

Introduction to Unscented Kalman Filter

1 Introduction

In many scientific fields, we use certain models to describe the dynamics of system, such as mobile robot, vision tracking and so on. The word “dynamics” means we already master the principles regarding how system evolves. Taking mobile robot for example, we want to predict or estimate the future position of the robot, which can be derived by current position of robot and the integration of motor command sent to robot. This is the dynamical model of the mobile robot system. However, in real world, there is always some noise(process noise) which makes the dynamical control model not so accurate. Thus we keep losing information about position of robot. Generally, we can make use of location sensors to gain new information about current position so as to keep track of robot. The values of sensors are not perfectly reliable either(measurement noise). So we still cannot locate robot very well. Kalman filter is a powerful tool to filter out these two noise and give an accurate estimation of the position of robot.

2 Classic Kalman Filter

Essentially, Kalman filter is just a set of equations or computational tools that helps us to estimate the most possible future state of system.

Primitive Kalman filter can only be used to model linear system, which means we can use concise transformation matrix to formulate the dynamics of system and sensor models. A linear discrete time system can be modeled like this :

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k) \quad (\text{evolution or dynamics of system})$$

$$y(k) = H(k)x(k) + w(k) \quad (\text{sensor system})$$

Vector $x(k)$ is the system state at time point k , and then $x(k+1)$ is the system state at the following time point $k+1$. The components of $x(k)$ stand for those variables that we are most interested in, like position coordinates, velocity and bearing. Matrix $F(k)$ encodes the dynamics of system. Vector $u(k)$ is used to represent the system input such as velocity commands, torques, or forces intentionally applied to the robot, and Matrix $G(k)$ describes how the input drives the dynamics. Vector $v(k)$ is the process noise and is assumed to be white Gaussian noise with zero mean and covariance matrix $V(k)$. Vector $y(k)$ is the sensor output at time point k , and matrix $H(k)$ describes how state vectors are mapped into outputs. Vector $w(k)$ called measurement noise is also assumed to be white Gaussian noise with zero mean and covariance matrix $W(k)$.

The working procedure of Kalman filter includes two steps, which are “prediction” and “update”. In prediction step, we use the known system dynamics to predict the following system state. Then in update step, we use sensor values to adjust the former prediction in order to get a more accurate estimation of future system state. I need to mention that the output of Kalman filter in each step is actually a Gaussian Distribution of system state, and

Kalman filter gives us the mean and covariance matrix of this Gaussian Distribution.

The Kalman filter equations are summarized as follows:

Prediction:

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k) &= \mathbf{F}(k)\hat{\mathbf{x}}(k|k) + \mathbf{G}(k)\mathbf{u}(k) \\ \mathbf{P}(k+1|k) &= \mathbf{F}(k)\mathbf{P}(k|k)\mathbf{F}(k)^T + \mathbf{V}(k)\end{aligned}$$

Update:

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) + \mathbf{R}\mathbf{v} \\ \mathbf{P}(k+1|k+1) &= \mathbf{P}(k+1|k) - \mathbf{R}\mathbf{H}(k+1)\mathbf{P}(k+1|k)\end{aligned}$$

Where

$$\begin{aligned}\mathbf{v} &= \mathbf{y}(k+1) - \mathbf{H}(k+1)\hat{\mathbf{x}}(k+1|k) \\ \mathbf{S} &= \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}(k+1)^T + \mathbf{W}(k+1) \\ \mathbf{R} &= \mathbf{P}(k+1|k)\mathbf{H}(k+1)^T\mathbf{S}^{-1}\end{aligned}$$

$\hat{\mathbf{x}}(k+1|k)$ indicates the mean of system state at $k+1$ given all the sensor output before k
 $\mathbf{P}(k+1|k)$ indicates the covariance matrix of system state at $k+1$ given all the sensor before
 Similar for $\hat{\mathbf{x}}(k+1|k+1)$ and $\mathbf{P}(k+1|k+1)$, which are only the new estimation of system state taking into account the latest sensor output.

We can view \mathbf{R} as the weighting factor that takes into account the relationship between the accuracy of the predicted estimate and the measurement noise. If \mathbf{R} is large, then the sensor readings are more believable than the prediction and Kalman filter weights the sensor readings highly when computing the updated estimate. As we can see, the estimate of system state $\hat{\mathbf{x}}(k+1|k+1)$ and $\mathbf{P}(k+1|k+1)$ is based on distribution of state at the previous time point, $\hat{\mathbf{x}}(k|k)$ and $\mathbf{P}(k|k)$, which have already been computed. The other parameters are all known in the system.

(The complete derivation process of the Kalman filter equations can be found at Choset's Principles of Robot Motion: Theory, Algorithm and Implementations Chapter 8, section 8.2 Linear Kalman filter)

3 Unscented Kalman Filter

In cases, we need Kalman filter to deal with nonlinear systems, in which the state transition and observation models need not be linear functions of the state but may instead be differentiable functions like"

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k\end{aligned}$$

The function \mathbf{f} can be used to compute the predicted state from the previous estimate and similarly the function \mathbf{h} can be used to compute the predicted measurement from the predicted state. However, \mathbf{f} and \mathbf{h} cannot be applied to the covariance directly. Instead a matrix of partial derivatives (the Jacobian) is computed. At each timestep the Jacobian is

evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate. This is the basic principle of Extended Kalman filter(EKF). When the state transition and observation models – that is, the predict and update functions f and h (see above) – are highly non-linear, the extended Kalman filter can give particularly poor performance [JU97]. This is because only the mean is propagated through the non-linearity.

The unscented Kalman filter (UKF) is a recursive MMSE estimator that addresses some of the approximation issues of the EKF (Julier and Uhlmann 1997b). Because the EKF only uses the first order terms of the Taylor series expansion of the nonlinear functions, it often introduces large errors in the estimated statistics of the posterior distributions of the states. This is especially evident when the models are highly nonlinear and the local linearity assumption breaks down, i.e., the effects of the higher order terms of the Taylor series expansion becomes significant. Unlike the EKF, the UKF does not approximate the non-linear process and observation models, it uses the true nonlinear models and rather approximates the distribution of the state random variable. In the UKF the state distribution is still represented by a Gaussian random variable (GRV), but it is specified using a minimal set of deterministically chosen sample points. These sample points completely capture the true mean and covariance of the GRV, and when propagated through the true nonlinear system, captures the posterior mean and covariance accurately to the 2nd order for any nonlinearity, with errors only introduced in the 3rd and higher orders. Scaled unscented transformation is the algorithm core of UKF. I will introduce it next.

The unscented transformation (UT) is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation and builds on the principle that it is easier to approximate a probability distribution than an arbitrary nonlinear function (Julier and Uhlmann 1996). Consider propagating a n_x dimensional random variable x through an arbitrary nonlinear function g : $y = g(x)$

Assume x has mean \hat{x} and covariance $P(x)$. To calculate the statistics (first two moments) of y using the UT, we proceed as follows: First, a set of $2n_x + 1$ weighted samples or sigma points $S_i = \{W_i, X_i\}$ are deterministically chosen so that they completely capture the true mean and covariance of the prior random variable x . A selection scheme that satisfies this requirement is

$$X_0 = \hat{x} \quad W_0 = k/(n_x + k) \quad i = 0$$

$$X_i = \hat{x} + (\sqrt{(n_x + k)P(x)})_i \quad W_i = 1/\{2(n_x + k)\} \quad i = 1, 2, \dots, n_x$$

$$X_i = \hat{x} - (\sqrt{(n_x + k)P(x)})_i \quad W_i = 1/\{2(n_x + k)\} \quad i = n_x + 1, \dots, 2n_x$$

where k is a scaling parameter and $(\sqrt{(n_x + k)P(x)})_i$ is the i th row or column of the matrix square root of $(n_x + k)P(x)$. W_i is the weight associated with the i th point such that

$$\sum_{i=0}^{2n_x} W_i = 1$$

Each sigma point is now propagated through the nonlinear function

$$Y_i = g(X_i) \quad i = 0 \dots 2n_x$$

and the estimated mean and covariance of y are computed as follows

$$\hat{y} = \sum_{i=0}^{2n_x} W_i * Y_i$$

$$P(y) = \sum_{i=0}^{2n_x} W_i * (Y_i - \hat{y})(Y_i - \hat{y})^T$$

These estimates of the mean and co variance are accurate to the second order (third order for Gaussian priors) of the Taylor series expansion of $g(x)$ for any nonlinear function. Errors are introduced in the third and higher order moments but are scaled by the choice of the parameter κ . In comparison, the EKF only calculates the posterior mean and covariance accurately to the first order with all higher order moments truncated.

The sigma point selection scheme used in the UT has the property that as the dimension of the state-space increases, the radius of the sphere that bounds all the sigma points increases as well. Even though the mean and covariance of the prior distribution are still captured correctly, it does so at the cost of sampling non-local effects. If the nonlinearities in question are very severe, this can lead to significant difficulties. In order to address this problem, the sigma points can be scaled towards or away from the mean of the prior distribution by a proper choice of κ . The distance of the i th sigma point from \hat{x} , $|X_i - \hat{x}|$ is proportional to $\sqrt{n_x + \kappa}$. When $\kappa = 0$, the distance is proportional to n_x . When $\kappa > 0$ the points are scaled further from \hat{x} and when $\kappa < 0$ the points are scaled towards \hat{x} . For the special case of $\kappa = 3 - n_x$, the desired dimensional scaling invariance is achieved by canceling the effect of n_x . However, when $\kappa = 3 - n_x < 0$ the weight $W_0 < 0$ and the calculated covariance can be non-positive semidefinite. The scaled unscented transformation was developed to address this problem (Julier 2000).

Here is the actual computing process using scaled unscented transformation of unscented Kalman filter:

Predict

As with the EKF, the UKF prediction can be used independently from the UKF update, in combination with a linear (or indeed EKF) update, or vice versa.

The estimated state and covariance are augmented with the mean and covariance of the process noise.

$$\mathbf{x}_{k-1|k-1}^a = [\hat{\mathbf{x}}_{k-1|k-1}^T \quad E[\mathbf{w}_k^T]]^T$$

$$\mathbf{P}_{k-1|k-1}^a = \begin{bmatrix} \mathbf{P}_{k-1|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix}$$

A set of $2L+1$ sigma points is derived from the augmented state and covariance where L is the dimension of the augmented state.

$$\chi_{k-1|k-1}^0 = \mathbf{x}_{k-1|k-1}^a$$

$$\chi_{k-1|k-1}^i = \mathbf{x}_{k-1|k-1}^a + \left(\sqrt{(L + \lambda) \mathbf{P}_{k-1|k-1}^a} \right)_i \quad i = 1..L$$

$$\chi_{k-1|k-1}^i = \mathbf{x}_{k-1|k-1}^a - \left(\sqrt{(L + \lambda) \mathbf{P}_{k-1|k-1}^a} \right)_i \quad i = L + 1, \dots, 2L$$

where

$$\left(\sqrt{(L + \lambda) \mathbf{P}_{k-1|k-1}^a} \right)_i$$

is the i th column of the matrix square root of

$$(L + \lambda) \mathbf{P}_{k-1|k-1}^a$$

using the definition: square root A of matrix B satisfies

$$B \equiv AA^T.$$

The matrix square root should be calculated using numerically efficient and stable methods such as the [Cholesky decomposition](#).

The sigma points are propagated through the transition function f .

$$\chi_{k|k-1}^i = f(\chi_{k-1|k-1}^i) \quad i = 0..2L$$

The weighted sigma points are recombined to produce the predicted state and covariance.

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2L} W_s^i \chi_{k|k-1}^i$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2L} W_c^i [\chi_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1}][\chi_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1}]^T$$

where the weights for the state and covariance are given by:

$$W_s^0 = \frac{\lambda}{L + \lambda}$$

$$W_c^0 = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta)$$

$$W_s^i = W_c^i = \frac{1}{2(L + \lambda)}$$

$$\lambda = \alpha^2(L + \kappa) - L$$

Typical values for α , β , and κ are 10^{-3} , 2 and 0 respectively. (These values should suffice for most purposes.)

Update

The predicted state and covariance are augmented as before, except now with the mean and covariance of the measurement noise.

$$\mathbf{x}_{k|k-1}^a = [\hat{\mathbf{x}}_{k|k-1}^T \ E[\mathbf{v}_k^T]]^T$$

$$\mathbf{P}_{k|k-1}^a = \begin{bmatrix} \mathbf{P}_{k|k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_k \end{bmatrix}$$

As before, a set of $2L + 1$ sigma points is derived from the augmented state and covariance where L is the dimension of the augmented state.

$$\chi_{k|k-1}^0 = \mathbf{x}_{k|k-1}^a$$

$$\chi_{k|k-1}^i = \mathbf{x}_{k|k-1}^a + \left(\sqrt{(L + \lambda)\mathbf{P}_{k|k-1}^a} \right)_i \quad i = 1..L$$

$$\chi_{k|k-1}^i = \mathbf{x}_{k|k-1}^a - \left(\sqrt{(L + \lambda)\mathbf{P}_{k|k-1}^a} \right)_{i-L} \quad i = L + 1, \dots, 2L$$

The sigma points are projected through the observation function h .

$$\gamma_k^i = h(\chi_{k|k-1}^i) \quad i = 0..2L$$

The weighted sigma points are recombined to produce the predicted measurement and predicted measurement covariance.

$$\hat{\mathbf{z}}_k = \sum_{i=0}^{2L} W_s^i \gamma_k^i$$

$$\mathbf{P}_{z_k z_k} = \sum_{i=0}^{2L} W_c^i [\gamma_k^i - \hat{\mathbf{z}}_k][\gamma_k^i - \hat{\mathbf{z}}_k]^T$$

The state-measurement cross-covariance matrix,

$$\mathbf{P}_{x_k z_k} = \sum_{i=0}^{2L} W_c^i [\chi_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1}][\gamma_k^i - \hat{\mathbf{z}}_k]^T$$

is used to compute the UKF Kalman gain.

$$\mathbf{K}_k = \mathbf{P}_{x_k z_k} \mathbf{P}_{z_k z_k}^{-1}$$

As with the Kalman filter, the updated state is the predicted state plus the innovation weighted by the Kalman gain,

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k)$$

And the updated covariance is the predicted covariance, minus the predicted measurement covariance, weighted by the Kalman gain.

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{z_k z_k} \mathbf{K}_k^T$$

References

- [1] Wikipedia article “Kalman Filter” http://en.wikipedia.org/wiki/Unscented_Kalman_filter
- [2] The Unscented Particle Filter, Tech report CUED/F-INFENG/TR-380, Cambridge University Engineering Department, Cambridge, England, Aug, 2000