The Extended Kalman filter

Carolyn Johnston

December 11, 2019

1 Introduction

This writeup is a follow-on to my earlier writeup, "Derivation of The Bayes and Kalman Filters" ([2]). As with that one, the purpose of this writeup is to provide mathematical details for topics covered in the textbook "Probabilistic Robotics" by Sebastian Thrun, Wolfram Burgard, and Dieter Fox [1], specifically in Chapter 3.3.

The Extended Kalman Filter (EKF) is, as the name implies, an extension of the original, linear Kalman filter to cases where the state transition equation or sensor model equation is non-linear, so that the Kalman filter can't be applied.

The Kalman filter is the Bayes filter that is derived when the following assumptions are made:

1. The prior $f(x_0)$ is a multivariate Gaussian in the state vector x_0;

2. For each time step, the state transition equation must be linear in its arguments x_{t-1} and u_t, with additive mean-zero Gaussian noise ϵ_t:

$$x_t = A_t x_{t-1} + B_t u_u + \epsilon_t \tag{1}$$

3. For each time step, the sensor equation must be linear in its argument x_t with additive mean-zero Gaussian noise δ_t :

$$z_t = C_t x_t + \delta_t. \tag{2}$$

A_t, B_t, C_t are all linear (matrices), and their values (as well as the parameters of the noise vectors ϵ_t and δ_t) can vary with each time step. These conditions ensure that the beliefs $bel(x_t) = f(x_t|z_{1:T}, u_{1:T})$ are multivariate Gaussian distributions at every time step t.

For most real-world systems, conditions 1 and 2 of the Kalman filter model are violated; neither the motion model nor the sensor model are linear in their arguments. As a result, the predicted and corrected beliefs $\overline{bel}(x_t)$ and $bel(x_t)$ are usually not Gaussian, because applying a non-linear function to a Gaussian random variable does not necessarily result in a Gaussian random variable.

The EKF is obtained by replacing, at each step of the algorithm, the nonlinear motion model and sensor model functions with their linear approximations.

Since the motion model and sensor model functions are linearized at each point, the requirements for the original Kalman filter are met; therefore, all the posterior beliefs remain Gaussian.

This makes the EKF ill-suited for problems in which the true posterior probabilities are not well represented by Gaussians. For example, multimodal beliefs occur frequently in robotics problems and are completely unrepresentable with single Gaussians. The EKF is only suited for problems in which the bulk of the prior belief $bel(x_t)$ lies in a near-linear portion of the nonlinear function $g(u_t, x_{t-1})$, and the bulk of $\overline{bel}(x_t)$ lies in a near-linear portion of $h(x_t)$; in particular, it can't represent multimodal beliefs at all, and other methods have been invented to handle that case. The EKF is, essentially, a first step into the domain of nonlinear transition and sensor model functions in Bayesian filtering.

2 The EKF prediction and update steps

The goal of the Bayes filter algorithm is to calculate the 'belief' distribution at time T, $bel(x_T) = f(x_T|z_{1:T}, u_{1:T})$. This is the p.d.f. of the current state conditioned on all the data and measurements to that point, including the current sensor measurement z_T; in short, it is our probabilistic best guess of the system state at time T.

Each Bayes filter recursion is a two step process. In the first *prediction* step, the control measurement u_T is incorporated, and an intermediate belief $\overline{bel}(x_T) = f(x_T|z_{1:T-1}, u_{1:T})$, conditioned only on all past data and the current control measurement, is calculated. This intermediate belief step generally increases the uncertainty in the state estimate; the previous belief is convolved (essentially smeared) by the transition p.d.f.. The second step of the Bayes filter algorithm multiplies $\overline{bel}(x_T)$ by the likelihood of x_T after the measurement z_T, so this step generally reduces the uncertainty of $\overline{bel}(x_T)$.

In the EKF, we begin by assuming that the prior $bel(x_0)$ is a Gaussian. The state transition probability is assumed to be governed by a nonlinear function of x_{t-1}, plus additive Gaussian noise. Equation 1 is replaced by:

$$x_t = g(u_t, x_{t-1}) + \epsilon_t. \tag{3}$$

The sensor measurement probability is assumed to be governed by a nonlinear function of x_t, plus additive Gaussian noise. So equation 2 is replaced by

$$z_t = h(x_t) + \delta_t. \tag{4}$$

The trick to the EKF is that the functions g and h are replaced by their first-order (linear) Taylor expansions on every prediction and update step. This keeps the posterior belief, $bel(x_t)$, Gaussian at each step.

Calculating $\overline{bel}(x_t)$

Think of the transition function $g(u_t, x_{t-1})$ as though u_t were a constant: define $g(u_t,x_{t-1}) \equiv g_{u_t}(x_{t-1})$. The function g_{u_t} maps R^N to R^N, where N is the dimension of the state space, and therefore its Jacobian (derivative) at a (state) point x_0 is an NxN matrix, defined by:

$$[G_t|_{x_0}]_{ij} = [\left.\frac{\delta(g_{u_t})_j}{\delta x_i}\right|_{x_0}]; \tag{5}$$

i.e., the derivative matrix consists of gradient vectors of each component of the vector function g_{u_t} arranged in columns.

In the equation 3, the state x_t is given by the function $g_{u_t}(x_{t-1})$ plus a noise term ϵ_t. In order to define a Taylor expansion for the estimate, we must choose a representative point $x_{t-1} \equiv x_0$ in the domain of $bel(x_{t-1})$ to expand the function g_{u_t} around. This point should be central to the support of $bel(x_{t-1})$, since that represents our best knowledge about the state at time $t-1$. Since the prior belief $bel(x_{t-1})$ is assumed to be Gaussian, defined as $bel(x_{t-1}) \sim N(\mu_{t-1}, \Sigma_{t-1})$, we take the mean μ_{t-1} to be the representative point for the Taylor expansion. We therefore approximate $g_{u_t}$by the linear function

$$g_{u_t}(x_{t-1}) \approx g_{u_t}(\mu_{t-1}) + G_t|_{\mu_{t-1}} \cdot (x_{t-1} - \mu_{t-1}),$$

where G_t is the NxN matrix given in equation 5. This should be a close approximation to g_{u_t} over the domain of $bel(x_{t-1})$ in order for the EKF to give good results.

With this approximation, we get the Gaussian predicted belief

$$p(x_t|u_t, x_{t-1}) \propto$$

$$\exp(-\frac{1}{2}(x_t-[g_{u_t}(\mu_{t-1})+G_t|_{\mu_{t-1}}\cdot(x_{t-1}-\mu_{t-1})])^t R_t^{-1}(x_t-[g_{u_t}(\mu_{t-1})+G_t|_{\mu_{t-1}}\cdot(x_{t-1}-\mu_{t-1})]),$$

where R_t is the covariance of the noise term ϵ_t (note that, although the above equation looks pretty gruesome, the expression still has the basic Gaussian form: $\exp(-\frac{1}{2}(x-\mu)^t R^{-1}(x-\mu))$).

It follows from the Bayes filter prediction step that

$$\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) \cdot bel(x_{t-1})dx_{t-1} \propto$$

$$\int \exp(-\frac{1}{2}(x_t-[g_{u_t}(\mu_{t-1})+G_t|_{\mu_{t-1}}\cdot(x_{t-1}-\mu_{t-1})])^t R_t^{-1}(x_t-[g_{u_t}(\mu_{t-1})+G_t|_{\mu_{t-1}}\cdot(x_{t-1}-\mu_{t-1})])$$

$$\cdot\exp(-\frac{1}{2}(x_{t-1} - \mu_{t-1})^t \Sigma_{t-1}^{-1}(x_{t-1} - \mu_{t-1}))dx_{t-1}.$$

This integrand, awful though it looks, is just the product of two multivariate normal distributions. It can be calculated using the result of the Convolution Theorem from the earlier writeup on Kalman filters [2]. The Convolution Theorem is restated here:

$$f(x) = \int \exp(-\frac{1}{2}[(x-Ay-w)^T R^{-1}(x-Ay-w)+(y-\mu)^T\Sigma^{-1}(y-\mu)]dy$$

$$\implies f(x) \sim N(A\mu + w, A\Sigma A^T + R).$$

We apply the Convolution Theorem to the previous formula by setting $x = x_t$, $y = x_{t-1}$, $A = G_t|_{\mu_{t-1}}$, $w = g(u_t, \mu_{t-1}) - G_t|_{\mu_{t-1}} \mu_{t-1}$, $\mu = \mu_{t-1}$, $R = R_t$, and $\Sigma = \Sigma_{t-1}$. We end up with

$$\overline{bel}(x_t) \sim N(g(u_t, \mu_{t-1}), G_t|^t_{\mu_{t-1}} \Sigma_{t-1} G_t|_{\mu_{t-1}} + R_t). \qquad (6)$$

Thus $\overline{bel}(x_t)$ is estimated as a Gaussian with mean $\overline{\mu}_t = g(u_t, \mu_{t-1})$, and covariance $\overline{\Sigma_t} = G_t|^t_{\mu_{t-1}} \Sigma_{t-1} G_t|_{\mu_{t-1}} + R_t$.

Calculating $bel(x_t)$

Consider the function $h(x_t)$ in equation 4. This function maps points in the state space R^N to points in the sensor measurement space, R^M. Therefore its Jacobian at a point x_0 in the state space will be an NxM matrix,

$$[H_t|_{x_0}]_{ij} = [\left.\frac{\delta(h_t)_j}{\delta x_i}\right|_{x_0}],$$

i.e., the derivative matrix consists of the gradient vectors of each of the M components of the vector function h_t arranged in columns.

According to the definition of the update step for Bayes filters,

$$bel(x_t) \propto p(z_t|x_t)\overline{bel}(x_t).$$

In equation 4, we replace $h(x_t)$ with its Taylor expansion around a representative point in the domain of x_t. As before, this point should be central to the support of our current best estimate of the state at time t, and so we select $\overline{\mu}_t$ as the expansion point. This gives a linear approximation

$$h(x_t) \approx h(\overline{\mu}_t) + H_t|_{\overline{\mu}_t} \cdot (x_t - \overline{\mu_t}), \qquad (7)$$

which should be verified to be a good approximation in the neighborhood of $\overline{\mu}_t$ where the predicted belief $\overline{bel}(x_t)$ is concentrated. Then $bel(x_t) = \eta\exp(-J_t)$, with

$$J_t = \frac{1}{2}[(z_t-[h(\overline{\mu}_t)+H_t|_{\overline{\mu}_t}\cdot(x_t-\overline{\mu_t})])^t Q_t^{-1}(z_t-[h(\overline{\mu}_t)+H_t|_{\overline{\mu}_t}\cdot(x_t-\overline{\mu_t})])+(x_t-\overline{\mu}_t)^t\overline{\Sigma}_t^{-1}(x_t-\overline{\mu}_t).$$

Note that this is of the same form as the analogous expression in the linear Kalman filter, discussed in [2]:

$$J_t = \frac{1}{2}[(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t) + (x_t - \overline{\mu}_t)^T \overline{\Sigma}_t^{-1}(x_t - \overline{\mu}_t)],$$

with C_t replaced by the (constant) matrix $H_t|_{\overline{\mu_t}}$, and z_t replaced by the constant term $z_t - h(\overline{\mu}_t) + H_t|_{\overline{\mu}_t} \overline{\mu_t}$. J_t is therefore a quadratic in x_t, and $bel(x_t)$ is a Gaussian in x_t. To find the mean of the Gaussian, we solve

$$\frac{\delta J_t}{\delta x_t} = -H_t^t Q_t^{-1}(z_t - h(\overline{\mu}_t) - H_t(x_t - \overline{\mu}_t)) + \overline{\Sigma}_t^{-1}(x_t - \overline{\mu}_t) = 0$$

$$\implies H_t^t Q_t^{-1}(z_t - h(\overline{\mu}_t)) = (H_t^t Q_t^{-1} H_t + \overline{\Sigma}^{-1})(x_t - \overline{\mu}_t),$$

where we denote $H_t|_{\overline{\mu}_t}$ simply as H_t. Since

$$\frac{\delta^2 J_t}{\delta^2 x_t} = \Sigma^{-1} = H_t^t Q_t^{-1} H_t + \overline{\Sigma}^{-1},$$

we have (as with the linear Kalman case) that

$$\Sigma_t H_t^t Q_t^{-1}(z_t - h(\overline{\mu}_t)) = (x_t - \overline{\mu}_t).$$

Solving for x_t gives

$$x_t = \overline{\mu}_t + \Sigma_t H_t^t Q_t^{-1}(z_t - h(\overline{\mu}_t)).$$

This expresses the mean of $bel(x_t)$ as

$$\mu_t = \overline{\mu}_t + \Sigma_t H_t^t Q_t^{-1}(z_t - h(\overline{\mu}_t)) = \overline{\mu}_t + K_t(z_t - h(\overline{\mu}_t)),$$

where $K_t = \Sigma_t H_t^t Q_t^{-1}$ is the Kalman gain. Therefore we have

$$bel(x_t) \sim N(\overline{\mu}_t + K_t(z_t - h(\overline{\mu}_t)), (H_t^t Q_t^{-1} H_t + \overline{\Sigma}^{-1})^{-1}),$$

but as in the linear case, replacing C_t by H_t, we can express the covariance Σ_t as

$$\Sigma_t = (H_t^t Q_t^{-1} H_t + \overline{\Sigma}^{-1})^{-1} = (I - K_t H_t)\overline{\Sigma}_t.$$

The EKF prediction and update steps therefore proceed as for the linear Kalman Filter, but at every step we must be able to calculate the values $g(u_t, \mu_{t-1})$ and $h(\overline{\mu}_t)$, and to evaluate their Jacobians $G_t|_{\mu_{t-1}}$ and $H_t|_{\overline{\mu}_t}$.

The EKF will fail if there are significant nonlinearities in the function $g_{u_t}(x_{t-1})$ in the support of $bel(x_{t-1})$, or in the function $h(x_t)$ in the support of $\overline{bel}(x_t)$, since the assumption that the true posterior beliefs are nearly Gaussian will not hold. The EKF is not applicable to these problems; other Bayesian filter techniques (particle filtering, for example) are able to handle this case.

References

[1] Thrun, S., Burgard, W., Fox, D., *Probabilistic Robotics.* MIT Press, 2006.

[2] Johnston, C., "Derivation of the Bayes and Kalman Filters", 2019.
Posted online at: https://carolynpjohnston.files.wordpress.com/2019/12/20191202_kalmanfilter.pdf.