td>
<td>0.0052</td>
</tr>
<tr>
<td>Heading (deg)</td>
<td>0.0034</td>
<td>0.0011</td>
</tr>
</tbody>
</table>

Table 3: Position and Heading angle estimation accuracy: SPKF vs. EKF
a. Vehicle State Estimation

To perform this simulation, the vehicle was driven in the 10m×10m bounded space by generating random accelerations. Here it was assumed that the estimator knew perfectly the terrain maps. Fig. 6 displays the true and SPKF/EKF estimated positions of the vehicle. In this resolution, though the true, SPKF and EKF estimated paths seems indistinguishable from one another, actually both position and yaw angle error is reduced with SPKF as seen in Fig. 7. Table 3 shows the tracking accuracy for both algorithms in terms of MSE. The MSE of the 2D position was computed by a moving average window over time. Each window is 10 sec long with a 4 sec overlap between successive windows. From Fig. 6 and Table 3, it can be observed that the SPKF provided less error in both vehicle position and heading angle over the EKF, though the performance improvement was not much significant in this case. This experiment, simply illustrates that the Kalman state-estimation framework is sufficient to localize the vehicle given the true maps.

b. Map Parameter Estimation

In this section, the SPKF/EKF performance was evaluated on map estimation. In this experiment, it was assumed that the true vehicle trajectory is known. As described earlier, the true maps in this case consist of a mixture of Gaussians. However, since we cannot assume prior knowledge of this, we attempted to learn this map using both two layer Multilayer Perceptrons (MLP) and Radial Basis function (RBF) neural networks.
Each network implements a mapping from a two dimensional input space to a one dimensional output space. A different network is used for each sensor map. Typically, this would be trivial matter. However, it is made more challenging by the fact that the inputs and outputs are limited to the random trajectory and corresponding sensor traces from the vehicle. The state vector in the map estimation filter contains all the hidden and output layer weights. In RBF, gaussian prototypes are used at the hidden layer. Weights of the hidden layer and output layer of each network (MLP or RBF) are trained using either the EKF or SPKF. We also implemented conjugate gradient descent for a baseline comparison. The number of hidden nodes in each type of network was decided using a ten fold cross validation technique. It was found that the optimum number of hidden nodes in the MLP and RBF networks were 25 and 12 respectively.

<table>
<thead>
<tr>
<th>Parameters Estimation Error (MLP)</th>
<th>Parameters Estimation Error (MLP)</th>
<th>Parameters Estimation Error (MLP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>SPKF</td>
<td>Conjgrad</td>
</tr>
<tr>
<td>MSE (m)</td>
<td><strong>MSE (m)</strong></td>
<td><strong>MSE (m)</strong></td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>0.1</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>0.2</td>
<td>0.125</td>
<td>0.125</td>
</tr>
<tr>
<td>0.3</td>
<td>0.125</td>
<td>0.125</td>
</tr>
<tr>
<td>0.4</td>
<td>0.102</td>
<td>0.102</td>
</tr>
</tbody>
</table>

Table 4: Performance comparison of various learning algorithms (MSE of each algorithm over each map is the average of MSE obtained in MLP and RBF)
The convergence of map 1, 2 and 3 using both MLP and RBF are shown in Fig. 8. For learning curves shown, each epoch represents a random vehicle trajectory of 1000 output observations. Table 4 depicts the accuracy of the SPKF, EKF, and conjugate gradient descent algorithms in terms of MSE after convergence. The true and learned maps were divided into grids and the MSE was the average squared difference between those grid points. In each learning curve it can be seen that both the SPKF and EKF performed better than conjugate gradient descent learning. Comparing the SPKF and EKF, the SPKF converged earlier than the EKF and after convergence also provided lower MSE. Fig. 9 displays EKF/SPKF reconstructed maps (MLP cases only, as RBF cases are similar to MLP) after convergence of the map parameters.
Finally, the SPKF and EKF performance was evaluated for the dual estimation or SLAM problem. Comparisons are made for 1) speed of convergence for the state and map estimation, 2) tracking accuracy of the state after convergence, and 3) accuracy of the learned map.

Unlike the state and parameter estimation cases, neither the initial values of the state nor the maps were assumed known. At each epoch, the vehicle was driven randomly over a 10m×10m bounded space and 1000 sensor observations were collected from the vehicle’s trajectory. A new vehicle trajectory was generated at each epoch in the same bounded region. The estimated state and map parameters from the previous epoch were used to initialize the next epoch. A dual kalman framework is applied to learn both the hidden state and the sensor maps. At every time step, the vehicle state filter estimates the state \( \hat{x}_k \) using the current map parameter estimate \( \hat{w}_k \), while the weight filter estimates the weights \( \hat{w}_k \) using the current state estimate \( \hat{x}_k \) as described in section II. Though the SPKF and EKF were initialized with the same random initial state and maps, the SPKF was able to converge to the true state and maps earlier than the EKF, as shown in Fig. 10.
Typically, the SPKF convergence was 100-150 epochs earlier than the EKF. In addition to that, SPKF approach converged to a lower MSE than EKF in both map parameter estimation and vehicle position estimation cases.

Fig. 11-12 and Table 5 show that the SPKF also provided significant gain over the EKF in final performance. Fig. 11 shows the estimated trajectory of the SPKF and EKF versus the true trajectory after convergence of the state and map. As seen in Fig. 12, both the position and heading error is reduced with the SPKF, especially at the periods of peak errors during rapid changes in motion of the vehicle.
Fig. 13: Reconstructed terrain maps after convergence

Fig. 13 displays the SPKF/EKF reconstructed maps after convergence. Reconstructed maps are shown for the MLP cases only, as the RBF performance was similar.

VIII. CONCLUSION AND FUTURE WORK

This paper has presented a dual Kalman estimation framework for the problem of navigating an unmanned vehicle over unsurveyed environments. Results of simulations conducted in two dimensions showed that the SPKF based implementation of state and parameters not only converges faster than those of the EKF, but also tracks both state and parameters at a lower MSE. The major advantage of this algorithm is that vehicle localization is possible from it’s sensor maps, though there is no explicit position information available. The SNR at the terrain sensors was
kept at 30 db. It was noted, with higher noise power for these terrain sensors, dual estimation did not consistently converge. It was also observed from the simulation results that the number of epochs taken by the vehicle state and map parameters to converge in dual estimation case was considerably high, requiring further investigation. The results clearly show that without a priori knowledge of the environment and vehicle state, the algorithm has been able to estimate the vehicle location within 0.13 m for EKF and within 0.05 m for SPKF. The three sensor maps are also converged within an error bound of 0.12 m for EKF and 0.06 m for SPKF. In addition to that, both EKF and SPKF algorithm have successfully bounded the IMU drift. While these results are clearly preliminary, it does show feasibility of the approach to the general SLAM problem and the SPKF supremacy over the conventional and widely used EKF algorithm.

The next direction of this research would be to use realistic maps such as ground elevation map or earth magnetic map as sensor observations and estimate both vehicle state and map parameters based on these observations. Another possible path of continuing this research would be to fuse terrain sensors with radar/sonar sensor measurements and perform dual estimation of vehicle state and parameter. The fusion among a number of sensors would definitely help to reduce the number of epochs required to converge the state and map parameters. The author has started to combine the terrain aided navigation framework (as studied here) with landmark location based estimation. In that framework, terrain sensors are fused with radar sensors which provide range and bearing measurements to certain landmarks scattered throughout the environment. Initial results demonstrated that SPKF algorithm significantly reduced the number of epochs taken by the state and map to converge and also provided lower MSE than those results presented in this paper. The author also plans to continue to investigate the performance of the SPKF/EKF, with extensions to a 3D environment for Unmanned Aerial Vehicle (UAV)’s. Additional extensions are also planned with more realistic maps of altitude, pressure, and vision sensors as observations.

REFERENCE


