Unscented Kalman Filter Implementation Notes

16-662 Spring 2015

John Yao

This document provides implementation details designed to help students with Part II of Project 2B for 16-662. Section III A of [1] provides useful background information on the Unscented Kalman Filter (UKF).

1 Notation

$(\cdot)_{k\vert k-1}$, $k \in \mathbb{N}$	estimate of $(\cdot)$ at timestep k, conditioned on measurements taken up to and including timestep $k-1$
$\bar{(\cdot)}$	mean of $(\cdot)$
$\mathbf{x} \in \mathbb{R}^n$	state
$\mathbf{P} \in \mathbb{R}^{n\times n}$	state covariance
$\mathbf{z} \in \mathbb{R}^p$	observation vector
$\mathbf{n} \in \mathbb{R}^q$	process noise vector
$\mathbf{m} \in \mathbb{R}^r$	observation noise vector
$\mathbf{Q} \in \mathbb{R}^{q\times q}$	process noise covariance matrix
$\mathbf{R} \in \mathbb{R}^{r\times r}$	observation noise covariance matrix
$\mathbf{e}_i$	the ith column of a 3×3 identity matrix
g	gravitational acceleration magnitude
$\mathbf{R}_{wb}$	rotation that takes vectors from body frame to world frame
ϕ, θ, ψ	roll, pitch, yaw (ZYX Euler angles corresponding to $\mathbf{R}_{wb}$)
$\boldsymbol{\omega}$	body angular velocity (expressed in body frame)
$\mathbf{p}$	vehicle position (expressed in world frame)
$\mathbf{v}$	vehicle velocity (expressed in world frame)
$\mathbf{b}_\omega$	IMU gyrometer angular velocity bias
$\mathbf{b}_a$	IMU accelerometer acceleration bias
$(\cdot)_m$	a measured quantity

2 Motivation

The UKF is an algorithm used to perform state estimation, which seeks to infer the probability distribution associated with a dynamic system's states by fusing multiple different sources of information. Kalman filter algorithms model the state probability distribution as Gaussian, propagating them through nonlinear functions representing state transition dynamics and sensor observation models. The Extended Kalman Filter (EKF) does this by linearizing the nonlinear function, obtaining the new mean by propagating the old mean through the nonlinear function and the new covariance by linearly transforming the old covariance via the Jacobian. In highly nonlinear functions, the error in the resultant distribution may lead to sub-optimal performance (see center of Figure 2).

The UKF addresses this problem by representing the probability distribution with a set of deterministically-chosen sample points (called *sigma points*), which after being propagated through the nonlinear function can be used to reconstruct the mean and covariance through regression. Compared with the first-order (Taylor series) accuracy of the EKF, the UKF is accurate to at least second order. While the UKF and EKF both have $\mathcal{O}(n^3)$ computational complexity, the former is significantly easier to implement because it does not require the derivation of analytic Jacobians.

3 State Estimate

The state estimate consists of the state mean

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \gamma \\ \mathbf{v} \\ \mathbf{b}_\omega \\ \mathbf{b}_a \end{bmatrix}, \; \mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}, \; \mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \; \boldsymbol{\gamma} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \; \mathbf{b}_\omega = \begin{bmatrix} b_{\omega x} \\ b_{\omega y} \\ b_{\omega z} \end{bmatrix}, \; \mathbf{b}_a = \begin{bmatrix} b_{ax} \\ b_{ay} \\ b_{az} \end{bmatrix} \tag{1}$$

and the state (error) covariance

$$\mathbf{P} = \mathbb{E}[(\mathbf{x} - \mathbf{x}_{\text{true}})(\mathbf{x} - \mathbf{x}_{\text{true}})^T] \tag{2}$$

The state mean's position, velocity and orientation states are initialized to user-supplied values (that represent the best knowledge about the environment) while the bias states are set to zero. The state covariance is generally initialized to a diagonal matrix whose elements are tuning parameters that encode the uncertainties in the initial conditions.

4 IMU Process Dynamics Model

This model specifies a set of equations that relate the current state to the next state in time. Here we give the continuous time equations, but in the project you will need to discretize them (e.g. using the Euler method). The accelerometer model (3) relates the vehicle's orientation, coordinate acceleration, gravity, accelerometer bias, and accelerometer noise to the linear acceleration measurement obtained from the IMU.

$$\mathbf{a}_m = \mathbf{R}_{wb}(\boldsymbol{\gamma})^T\left(\mathbf{a} + \mathbf{e}_3 g\right) + \mathbf{b}_a + \mathbf{n}_a \tag{3}$$

The gyrometer model (4) relates the vehicle's angular velocity, gyrometer bias, and gyrometer noise to the angular velocity measurement obtained from the IMU.

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_\omega + \mathbf{n}_\omega \tag{4}$$

We model the IMU biases as random walks.

$$\dot{\mathbf{b}}_\omega = \mathbf{n}_{b\omega} \tag{5}$$

$$\dot{\mathbf{b}}_a = \mathbf{n}_{ba} \tag{6}$$

The position, velocity and orientation evolve as follows:

$$\dot{\mathbf{p}} = \mathbf{v} \tag{7}$$

$$\dot{\mathbf{v}} = \mathbf{a} \tag{8}$$

$$\dot{\boldsymbol{\gamma}} = \mathbf{S}(\boldsymbol{\gamma})\boldsymbol{\omega} \tag{9}$$

where

$$\mathbf{S}(\phi, \theta, \psi) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \tag{10}$$

Let us concatenate all the measurement terms into an input vector $\mathbf{u}$:

$$\mathbf{u} = \begin{bmatrix} \boldsymbol{\omega}_m \\ \mathbf{a}_m \end{bmatrix} \tag{11}$$

and all process noise terms into a single process noise vector $\mathbf{n}$

$$\mathbf{n} = \begin{bmatrix} \mathbf{n}_\omega \\ \mathbf{n}_{b\omega} \\ \mathbf{n}_a \\ \mathbf{n}_{ba} \end{bmatrix} \tag{12}$$

The continuous time process model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{n})$ can be obtained by composing the equations above. Note that in the project you are advised to implement the discrete time update model $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_{k+1}, \mathbf{x}_k)$.

We assume that the noise vectors are random variables drawn from zero mean Gaussians with diagonal covariance matrices.

$$\mathbf{n}_\omega \sim \mathcal{N}(\mathbf{0}, \sigma_\omega^2 \mathbf{I}_{3\times3}) \tag{13}$$

$$\mathbf{n}_{b\omega} \sim \mathcal{N}(\mathbf{0}, \sigma_{b\omega}^2 \mathbf{I}_{3\times3}) \tag{14}$$

$$\mathbf{n}_a \sim \mathcal{N}(\mathbf{0}, \sigma_a^2 \mathbf{I}_{3\times3}) \tag{15}$$

$$\mathbf{n}_{ba} \sim \mathcal{N}(\mathbf{0}, \sigma_{ba}^2 \mathbf{I}_{3\times3}) \tag{16}$$

In the assignment, the $\sigma_{(\cdot)}$ terms are tuning parameters used to adjust the process noise covariance. However, in practice one should characterize them in an offline calibration process. The process noise covariance matrix is given by

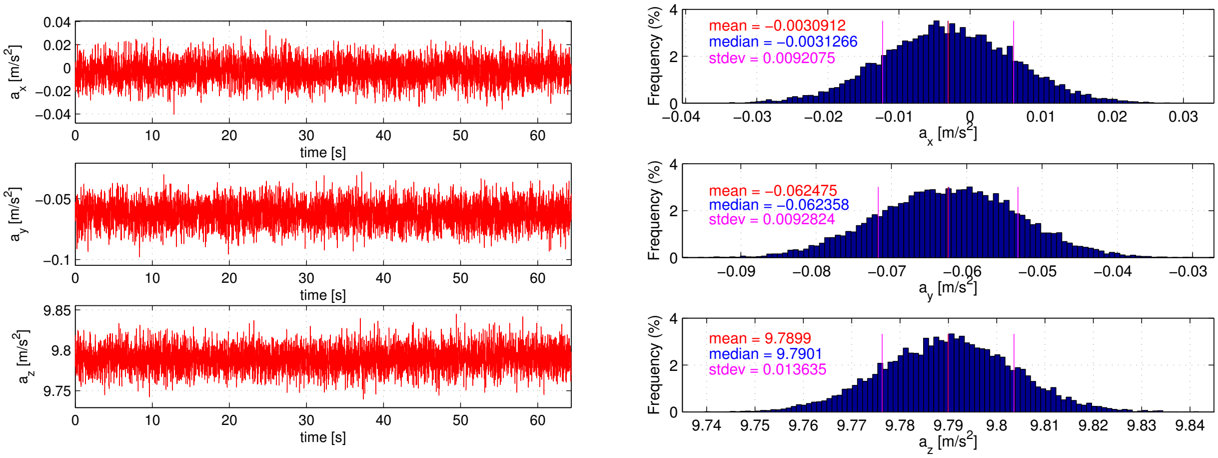
$$\mathbf{Q} = \mathbb{E}[\mathbf{n}\mathbf{n}^T] = \begin{bmatrix} \sigma_\omega^2 \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \sigma_{b\omega}^2 \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \sigma_a^2 \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \sigma_{ba}^2 \mathbf{I}_{3\times3} \end{bmatrix} \tag{17}$$

Figure 1 shows one method to determine the σ_a noise parameter from IMU accelerometer readings taken in a static configuration by extracting the standard deviation from the distribution of empirical observations.

5 Exteroceptive Measurement Model

The exteroceptive sensor we have access to provides observations on the world frame position and world frame heading. It can be thought of as a combination of a GPS, barometer and magnetometer, or the output of a visual SLAM algorithm. For the purposes of this assignment, you do not have to be concerned with the nonlinear sensor model equations. Instead, the measurement model is simply

$$\mathbf{z}_m = \mathbf{h}(\mathbf{x}, \mathbf{m}) = \begin{bmatrix} p_x \\ p_y \\ p_z \\ \psi \end{bmatrix} + \mathbf{m} \tag{18}$$



(a) Raw acceleration readings from a static IMU

(b) A histogram of static acceleration readings

Figure 1: Characterization of σ_a from static acceleration readings

where $\mathbf{m}$ is the observation noise vector

$$\mathbf{m} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{19}$$

$$\mathbf{R} = \begin{bmatrix} \sigma_{xy}^2 & 0 & 0 & 0 \\ 0 & \sigma_{xy}^2 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 \\ 0 & 0 & 0 & \sigma_\psi^2 \end{bmatrix} \tag{20}$$

In practice, the $\sigma_{(\cdot)}$ parameters in (20) are either determined offline through a sensor calibration process or computed online by an intermediate algorithm that produces a summarized observation from raw sensor data. An example of the former method is the use of a barometer for obtaining the p_z measurement. In this case σ_z can be determined by analyzing the statistical properties of the barometer readings under controlled environmental settings. An example of the latter method is the use of a 2D grid-based localization strategy for determining p_x and p_y. Intuitively, the uncertainty in the position observations (σ_{xy}) depends on grid size.

6 Working with Sigma Points

Sigma points provide a method to propagate a Gaussian random vector $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P}) \in \mathbb{R}^n$ through a nonlinear function and $\mathbf{y} = \mathbf{g}(\mathbf{x}) \in \mathbb{R}^p$.

6.1 Sigma Point Generation

Sigma points are a set of $2n + 1$ sample points defined as follows:

$$\mathcal{X}_i = \begin{cases} \bar{\mathbf{x}} + \zeta \left(\sqrt{\mathbf{P}}\right)_i & 1 \le i \le n \\ \bar{\mathbf{x}} - \zeta \left(\sqrt{\mathbf{P}}\right)_i & n+1 \le i \le 2n \\ \bar{\mathbf{x}} & i = 2n+1 \end{cases} \tag{21}$$

ζ is a tuning parameter that determines the spread of the sigma points in each of the dimensions of the state space. $\left(\sqrt{\mathbf{P}}\right)_i$ denotes the ith column of the matrix square root of the covariance matrix associated with $\mathbf{x}$. It is typically computed using a Cholesky decomposition.

6.2 Reconstructing the Mean and Covariance

After applying the nonlinear function to each of the input sigma points, we arrive at a set of output sigma points

$$\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_i) \tag{22}$$

The output mean is computed as

$$\bar{\mathbf{y}} = \sum_{i=0}^{2n} w_i^m \mathcal{Y}_i \tag{23}$$

The output covariance and cross-covariances between input and output can be approximated as follows:

$$\mathbf{P}^{\mathbf{yy}} = \sum_{i=0}^{2n} w_i^c \left(\mathcal{Y}_i - \bar{\mathbf{y}}\right)\left(\mathcal{Y}_i - \bar{\mathbf{y}}\right)^T \tag{24}$$

$$\mathbf{P}^{\mathbf{xy}} = \sum_{i=0}^{2n} w_i^c \left(\mathcal{X}_i - \bar{\mathbf{x}}\right)\left(\mathcal{Y}_i - \bar{\mathbf{y}}\right)^T \tag{25}$$

The w_i^m and w_i^c are weighting coefficients for the sigma points. These coefficients (and ζ) are related to the filter tuning parameters α, β and κ by:

$$\zeta = \alpha\sqrt{n+\kappa} \tag{26}$$

$$w_i^m = \begin{cases} \frac{1}{2\alpha^2(n+\kappa)} & i < 2n+1 \\ 1 - \frac{n}{\alpha^2(n+\kappa)} & i = 2n+1 \end{cases} \tag{27}$$

$$w_i^c = \begin{cases} \frac{1}{2\alpha^2(n+\kappa)} & i < 2n+1 \\ 2 - \frac{n}{\alpha^2(n+\kappa)} - \alpha^2 + \beta & i = 2n+1 \end{cases} \tag{28}$$

These equations as well as typical parameter values can be found in Appendix A of [1]. Figure 2 provides a graphical illustration of the sigma point approach to propagating a probability distribution through a nonlinear function.

7 Augmented State

In order to handle non-additive noise (i.e. $\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{n})$ as opposed to $\mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{n}$), we augment the state with the noise vector. In the process dynamics equation, the augmented state and covariance are given by

$$\mathbf{x}_a = \begin{bmatrix} \mathbf{x} \\ \mathbf{n} \end{bmatrix} \in \mathbb{R}^{n+q},\ \mathbf{P}_a = \begin{bmatrix} \mathbf{P} & \mathbf{0}_{n\times q} \\ \mathbf{0}_{q\times n} & \mathbf{Q} \end{bmatrix} \in \mathbb{R}^{(n+q)\times(n+q)} \tag{29}$$

In the observation model equation, the augmented state and covariance are given by

$$\mathbf{x}_a = \begin{bmatrix} \mathbf{x} \\ \mathbf{m} \end{bmatrix} \in \mathbb{R}^{n+r},\ \mathbf{P}_a = \begin{bmatrix} \mathbf{P} & \mathbf{0}_{n\times r} \\ \mathbf{0}_{r\times n} & \mathbf{R} \end{bmatrix} \in \mathbb{R}^{(n+r)\times(n+r)} \tag{30}$$

When performing a process update or correction update in the UKF, the sigma point generation and propagation can be done in the manner of Section 6, replacing $\mathbf{x}$ with $\mathbf{x}_a$ and $\mathbf{P}$ by $\mathbf{P}_a$ (n will get replaced by either $n+q$ or $n+r$).

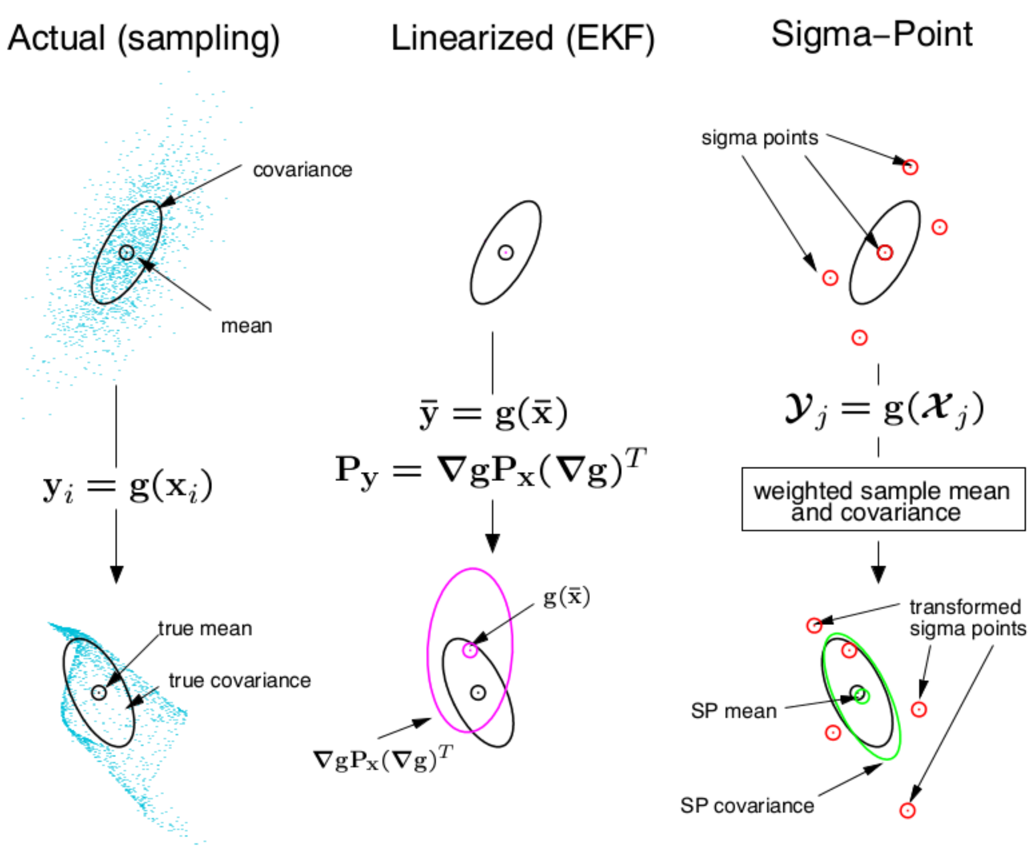


Figure 2: Comparison of Monte Carlo, EKF, and UKF (sigma point) approaches to propagate a 2D probability distribution through a nonlinear function (image taken from [1]).

8 Process Update

This occurs whenever the UKF receives an IMU measurement $\mathbf{u}$. Propagate the augmented state sigma points through the IMU process dynamics defined in Section 4. Then use the equations in Section 6 to reconstruct the prior mean $\mathbf{x}_{k|k-1}$ and covariance $\mathbf{P}_{k|k-1}$. Note that the input sigma points are of dimension $n+q$, but the propagated sigma points should be of dimension n.

9 Correction Update

This occurs whenever the UKF receives an exteroceptive sensor observation $\mathbf{z}$. Propagate the augmented state sigma points through measurement model defined in Section 5. Then use the equations in Section 6 to reconstruct the mean $\mathbf{z}$, covariance $\mathbf{P}^{\mathbf{zz}}$, and cross-covariance $\mathbf{P}^{\mathbf{xz}}$ of the predicted measurement. Note that the input sigma points are of dimension $n+r$, but the propagated sigma points should be of dimension p. The Kalman gain is given by

$$\mathbf{K} = \mathbf{P}^{\mathbf{xz}} \left(\mathbf{P}^{\mathbf{zz}}\right)^{-1} \tag{31}$$

This is used to compute the posterior mean

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}\left(\mathbf{z}_m - \mathbf{z}\right) \tag{32}$$

and the posterior covariance

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}\mathbf{P}^{\mathbf{zz}}\mathbf{K}^T \tag{33}$$

References

[1] R. Van Der Merwe, E. A. Wan, S. Julier *et al.*, “Sigma-point kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation,” in *Proceedings of the AIAA Guidance, Navigation & Control Conference*, 2004, pp. 16–19.