Estimators + Kalman Filters

Recall, for a system with no noise:
\[ \dot{x} = Ax + Bu \]
\[ y = Cx \]

We can design an **observer**:

\[ \hat{y} = C\hat{x} \]
\[ \hat{x} = A\hat{x} + Bu + L(y - C\hat{x}) \]

estimate of \( x(t) \)

model of PLANT dynamics

feedback term:
Observer gain, \( L \),
times the output error:
\[ e_y = y - \hat{y} \]
\( (e_y \text{ sometimes written as } \hat{y}) \)
We saw that the error in the estimate:

\[ e_x = x - \hat{x} \]

has dynamics described by:

\[ \dot{e}_x = (A - LC) e_x \]

So, selection of the observer gains, \( L \), sets the poles of the observer dynamics, described by \( A_o = A - LC \) ... which we want to be asymptotically stable and notably "faster" than the dynamics of the actually plant: \( \dot{x} = (A - BK)x \) \( \{ \text{closed-loop dynamics} \} \)

We can set the poles of \( (A - LC) \) to arbitrary desired locations IF AND ONLY IF \( (C, A) \) IS OBSERVABLE.

* **Note** \( (A - LC) \) \( \neq (A - LC)^T \) have the same poles... Therefore, we can exploit "duality" to solve:

**Duality**:

\[
\begin{align*}
\text{Controller:} & \\
A - BK
\end{align*}
\]

\[
\begin{align*}
\text{Observer:} & \\
(A - LC)^T = A^T - C^T L^T
\end{align*}
\]

Form puts "free matrix" to solve for (e.g. in MATLAB, to the **RIGHT**! In this form, we can treat the pole placement problems just the same for each.
Kalman filter: Acts like a "low pass filter", with good noise rejection...

Real world systems include:
- modeling inaccuracies
- disturbances (to the plant dynamics)
- noise (in the measurement outputs)

A popular type of observer for reconstructing the state in noisy situations is a Kalman filter. Stochastic dynamic system is given by:

\[ \dot{x} = Ax + Bu + Gw \]  
\[ y = Cx + v \]

Assumptions:

\[ \begin{align*}
W(t) & \text{ and } V(t) \text{ have zero mean (no bias) and are white noise processes.} \\
R_w(t) & = E[W(t)W^T(t)] = Q \delta(t) \\
R_v(t) & = E[V(t)V^T(t)] = R \delta(t)
\end{align*} \]

However:

There are "tricks" to augment the true system if \( W(t) [\text{and/or } V(t)] \) is NOT white...
Say \( w(t) \) is \textbf{NOT} white, then we can define:

\[
\begin{align*}
\dot{x}_w &= A_w x_w + B_w n \quad \text{where } "n" \text{ is white noise} \\
W &= C_w x_w + D_w n
\end{align*}
\]

Now, augment the true state, \( x, w \) with new state, \( x_w \). (This technique uses a noise-shaping filter.)

\[
\begin{bmatrix}
\dot{x} \\
\dot{x}_w
\end{bmatrix} =
\begin{bmatrix}
A & GC_w \\
0 & A_w
\end{bmatrix}
\begin{bmatrix}
x \\
x_w
\end{bmatrix} +
\begin{bmatrix}
B \\
0
\end{bmatrix} u +
\begin{bmatrix}
C \\
B_w
\end{bmatrix} n
\]

\[
y =
\begin{bmatrix}
C & 0
\end{bmatrix}
\begin{bmatrix}
x \\
x_w
\end{bmatrix} + v
\]

\( \textbf{(Back to Kalman filtering...)} \)

We now want to select \underline{\textbf{OBSERVER}} gains. As before, our estimated state is:

\[
\hat{x} = A \hat{x} + B u + L (y - \hat{y})
\]

\[
= (A - LC) \hat{x} + Bu + Ly
\]

\[
\hat{y} = E \xi C x + v \xi = C \hat{x}
\]

\[
e_x = x - \hat{x}
\]

\[
= (A - LC) e_x + G w - L v
\]

\[
e_x = A_0 e_x + G w - L v \quad \text{where } A_0 = A - LC
\]

\[
e_y = y - \hat{y}
\]

\[
= C x - C \hat{x}
\]

\[
= C e_x
\]
Now, the error covariance is:

\[ P(t) = E[e_x e_x^T] \]  
\[ \text{time-varying} \]
\[ \text{measures uncertainty of our estimate} \]

\[ P(t) \text{ small} \rightarrow \text{estimate is good.} \]

\[ \text{The optimal gain, } L, \text{ minimizing the steady-state error covariance, } P. \]

\[ \text{(Note: The optimal gain } L \text{ is a constant matrix of observer gains.)} \]

Error covariance for a particular } L \text{ goes as:

\[ \dot{P} = A_o P + P A_o^T + L R L^T + G Q G^T \]

\[ \text{At steady state, } \dot{P} = 0 \quad \text{(eqn above)} \]

\[ O = A_o P + P A_o^T + L R L^T + G Q G^T \]

It turns out, the optimal } L \text{ is:

\[ L = P C^T R^{-1} \]

\[ \text{plug in } \]

\[ \text{for } L^{..} \]

\[ \text{Plugging in for } "L": \]

\[ O = (A - P C^T R^{-1} C) P + P (A - P C^T R^{-1} C)^T + P C^T R^{-1} C P + G Q G^T \]

\[ \text{Recall, } R \text{ is } 0 \text{ for measurement noise.} \]

\[ \text{So, first solve for } P \text{ (eqn above), then solve for } L: \]
"Predictor" \\& "Corrector"

1. \[ \hat{x}_k^- = A \hat{x}_{k-1} + B u_{k-1} \]
2. \[ P_k^- = A P_{k-1} A^T + Q \]

\[ \text{or} \text{ "} K_k \text{"} \]

3. \[ L_k = P_k^- C^T (C P_k^- C^T + R)^{-1} \]
4. \[ \hat{x}_k^+ = \hat{x}_k^- + L_k (y - C \hat{x}_k^-) \]
5. \[ P_k^+ = P_k^- - L_k C P_k^- \]
   \[ = (I - L_k C) P_k^- \]

Usually, \[ L_k \rightarrow K_k \]
\[ C \rightarrow H \]
\[ y \rightarrow z \]

in this iterative form...