The Kalman Filter

Josiah Hanna

February 4, 2025

Background: Bayes Filter, Gaussian distributions, Taylor Expansions, Jacobian matrix. See appendix A for a review.

1 Introduction

The last reading introduced the Bayes Filter as a means to compute a robot's posterior belief, $p(x_t|z_{1:t}, u_{1:t})$, online, using recursive updates in which computation does not increase with t. At every time-step, the Bayes Filter alternates the two steps:

1. **Prediction:** The prior belief about the state is updated based on the system dynamics:

$$\overline{\texttt{bel}}(x_t) \leftarrow \int_{x_{t-1}} p(x_t \mid u_t, x_{t-1}) \cdot \texttt{bel}(x_{t-1})\, dx_{t-1}.$$

2. **Update:** The prior belief is refined using sensor observations.

$$\texttt{bel}(x_t) \leftarrow \eta \cdot p(z_t \mid x_t) \cdot \overline{\texttt{bel}}(x_t).$$

When the number of possible states is finite, the integral is replaced with a summation. A main limitation of Bayes Filter is that the updates in the prediction and update step are not tractable if the number of possible states is large or states are continuous. While the Bayes Filter is intractable in general, there are special cases where it can be applied even with continuous states. In this reading, we will introduce the Kalman Filter which will enable exact implementation of a Bayes Filter for the special case of a linear state transition and observation model with Gaussian uncertainty. Furthermore, the Kalman Filter assumes that the belief distribution is Gaussian, which allows it to be fully described by its mean and covariance. These assumptions enable the use of linear algebra to efficiently compute the optimal state estimate through the prediction and update steps, making the Kalman Filter a computationally efficient realization of the Bayes Filter. The Kalman Filter is an optimal recursive algorithm used for estimating the state of a dynamic system from a series of noisy measurements. In addition to robotics, it is widely applied in control systems and finance.

2 System Model

The basic Kalman Filter makes specific assumptions about the robot's environment. Specifically, we assume linear Gaussians for both state transitions and sensor observations. Formally, we have that:

$$x_t = Ax_{t-1} + Bu_t + w_t, \tag{1}$$

$$z_t = Hx_t + v_t, \tag{2}$$

where:

- x_t is the vector of state variables at time t,
- u_t is the vector of control inputs at time t,
- z_k is the vector of sensor observations at time t,
- A is the state transition matrix,
- B is the control matrix,
- H is the observation matrix,
- $w_t \sim \mathcal{N}(0, Q)$ is noise in the robot's state transition,
- $v_t \sim \mathcal{N}(0, R)$ is noise in the robot's observation.

Note that we are assuming both transition noise and observation noise have Gaussian distributions with mean $\mathbf{0}$ and covariance matrices Q and R respectively. Ignoring the noise, we can see that both state transitions and observations are linear in x_t and u_t. Taken together, we are assuming that state transitions are produced by a linear transformation of the state and control at time t followed by adding Gaussian noise. A similar process is used to determine the observation at time t. This is the *linear-Gaussian* assumption.

Another way of writing the linear-Gaussian assumption is that $p(x_t|x_{t-1}, u_t) = \mathcal{N}(x_t; Ax_{t-1} + Bu_t, Q)$ and $p(z_t|x_t) = \mathcal{N}(z_t; Hx_t, R)$.

3 Kalman Filter

We are now ready to instantiate the Bayes Filter under the specific assumptions from the previous section. One final assumption we must make is that the robot's initial belief about its state is Gaussian: $\texttt{bel}(x_0) = \mathcal{N}(x; \mu_0, \Sigma_0)$. This assumption and the assumption of a linear-Gaussian environment means that the robot's belief will maintain a Gaussian form under Kalman Filter updates: $\texttt{bel}(x_t) = \mathcal{N}(x; \mu_t, \Sigma_t)$.

3.1 Prediction Step

First, we compute the posterior belief about the next state given the robot's control, u_t, but before incorporating the observation:

$$\bar{\mu}_t = A\mu_{t-1} + Bu_t \tag{3}$$

$$\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + Q \tag{4}$$

where:

- $\bar{\mu}_t$ is the predicted mean of the belief distribution before the sensor observation is incorporated.
- $\bar{\Sigma}_t$ is the predicted covariance of the belief distribution.

3.2 Update Step

We then incorporate the sensor observation to produce the final posterior belief, $\texttt{bel}(x_t)$.

$$K_t = \bar{\Sigma}_t H^T (H\bar{\Sigma}_t H^T + R)^{-1} \tag{5}$$

$$\bar{\mu}_t = \bar{\mu}_t + K_t(z_t - H\bar{\mu}_t) \tag{6}$$

$$\Sigma_t = (I - K_k H)\bar{\Sigma}_t \tag{7}$$

where:

- K_t is known as the Kalman gain. The Kalman gain balances how much to trust the new observation instead of the prediction alone. If R is large then the measurement is very noisy and hence less trustworthy; consequently, K_t will be small.
- $\bar{\mu}_t$ is the updated state estimate.
- Σ_t is the updated covariance.

Following these updates, we have that the robot's belief will be $\texttt{bel}(x_t) = \mathcal{N}(x_t; \mu_t, \Sigma_t)$.

This prediction and update step are an exact implementation of a Bayes Filter under specific assumptions on the system model and initial belief. That is, the Kalman prediction and update steps are obtained from plugging the linear Gaussian system model into the Bayes Filter update steps.

Algorithm 1 Kalman Filter

Input: Previous belief represented as a Gaussian with mean, μ_{t-1}, and covariance, Σ_{t-1}, control input u_t, measurement z_t
Output: Updated belief represented as a Gaussian with mean, μ_t, and covariance Σ_t.
procedure KalmanFilter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)
    // Prediction step
    $\bar{\mu}_t = A\mu_{t-1} + Bu_t$
    $\bar{\Sigma}_t = A\Sigma_{t-1}A^\top + Q$
    // Correction step
    $K_t = \bar{\Sigma}_t H^T (H\bar{\Sigma}_t H^T + R)^{-1}$ // Compute Kalman gain
    $\bar{\mu}_t = \bar{\mu}_t + K_t(z_t - \bar{\mu}_t)$
    $\Sigma_t = (I - K_k H)\bar{\Sigma}_t$
    return μ_t, Σ_t
end procedure

3.3 Strengths and Weaknesses

The Kalman Filter has the following strengths:

- The Kalman Filter inherits the optimality of a Bayes Filter assuming the system is actually linear with Gaussian noise.
- Like a Bayes Filter, the Kalman Filter is recursive which implies that computation does not grow with t. Furthermore, the linear algebra computations can be done in Polynomial time and are typically fast with modern linear algebra libraries (e.g., Eigen or Numpy).
- Like a Bayes Filter, the Kalman Filter provides uncertainty estimates through the covariance, Σ_t. If the determinant of Σ_t is large then the robot is uncertain about the true state. If the determinant of Σ_t is small then the robot is more certain that μ_t is its current state.

These advantages are offset by some important weaknesses:

- The Kalman Filter assumes Gaussian noise which limits its applicability to systems with non-Gaussian noise.
- If the system is highly nonlinear then the Kalman Filter will compute a sub-optimal belief.
- With a Kalman Filter, the belief is always Gaussian which means the robot cannot represent multi-modal uncertainty. For example, the robot cannot represent the belief that it is located in one of two locations.

4 Extended Kalman Filter

The Extended Kalman Filter (EKF) is a nonlinear extension of the Kalman Filter that linearizes the system dynamics using a first-order Taylor expansion (see the appendix for a quick review of what this is). The EKF mitigates the KF's assumption that the system is linear.

4.1 Nonlinear System Model

For a nonlinear system:

$$x_t = f(x_{t-1}, u_t) + w_t, \tag{8}$$

$$z_t = h(x_t) + v_t, \tag{9}$$

where f and h are nonlinear functions describing the state transition and observation models, respectively. Alternatively, we can write that $x_t \sim \mathcal{N}(f(x_{t-1}, u_t), Q)$ and $z_t \sim \mathcal{N}(h(x_t), R)$.

4.2 Linearization Using Jacobians

The key idea of the EKF is to make the system approximately linear by replacing the nonlinear f and h with a linear approximation. This is done using the method of Taylor series expansion (see Appendix A for a brief review). To apply the Kalman filter equations, the system is linearized about the current state estimate, μ_{t-1}. Specifically, we define matrices A_t and H_t as the Jacobian matrices of f and h respectively, both evaluated at $x = \mu_{t-1}$.

$$A_t = \frac{\partial f}{\partial x}\Big|_{x=\mu_{t-1}}, \quad H_t = \frac{\partial h}{\partial x}\Big|_{x=\mu_{t-1}}. \tag{10}$$

In multivariate calculus, the Jacobian matrix represents the first-order partial derivatives of a vector-valued function. Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$, the Jacobian matrix J is defined as:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}. \tag{11}$$

Each row represents one of the outputs of f and each column represents one of the inputs. Intuitively, each entry, i, j, captures how fast output i is changing as input j changes.

4.3 Prediction and Update

The EKF follows very similar updates to the Kalman Filter:

$$\bar{\mu}_t = f(\mu_t, u_t) \tag{12}$$

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^\top + Q \tag{13}$$

We then compute the Kalman gain:

$$K_t = \overline{\Sigma}_t H_t^\top (H_t \overline{\Sigma}_t H_t^\top + Q)^{-1} \tag{14}$$

And finally, apply the updates:

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)) \tag{15}$$

$$\Sigma_t = (I - K_t H_t)\overline{\Sigma}_t \tag{16}$$

5 Strengths and Weaknesses of Kalman Filters

5.1 Extended Kalman Filter (EKF)

Strengths:

- Extends the Kalman Filter to nonlinear systems.
- Works well for systems that can be *locally* linearized. The Taylor expansion is only an accurate representation of the transition/observation function for x close to μ_t. The closer these functions are to linear, the more accurate the Taylor expansion approximation will be.

Weaknesses:

- Accuracy depends on the quality of linearization. See discussion above. Highly nonlinear systems will have poor accuracy with an EKF.
- Computationally more expensive than the standard Kalman Filter due to need to compute Jacobians.
- May diverge if the system is highly nonlinear.

6 Conclusion

The Kalman Filter and its nonlinear extension, the Extended Kalman Filter, provide powerful tools for state estimation in dynamic systems. Their recursive nature makes them computationally efficient for real-time applications.

A Review

The following appendices provide background on some mathematical concepts used in these notes. If you are familiar with these topics, you can skip over them.

A.1 Gaussian Distributions

The Gaussian (or multi-variate normal) distribution is one of the most common probability distributions for continuous random variables. Let $x \in \mathbf{R}^d$ be a d-dimensional vector. We say that x has a Gaussian distribution if:

$$p(x) = \frac{1}{\sqrt{2\pi \det \Sigma}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)},$$

where $\mu \in \mathbf{R}^d$ is called the mean vector and $\Sigma \in \mathbf{R}^{d \times d}$ is the covariance matrix. The matrix Σ must be positive-definite.

A few additional notes on the Gaussian distribution:

1. When $d = 1$, the Gaussian distribution is the familiar bell curve.
2. Under a Gaussian distribution, $p(x)$ is maximized when $x = \mu$, i.e., the mean is the most likely outcome.
3. The covariance controls the spread of $p(x)$. When the covariance is large, then $p(x)$ is spread out and values of x that are farther from μ receive more probability density. When the covariance is small then most sampled values of x will be close to μ.

For notation, we will denote Gaussian distributions as $\mathcal{N}(\mu, \Sigma)$ and the density of x under a Gaussian with mean μ and covariance Σ as $\mathcal{N}(x; \mu, \Sigma)$.

A.2 Jacobian Matrix

In multivariate calculus, the Jacobian matrix represents the first-order partial derivatives of a vector-valued function. Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$, the Jacobian matrix J is defined as:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}. \tag{17}$$

The Jacobian is used in the Extended Kalman Filter to linearize nonlinear functions by approximating them with a first-order Taylor series expansion.

A.3 Taylor Series Expansion

The Taylor series is a mathematical tool used to approximate a function by expanding it around a given point. For a function $f(x)$ that is differentiable, the first-order Taylor expansion around a fixed point x_0 is given by:

$$f(x) \approx f(x_0) + \frac{df}{dx}\Big|_{x_0}(x - x_0). \tag{18}$$

Note that the right hand side of Equation (18) is a linear function in x since $\frac{df}{dx}\Big|_{x_0}$ and $f(x_0)$ are constant with respect to x. The approximation is perfect for $x = x_0$ and will generally become worse as x moves away from x_0. If f is linear ($f(x) = ax + b$) then the approximation is again perfect since $f(x) \approx ax_0 + b + a(x - x_0) = ax + b = f(x)$. However, for nonlinear f, the approximation has error that increases as $|x - x_0|$ grows.

In the case of multiple variables, the Taylor expansion generalizes to:

$$f(x) \approx f(x_0) + J(x_0)(x - x_0), \tag{19}$$

where $J(x_0)$ is the Jacobian matrix evaluated at x_0. The Taylor expansion is useful in the Extended Kalman Filter because it allows nonlinear functions to be locally approximated by linear functions, enabling the use of linear filtering techniques.