Derivation of the Bayes and Kalman filters

Carolyn Johnston

December 2, 2019

1 Bayes Filters

This writeup follows Chapters 2.4 (Bayes Filters) and 3.2 (the Kalman Filter) in "Probabilistic Robotics" [1], but adds more mathematical detail. It can be read as a companion to that chapter, or as a stand-alone document. The Kalman filter is a special case of a Bayes filter, which is where we'll start.

The state of a moving vehicle or robot consists of its position and orientation variables, and may include other relevant aspects of its condition (e.g. wipers on or anti-lock brakes engaged). Generally the precise state of a system is not known, but is inferred using its internal control systems (process model) and independent measurements. A Bayes filter is a recursive solution for estimating state probability density functions as they evolve over time. Using a Bayes filter, the state of a moving vehicle can be tracked, using control data such as odometry, and incoming measurements from sensors such as lidar or cameras.

We start by assuming that the true state is a time series of random vectors: we use the notation $x_{0:T} = \{x_0, ..., x_T\}$ to denote the full time series of state vectors, where x_t is the state vector at time t. Each transition $x_{t-1} \to x_t$ is driven by a (deterministic) control vector u_t. At each time step, a random sensor measurement vector z_t is collected; the full time series of measurement vectors is denoted $z_{\{0:T\}} = \{z_0, ..., z_T\}$.

For example, for the problem of tracking the location of a vehicle on a highway, the state x_t might consist of its position vector relative to a fixed earth frame, together with an orientation (Euler angles, or a quaternion) relative to the frame. The vector u_t might be odometry input from the vehicle's internal control system, and z_t might consist of GPS readings or lidar observations of landmarks (known or unknown).

The joint probability density function (p.d.f.) of all the states and measurements is $f(x_{0:T}, z_{1:T}|u_{1:T})$, but we are interested in the p.d.f. of the *final* state conditioned on all previous states, and all measurements: $f(x_T|x_{0:T-1}, z_{1:T}, u_{1:T})$.

The Bayes filter formulation assumes the state process is Markov. A Markov chain is a state process for which all future states are conditionally independent of past states and measurements, given the present state: i.e.,

$$f(x_T|x_{0:T-1}, z_{1:T}, u_{1:T}) = f(x_T|x_{T-1}, u_T). \tag{1}$$

It is further assumed that

$$f(z_T|x_{0:T}, z_{1:T-1}, u_{1:T}) = f(z_T|x_T), \tag{2}$$

i.e. the current measurement, given all past measurements and all states, depends only on the current state x_T.

It follows that the system evolves according to a Hidden Markov Model. In other words, the joint p.d.f. simplifies to:

$$f(x_0, ..., x_T, z_1, ..., z_T) = f(x_0) \cdot \Pi_{t=1}^{T} f(x_t|x_{t-1}) f(z_t|x_t). \tag{3}$$

2 Derivation of the general Bayes Filter algorithm

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 order to run the Bayes filter, we need the following:

1. The initial state probability density function, $f(x_0)$;
2. Sensor measurements z_t;
3. Control measurements u_t;
4. The p.d.f. of sensor measurements at time t, conditioned on the state at time t, $f(z_t|x_t)$ (this is called the *sensor model*);
5. The p.d.f. of the state vector at time t, conditioned on the state at time $t-1$ and the control measurement at time t, $f(x_t|x_{t-1}, u_t)$ (this is called the *transition model*).

The derivation of the Bayes filter proceeds by induction. First, we claim that $bel(x_0) = f(x_0)$; at time $t = 0$ we have not yet seen any control or sensor measurements, so the belief is simply equal to the prior for x_0.

Next, assume that we are given $bel(x_{T-1}) = f(x_{T-1}|z_{1:T-1}, u_{1:T-1})$; we will show how $bel(x_T) = f(x_T|z_{1:T}, u_{1:T})$ is calculated using data u_T and z_T, the sensor measurement model $f(z_T|x_T)$, and the transition model $f(x_T|x_{T-1}, u_T)$.

We work backward from the final target distribution, $bel(x_T) = f(x_T|z_{1:T}, u_{1:T})$. By Bayes' theorem, applied to x_T and z_T,

$$f(x_T|z_{1:T}, u_{1:T}) = f(x_T|z_T, z_{1:T-1}, u_{1:T}) = \frac{f(z_T|x_T, z_{1:T-1}, u_{1:T}) \cdot f(x_T|z_{1:T-1}, u_{1:T})}{f(z_T|z_{1:T-1}, u_{1:T})}.$$

Notice that the denominator term is a constant with respect to x_T, so set $\eta = 1/f(z_T|z_{1:T-1}, u_{1:T})$ to get

$$f(x_T|z_{1:T}, u_{1:T}) = \eta \cdot f(z_T|x_T, z_{1:T-1}, u_{1:T}) \cdot f(x_T|z_{1:T-1}, u_{1:T}). \tag{4}$$

From equation 2, we have

$$f(z_T|x_T, z_{1:T-1}, u_{1:T}) = f(z_T|x_T);$$

and the second term on the right hand side of equation 4, $(x_T|z_{1:T-1}, u_{1:T})$, is equal to the intermediate belief $\overline{bel}(x_T)$. Thus the second step of the recursive Bayes filter formula is

$$f(x_T|z_{1:T}, u_{1:T}) = \eta \cdot f(z_T|x_T) \cdot \overline{bel}(x_T). \tag{5}$$

Since we are given data z_T and the sensor model $f(z_T|x_T)$, if we know $\overline{bel}(x_T)$, then the above shows that we can execute the second step of the Bayes filter.

Working backward to the first step of the Bayes filter, we calculate

$$\overline{bel}(x_T) = f(x_T|z_{1:T-1}, u_{1:T}) = f(x_T|u_T, z_{1:T-1}, u_{1:T}),$$

which incorporates the control measurement u_T.

To do this, we re-introduce the previous state x_{T-1}, and then marginalize over it:

$$\overline{bel}(x_T) = f(x_T|z_{1:T-1}, u_{1:T}) = \int f(x_T, x_{T-1}|z_{1:T-1}, u_{1:T}) dx_{T-1}.$$

Therefore, using $P(A) = P(A|B) \cdot P(B)$ inside the integrand, we have:

$$\overline{bel}(x_T) = \int f(x_T, x_{T-1}|z_{1:T-1}, u_{1:T}) dx_{T-1} =$$

$$\int f(x_T|x_{T-1}, z_{1:T-1}, u_{1:T}) \cdot f(x_{T-1}|z_{1:T-1}, u_{1:T}) dx_{T-1}.$$

The first term in the integrand simplifies according to equation 1. Note that the second term in the integrand satisfies

$$f(x_{T-1}|z_{1:T-1}, u_{1:T}) = f(x_{T-1}|z_{1:T-1}, u_{1:T-1}),$$

i.e., the p.d.f. for x_{T-1} at time step $T-1$ is independent of u_T, the control at time T. Thus the second term in the integrand is equal to $bel(x_{T-1})$, and we have

$$\overline{bel}(x_T) = \int f(x_T|x_{T-1}, u_T) \cdot bel(x_{T-1}) dx_{T-1}. \tag{6}$$

Equations 5 and 6 together form the Bayes filter algorithm to transition from $bel(x_{T-1})$ to $bel(x_T)$.

The first step of the Bayes filter algorithm, calculating $\overline{bel}(x_T)$, is essentially a convolution of the previous belief with the transition p.d.f. in equation 1; the previous belief is essentially smeared by the transition p.d.f., and so $\overline{bel}(x_T)$ has increased uncertainty relative to $bel(x_{T-1})$. 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 reduces the uncertainty of $\overline{bel}(x_T)$ in accordance with the sensor measurement.

3 The Kalman filter

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:

$$x_t = A_t x_{t-1} + B_t u_u + \epsilon_t \tag{7}$$

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

$$z_t = C_t x_t + \delta_t. \tag{8}$$

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, which we will show.

It follows from equation 7 that the transition model p.d.f. is given by

$$f(x_t|x_{t-1}, u_t) \propto \exp(\frac{-(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)}{2}),$$

where R_t is the covariance of ϵ_t. Similarly, it follows from equation 8 that the sensor model p.d.f. is given by

$$f(z_t|x_t) \propto \exp(\frac{-(z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t)}{2}),$$

where Q_t is the covariance of δ_t.

4 Derivation of the prediction step of the Kalman filter

The Kalman filter, as a special case of the Bayes filter, has two steps, mirroring the prediction equation 6 and the sensor update equation 5. We prove that each belief $bel(x_t)$ is Gaussian, and derive the equations by induction on t.

We are given that $bel(x_0)$ is Gaussian, with $bel(x_0) \sim N(\mu_0, \Sigma_0)$.

Now, suppose that $bel(x_{t-1})$ is Gaussian, with $bel(x_{t-1}) \sim N(\mu_{t-1}, \Sigma_{t-1})$. From equation 6, we have that the intermediate belief is

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

$$\int \exp(-\frac{1}{2}[(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t) + (x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1})]) dx_{t-1}. \tag{9}$$

The result for the prediction step follows directly from the convolution theorem for Gaussians, which is proven in Section (6), after the substitutions $x = x_t$, $y = x_{t-1}$, $A = A_t$, $w = B_t u_t$,$\mu = \mu_{t-1}$, $R = R_t$, and $\Sigma = \Sigma_{t-1}$.

$$\overline{bel}(x_t) \sim N(A_t \mu_{t-1} + B_t u_t, A_t \Sigma_{t-1} A_t^T + R_t) = N(\overline{\mu}_t, \overline{\Sigma}_t).$$

Note that where the covariance of the Gaussian $bel(x_{t-1})$ was Σ_{t-1}, that of $\overline{bel}(x_t)$ is $A_t \Sigma_{t-1} A_t^T + R_t$. Essentially, we have propagated the uncertainty in the state x_{t-1}through the linear function A_t, and added the process noise from ϵ_t. $\overline{bel}(x_t)$ is a multivariate Gaussian with parameters $\overline{\mu_t} = A_t \mu_{t-1} + B_t u_t$, and $\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$.

5 Derivation of the update step of the Kalman filter

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

$$bel(x_t) = \eta p(z_t|x_t) \overline{bel}(x_t),$$

where η is a constant with respect to x_t. From the prediction step, we have $\overline{bel}(x_t) \sim N(\overline{\mu}_t, \overline{\Sigma}_t)$. We are given $p(z_t|x_t) \sim N(C_t x_t, Q_t)$. The product is given by $bel(x_t) = \eta \exp(-J_t)$, with

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

This expression is quadratic in x_t, and so $bel(x_t)$ is a Gaussian in x_t. To determine its parameters, we calculate the first two derivatives with respect to x_t.

We have

$$\frac{dJ_t}{dx_t} = -C_t^T Q_t^{-1} (z_t - C_t x_t) + \overline{\Sigma}_t^{-1} (x_t - \overline{\mu}_t). \tag{10}$$

and

$$\frac{d^2 J_t}{dx_t^2} = C_t^T Q_t^{-1} C_t + \overline{\Sigma}_t^{-1}. \tag{11}$$

By equation 11, the covariance of $bel(x_t)$ is

$$\Sigma_t = (C_t^T Q_t^{-1} C_t + \overline{\Sigma}_t^{-1})^{-1}. \tag{12}$$

To find the mean of $bel(x_t)$, we solve the following equation for x_t:

$$\frac{dJ_t}{dx_t} = -C_t^T Q_t^{-1}(z_t - C_t x_t) + \overline{\Sigma}_t^{-1}(x_t - \overline{\mu}_t) = 0. \tag{13}$$

This is easy enough, but it is traditional to massage the solution μ_t so that it is in the form

$$\mu_t = \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t),$$

for K_t some matrix. This expresses the mean as the sum of the prediction mean, $\overline{\mu}_t$, and the difference of the sensor measurement and the predicted sensor measurement, $z_t - C_t \overline{\mu}_t$, weighted by K_t.

To do the massaging, notice that equation 13 implies that

$$C_t^T Q_t^{-1}(z_t - C_t x_t) = \overline{\Sigma}_t^{-1}(x_t - \overline{\mu}_t).$$

We add $0 = C_t \overline{\mu}_t - C_t \overline{\mu}_t$ to the left hand side argument to get

$$C_t^T Q_t^{-1}(z_t - C_t \overline{\mu}_t + C_t \overline{\mu}_t - C_t x_t) = \overline{\Sigma}_t^{-1}(x_t - \overline{\mu}_t)$$

$$\implies C_t^T Q_t^{-1}(z_t - C_t \overline{\mu}_t) = (C_t^T Q_t^{-1} C_t + \overline{\Sigma}_t^{-1})(x_t - \overline{\mu}_t) = \Sigma_t^{-1}(x_t - \overline{\mu}_t),$$

where the last equality comes from equation 12. It follows that

$$x_t = \overline{\mu}_t + \Sigma_t C_t^T Q_t^{-1}(z_t - C_t \overline{\mu}_t),$$

is the solution of equation 13, and therefore we have

$$\mu_t = \overline{\mu}_t + \Sigma_t C_t^T Q_t^{-1}(z_t - C_t \overline{\mu}_t) = \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t), \tag{14}$$

with weight matrix $K_t = \Sigma_t C_t^T Q_t^{-1}$.

It also turns out, after more manipulation (shown in Section (7)), that

$$K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}, \tag{15}$$

and (using the Woodbury identity) that

$$\Sigma_t = (I - K_t C_t)\overline{\Sigma}_t.$$

Note that in the expression 15 , the weight given by $\overline{\Sigma_t}$ is offset by that of $C_t \overline{\Sigma}_t C_t^T + Q_t$. If the sensor noise Q_t is large with respect to the transition noise $\overline{\Sigma}_t$, then $(C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$ (therefore K_t) is small compared to $\overline{\Sigma_t}$. Thus the effect of the measurement offset $(z_t - C_t\overline{\mu}_t)$ is reduced relative to that of the predicted mean $\overline{\mu}$.

Conversely, if Q_t is small with respect to $\overline{\Sigma}_t$, then $(C_t\overline{\Sigma}_tC_t^T + Q_t)^{-1} \approx (C_t\overline{\Sigma}_tC_t^T)^{-1}$, and so $C_tK_t \approx C_t\overline{\Sigma}_tC_t^T(C_t\overline{\Sigma}_tC_t^T)^{-1} = I$.

It follows that

$$C_t\mu_t = C_t\overline{\mu_t} + C_tK_t(z_t - C_t\overline{\mu_t}) \approx C_t\overline{\mu}_t + I(z_t - C_t\overline{\mu}_t) = z_t,$$

so that the mean μ_t of $bel(x_t)$ respects the more accurate sensor measurement.

6 A convolution theorem for Gaussians

The following convolution identity was used in the derivation of the Kalman filter and will be derived here:

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

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

Proof. The proof proceeds by expressing the quadratic argument of the exponential in the integrand,

$$k(x,y) = (x - Ay - w)^TR^{-1}(x - Ay - w) + (y - \mu)^T\Sigma^{-1}(y - \mu),$$

as:

$$k(x,y) = (y - u(x))^TT^{-1}(y - u(x)) + \beta(x), \qquad (16)$$

where v is a constant with respect to both x and y, T is some covariance matrix, $u(x)$ is linear in x, and $\beta(x)$ is a function of x, independent of y.

Then

$$\int \exp(-\frac{1}{2}[(y - u(x))^TT^{-1}(y - u(x)) + \beta(x)])dy =$$

$$\exp(-\frac{1}{2}[\beta(x)) \cdot \int \exp(\text{-}\tfrac{1}{2}[(y - u(x))^TT^{-1}(y - u(x))])dy$$

$$\propto \exp(-\frac{1}{2}[\beta(x)]),$$

since the integral is a constant multiple of a p.d.f. in y. We will also show that the function $\beta(x)$ is quadratic in x alone: $\beta(x) = (x - v)^TL^{-1}(x - v)$ for some L, where v is constant with respect to both x and y; in fact, we will see that $v = A\mu + w$, and $L = A\Sigma A^T + R$.

We proceed by finding a quadratic function, $\alpha(y)$, such that the first and second derivatives of $\alpha(y)$ are equal to those of $k(x, y)$. Since $k(x, y)$ is quadratic in y, it follows that all the y-derivatives of α and k are equal, and therefore $\beta(x) = k(x, y) - \alpha(y)$ is constant with respect to y.

We have

$$\frac{dk(x,y)}{dy} = -2[A^T R^{-1}(x - Ay - w) + \Sigma^{-1}(y - \mu)] \qquad (17)$$

and

$$\frac{d^2k(x,y)}{dy^2} = -2A^T R^{-1}A - 2\Sigma^{-1}.$$

Solving $\frac{dk(x,y)}{dy} = 0$ for the constant term $u(x)$ gives

$$(A^T R^{-1}A + \Sigma^{-1})y_0 = \Sigma^{-1}\mu + A^T R^{-1}(x - w),$$

or

$$y_0 = u(x) = (A^T R^{-1}A + \Sigma^{-1})^{-1}(\Sigma^{-1}\mu + A^T R^{-1}(x - w)).$$

Therefore, defining

$$\alpha(y) = (y - u(x))^T T^{-1}(y - u(x)) = (y - y_0)^T (A^T R^{-1}A + \Sigma^{-1})(y - y_0)$$

guarantees that $\beta(x) = k(x, y) - \alpha(y)$ is constant with respect to y. For computational convenience, we'll define

$$P \equiv (A^T R^{-1}A + \Sigma^{-1})$$

in what follows, so that $y_0 = u(x) = P^{-1}(\Sigma^{-1}\mu + A^T R^{-1}(x - w))$ and

$$\alpha(y) = (y - y_0)^T P(y - y_0).$$

Next, we must find $\beta(x)$. We will begin by calculating $\frac{d\beta}{dx}$ and $\frac{d\beta^2}{dx^2}$, and will see that $\frac{d\beta^2}{dx^2}$ is constant with respect to x; therefore, $\beta(x)$ is a quadratic function of x.

We have

$$\beta \equiv k(x, y) - \alpha(y) = (x - Ay - w)^T R^{-1}(x - Ay - w)$$
$$+(y - \mu)^T \Sigma^{-1}(y - \mu) - (y - u(x))^T P(y - u(x)).$$

Since we have shown that β is independent of y, we can remove the explicit dependency on y by setting $y \equiv 0$, which gives:

$$\beta = (x - w)^T R^{-1}(x - w) + \mu^T \Sigma^{-1}\mu - (u(x)^T P u(x)).$$

Taking the first derivative with respect to x gives:

$$\frac{d\beta}{dx} = 2R^{-1}(x-w)-2(A^TR^{-1})^TP^{-1}(P)P^{-1}(\Sigma^{-1}\mu+A^TR^{-1}(x-w))$$
$$= 2R^{-1}(x-w) - 2R^{-1}AP^{-1}(\Sigma^{-1}\mu + A^TR^{-1}(x-w)).$$

Setting $\frac{d\beta}{dx} = 0$ to get the extremum of β gives

$$(R^{-1} - R^{-1}AP^{-1}A^TR^{-1})(x-w) = R^{-1}AP^{-1}\Sigma^{-1}\mu.$$

Fortunately, by the Woodbury formula, we have

$$R^{-1} - R^{-1}AP^{-1}A^TR^{-1} = R^{-1} - R^{-1}A(A^TR^{-1}A + \Sigma^{-1})^{-1}A^TR^{-1}$$
$$= (R + A\Sigma A^T)^{-1},$$

so that

$$(R + A\Sigma A^T)^{-1}(x-w) = R^{-1}A(A^TR^{-1}A + \Sigma^{-1})^{-1}\Sigma^{-1}\mu.$$

It follows that

$$x = w + (R + A\Sigma A^T)R^{-1}A(A^TR^{-1}A + \Sigma^{-1})^{-1}\Sigma^{-1}\mu. \tag{18}$$

Fortunately, more simplification is possible. We have

$$(A^TR^{-1}A + \Sigma^{-1})^{-1}\Sigma^{-1} = (\Sigma A^TR^{-1}A + I)^{-1},$$

and

$$(R + A\Sigma A^T)R^{-1}A = (A + A\Sigma A^TR^{-1}A) = A(I + \Sigma A^TR^{-1}A).$$

Making these substitutions in equation 18 gives

$$x = w + (R + A\Sigma A^T)R^{-1}A(A^TR^{-1}A + \Sigma^{-1})^{-1}\Sigma^{-1}\mu$$
$$= w + A(I + \Sigma A^TR^{-1}A)(I + \Sigma A^TR^{-1}A)^{-1}\mu = w + A\mu.$$

Thus, the extremum of the function $\beta(x)$ is the very simple expression $x_0 = w + A\mu$.

To establish the quadratic function β, we still need $\frac{d^2\beta}{d^2x}$.

$$\frac{d^2\beta}{dx^2} = \frac{d}{dx}(2R^{-1}(x-w) - 2R^{-1}AP^{-1}(\Sigma^{-1}\mu + A^TR^{-1}(x-w)))$$
$$= 2(R^{-1} - 2R^{-1}AP^{-1}A^TR^{-1}) = 2(R + A\Sigma A^T)^{-1},$$

again using the Woodbury identity. Since the second derivative is a constant with respect to x, $\beta(x)$ is a quadratic function which must be of the form $N(w + A\mu, R + A\Sigma A^T)$. This completes the proof of the convolution identity.

7 Deriving the Kalman gain expressions

In Section (5), we stated that the Kalman gain $K_t = \Sigma_t C_t^T Q_t^{-1}$ could be rewritten in terms of $\overline{\Sigma}_t$ as $K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$, and that we could write Σ_t as $(I - K_t C_t)\overline{\Sigma}_t$. In this section I reproduce those calculations from Chapter 3.2 in [1].

First, we show how we can express K_t without using Σ_t. We start with

$$K_t = \Sigma_t C_t^t Q_t^{-1}$$
$$= [\Sigma_t C_t^t Q_t^{-1} (C_t \overline{\Sigma}_t C_t^T + Q_t)](C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1},$$

where we have multiplied by the identity $(C_t \overline{\Sigma}_t C_t^T + Q_t)(C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$ on the right.

Multiplying out the expression in brackets on the left of this expression gives:

$$\Sigma_t C_t^t Q_t^{-1} (C_t \overline{\Sigma}_t C_t^T + Q_t) = \Sigma_t (C_t^t Q_t^{-1} C_t \overline{\Sigma}_t C_t^T + C_t^t).$$

The goal of the exercise now is to get rid of the Σ_t on the left. We do this by "pulling out" Σ_t^{-1} from the expression in parentheses, so that it cancels Σ_t.

Recall from equation (12) that $\Sigma_t^{-1} = C_t^T Q_t^{-1} C_t + \overline{\Sigma}_t^{-1}$. It follows that

$$\Sigma_t (C_t^t Q_t^{-1} C_t \overline{\Sigma}_t C_t^T + C_t^t) = \Sigma_t (C_t^t Q_t^{-1} C_t \overline{\Sigma}_t C_t^T + (\overline{\Sigma}_t^{-1} \overline{\Sigma}_t) C_t^t)$$
$$= \Sigma_t (C_t^t Q_t^{-1} C_t + \overline{\Sigma}_t^{-1}) \cdot (\overline{\Sigma}_t C_t^t) = \Sigma_t \Sigma_t^{-1} \overline{\Sigma}_t C_t^t = \overline{\Sigma}_t C_t^t.$$

This shows that

$$K_t = \overline{\Sigma}_t C_t^t (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}. \tag{19}$$

Second, we show that $\Sigma_t = (I - K_t C_t)\overline{\Sigma}_t$. We have

$$\Sigma_t = (C_t^T Q_t^{-1} C_t + \overline{\Sigma}_t^{-1})^{-1}.$$

From the Woodbury Identity, we have

$$\Sigma_t = (C_t^T Q_t^{-1} C_t + \overline{\Sigma}_t^{-1})^{-1} = \overline{\Sigma}_t - \overline{\Sigma}_t C_t^t (Q_t + C_t \overline{\Sigma}_t C_t^t)^{-1} C_t \overline{\Sigma}_t.$$

Factoring out $\overline{\Sigma}_t$ on the right, and using equation (19) gives the result:

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

References

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