9.7 Filter Initialization

The Kalman filter requires an initial estimate $\hat{\boldsymbol{x}}(0)$ and an associated error covariance matrix $P(0)$.

How do we obtain $\hat{\boldsymbol{x}}(0)$ and $P(0)$?

1. We may be able to measure $\boldsymbol{x}(0)$ directly. In this case, $P(0)$ reflects the accuracy of the sensor used for this measurement.
2. We may have some *a priori* knowledge of $\hat{\boldsymbol{x}}(0)$ and $P(0)$; e.g., the initial state may be deducible from physics.
3. We can set $\hat{\boldsymbol{x}}(0)$ arbitrarily and let $P(0) = c \cdot I$, with c very large. Problems with this approach:
   (a) In linear problems, the wrong $\hat{\boldsymbol{x}}(0)$ and $P(0)$ will make initial filter performance suboptimal.
   (b) In nonlinear problems, the filter may diverge and never recover.
4. We can apply non-Bayesian estimation to a block of data to obtain $\hat{\boldsymbol{x}}$ and P at the end of the block.

Let us further examine method 4. Measurements $\boldsymbol{z}(1), \boldsymbol{z}(2), ..., \boldsymbol{z}(k)$ can be written in terms of the state, input, and noises as

$$\begin{aligned}
\boldsymbol{z}(1) &= H(1)\boldsymbol{x}(1) + \boldsymbol{w}(1) = H(1)[F(0)\boldsymbol{x}(0) + G(0)\boldsymbol{u}(0) + \Gamma(0)\boldsymbol{v}(0)] + \boldsymbol{w}(1) \\
\boldsymbol{z}(2) &= H(2)\{F(1)[F(0)\boldsymbol{x}(0) + G(0)\boldsymbol{u}(0) + \Gamma(0)\boldsymbol{v}(0)] + G(1)\boldsymbol{u}(1) + \Gamma(1)\boldsymbol{v}(1)\} + \boldsymbol{w}(2) \\
\vdots \quad &= \qquad\qquad\qquad\qquad \vdots
\end{aligned}$$

When stacked these take the form

$$Z^k = H^k \boldsymbol{x}(0) + G^{k-1} U^{k-1} + \Gamma^{k-1} V^{k-1} + W^k$$

where

$$Z^k = \underbrace{\begin{bmatrix} \boldsymbol{z}(1) \\ \boldsymbol{z}(2) \\ \vdots \\ \boldsymbol{z}(k) \end{bmatrix}}_{(n_z \cdot k) \times 1}, \quad H^k = \underbrace{\begin{bmatrix} H(1)F(0) \\ H(2)F(1)F(0) \\ \vdots \\ H(k)F(k-1)F(k-2)\cdots F(0) \end{bmatrix}}_{(n_z \cdot k) \times n_x}, \quad U^{k-1} = \underbrace{\begin{bmatrix} \boldsymbol{u}(0) \\ \boldsymbol{u}(1) \\ \vdots \\ \boldsymbol{u}(k-1) \end{bmatrix}}_{(n_u \cdot k) \times 1}$$

$$G^{k-1} = \underbrace{\begin{bmatrix} H(1)G(0) & 0 & \cdots & 0 \\ H(2)F(1)G(0) & H(2)G(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \underbrace{H(k)F(k-1)\ldots F(1)G(0)}_{\boldsymbol{u}(0) \text{ columns}} & \underbrace{H(k)F(k-1)\ldots F(2)G(1)}_{\boldsymbol{u}(1) \text{ columns}} & \cdots & \underbrace{H(k)G(k-1)}_{\boldsymbol{u}(k-1) \text{ columns}} \end{bmatrix}}_{(n_z \cdot k) \times (n_u \cdot k)}$$

$$V^{k-1} = \underbrace{\begin{bmatrix} \boldsymbol{v}(0) \\ \boldsymbol{v}(1) \\ \vdots \\ \boldsymbol{v}(k-1) \end{bmatrix}}_{(n_v \cdot k)\times 1}, \quad W^k = \underbrace{\begin{bmatrix} \boldsymbol{w}(1) \\ \boldsymbol{w}(2) \\ \vdots \\ \boldsymbol{w}(k) \end{bmatrix}}_{(n_z \cdot k)\times 1}, \quad \Gamma^{k-1} = \underbrace{\begin{bmatrix} \text{same as } G^{k-1} \text{ but with} \\ G(j) \text{ replaced by } \Gamma(j) \end{bmatrix}}_{(n_z \cdot k)\times(n_v \cdot k)}$$

Let

$$R^k = \mathbb{E}\left[W^k \left(W^k\right)^T\right] = \begin{bmatrix} R(1) & & & \\ & R(2) & & \\ & & \ddots & \\ & & & R(k) \end{bmatrix}$$

$$Q^{k-1} = \mathbb{E}\left[V^{k-1} \left(V^{k-1}\right)^T\right] = \begin{bmatrix} Q(0) & & & \\ & Q(1) & & \\ & & \ddots & \\ & & & Q(k-1) \end{bmatrix}$$

with the remaining elements zero.

Our stacked measurement equation, together with R^k and Q^{k-1}, define a linear least-squares batch estimation problem that does not depend on $\hat{\boldsymbol{x}}(0)$ or $P(0)$. Cost can be written as a function of the initial state and the full process noise vector:

$$J\left[\boldsymbol{x}(0), V^{k-1}\right] = \tfrac{1}{2}\left[Z^k - H^k\boldsymbol{x}(0) - G^{k-1}U^{k-1} - \Gamma^{k-1}V^{k-1}\right]^T \left(R^k\right)^{-1}[\cdots] + \tfrac{1}{2}\left[V^{k-1}\right]^T \left(Q^{k-1}\right)^{-1}[\cdots]$$

The cost function properly weights errors in measurement and process noise. We minimize the cost by solving

$$0 = \left[\frac{\partial J}{\partial \boldsymbol{x}(0)}\right]^T, \quad 0 = \left[\frac{\partial J}{\partial V^{k-1}}\right]^T$$

Note that

$$\begin{bmatrix} P_{\boldsymbol{x}\boldsymbol{x}}(0) & P_{\boldsymbol{x}V^{k-1}} \\ P^T_{\boldsymbol{x}V^{k-1}} & P_{V^{k-1}V^{k-1}} \end{bmatrix} = \left[\frac{\partial^2 J}{\partial \begin{pmatrix} \boldsymbol{x}(0) \\ V^{k-1} \end{pmatrix}^2}\right]^{-1}$$

Remarks.

1. We must make k large enough so that the minimum is unique: H^k should have more rows than columns and the n_x columns of H^k should be linearly independent. H^k functions like an observability matrix for the batch estimation problem.
2. The resulting $\hat{\boldsymbol{x}}(0)$ is $E[\boldsymbol{x}(0)|Z^k]$, a smoothed estimate of $\boldsymbol{x}(0)$ using future data.
3. We usually want $E[\boldsymbol{x}(k)|Z^k]$. We can iterate the initial estimate forward using our knowledge of the dynamics and our estimates for $\boldsymbol{v}(j)$ to get $\hat{\boldsymbol{x}}(k|k)$. We can also iterate $P_{\boldsymbol{x}\boldsymbol{x}}(0)$ forward to get $P(k)$. We then start our Kalman filter at time k.
4. For a linear system, this is equivalent to choosing $\hat{\boldsymbol{x}}(0)$ arbitrarily, setting $P(0) = \infty$ and running the Kalman filter from 0 to k.
5. This technique for filter initialization can be applied in general to treat a linear sequential estimation problem as a batch estimation problem instead of as a Kalman filtering problem.

9.8 Modeling Errors

Suppose we implement a Kalman filter according to the model

$$\begin{aligned}\boldsymbol{x}(k+1) &= F_f(k)\boldsymbol{x}(k) + G_f(k)\boldsymbol{u}(k) + \boldsymbol{v}_f(k)\\ \boldsymbol{z}(k) &= H_f(k)\boldsymbol{x}(k) + \boldsymbol{w}_f(k)\end{aligned}$$

$$\begin{aligned}&E[\boldsymbol{v}_f] = 0, && E[\boldsymbol{w}_f] = 0\\ &E[\boldsymbol{v}_f(j)\boldsymbol{v}_f^T(k)] = \delta_{jk}Q_f(k) && E[\boldsymbol{w}_f(j)\boldsymbol{w}_f^T(k)] = \delta_{jk}R_f(k)\end{aligned}$$

leading to a Kalman filter with $S_f(k)$, $W_f(k)$, etc.

But suppose the real system obeys

$$\begin{aligned}\boldsymbol{x}(k+1) &= F(k)\boldsymbol{x}(k) + G(k)\boldsymbol{u}(k) + \boldsymbol{v}(k)\\ \boldsymbol{z}(k) &= H(k)\boldsymbol{x}(k) + \boldsymbol{w}(k)\end{aligned}$$

$$\begin{aligned}&E[\boldsymbol{v}] = 0, && E[\boldsymbol{w}] = 0\\ &E[\boldsymbol{v}(j)\boldsymbol{v}^T(k)] = \delta_{jk}Q(k) && E[\boldsymbol{w}(j)\boldsymbol{w}^T(k)] = \delta_{jk}R(k)\end{aligned}$$

with $F(k) \neq F_f(k)$, $G(f) \neq G_f(k)$, etc. If such modeling errors are large, they can produce unpleasant effects.

Let

$$\begin{aligned}\bar{\boldsymbol{e}}(k) &\triangleq \boldsymbol{x}(k) - \bar{\boldsymbol{x}}(k) \quad (a\ \textit{priori}\ \text{error})\\ \boldsymbol{e}(k) &\triangleq \boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k) \quad (a\ \textit{posteriori}\ \text{error})\end{aligned}$$

We can develop a dynamics model for these errors:

$$\begin{aligned}\bar{\boldsymbol{e}}(k+1) &= F_f(k)\boldsymbol{e}(k) + [F(k) - F_f(k)]\boldsymbol{x}(k) + [G(k) - G_f(k)]\boldsymbol{u}(k) + \boldsymbol{v}(k)\\ \boldsymbol{e}(k+1) &= [I - W_f(k+1)H_f(k+1)]\bar{\boldsymbol{e}}(k+1) + W_f(k+1)[H_f(k+1)\\ &\quad - H(k+1)]\boldsymbol{x}(k+1) + W_f(k+1)\boldsymbol{w}(k+1)\end{aligned}$$

Substituting $\bar{\boldsymbol{e}}(k+1)$ into $\boldsymbol{e}(k+1)$, we get

$$\boldsymbol{e}(k+1) = [I - W_f(k+1)H_f(k+1)]F_f(k)\boldsymbol{e}(k) + T_1 + T_2$$

where T_1 is a function of our usual noise terms $\boldsymbol{v}(k)$ and $\boldsymbol{w}(k)$, and T_2 includes terms multiplying $\boldsymbol{x}(k)$, $\boldsymbol{u}(k)$, and $\boldsymbol{v}(k)$ whose matrix coefficients contain the model mismatch errors $[F - F_f]$, $[G - G_f]$, and $[H - H_f]$. These terms do nasty things to the Kalman filter behavior and complicate our analysis.

Remarks.

1. A full analysis requires computation of $\mathbb{E}\left[\boldsymbol{e}(k+1)\boldsymbol{e}^T(k+1)\right] = P_{\text{true}}(k+1)$, which depends on the joint statistics of $\boldsymbol{x}(k)$, $\boldsymbol{u}(k)$, $\boldsymbol{v}(k)$, and $\boldsymbol{w}(k)$ (see Bar-Shalom 5.6.1).

2. Upshot: if the errors $[F(k) - F_f(k)]$, etc. are not too large, then the modeling errors in the filter will be small and the filter can still perform well: $P_{\text{true}}(k+1)$ will not be much larger than $P(k+1)$. But if the modeling errors are too large then the system will perform poorly.

3. If the modeling errors appear large, we may want to improve the system model via system identification techniques. Alternatively, we may perform a "what-if?" simulation analysis to determine whether the effects of the modeling errors are acceptable.