Today (12/6/05)

- Another way to find feedback gain: linear quadratic regulator (LQR)
- Application to observer design: Kalman Filter
- Combining LQR and Kalman Filter: Linear Quadratic Gaussian (LQG) controller
Quadratic Optimization

Choosing desired closed poles could be a tedious trial-and-error / hit-and-miss process. Another way to find the feedback gain is to pose the following optimization problem:

\[
\min_u \int_0^\infty \left( x^T Q x + u^T R u \right) dt
\]

This is called the linear quadratic regulator problem, or LQR.
Quadratic Optimization

Solution to the optimization problem:

\[ u = -Fx \]
\[ F = R^{-1}B^TP \]

where \( P \) solves a matrix quadratic equation called the Riccati Equation.

\[ A^TP + PA + Q - PB^TR^{-1}BP = 0 \]

\( P \) is a symmetric positive definite matrix (i.e., \( x^TPx \) is positive for all nonzero \( x \)).

Optimal cost: \( x_0^TPx_0 \) for the initial state \( x_0 \)
MATLAB Command

\[ F = \text{lqr}(A,B,Q,R); \quad \% \text{feedback gain} \]
\[ [F,P,E] = \text{lqr}(A,B,Q,R); \]
\% \text{P = solution of Riccati Equation}
\% \text{E = closed loop eigenvalues (i.e.,}
\% \quad \text{eig}(A-B*F) ) \]
Single Link Example

Consider the single link pendulum:

\[
\min_u \int_0^\infty \left( q_1 x_1^2(t) + q_2 x_2^2(t) + ru^2(t) \right) dt
\]

Tuning parameters are now the relative weights, \((q_1,q_2,r)\), instead of the closed loop poles.

Large \(q_2\) \(\Rightarrow\) slower position response

Small \(r\) \(\Rightarrow\) faster overall response

\[
Q=\text{diag}([1, .1]); R=.01;
\]

\[
F=\text{lqr}(A,B,Q,R); \ \% \ \text{feedback gain}
\]
Robustness

Property of LQR:

Guaranteed (-60°,60°) phase margin and (-6dB,inf dB) gain margin (allowable gain variation from –1/2 to infinity).

So we get the robustness for free! We can concentrate on choosing Q and R to get the time domain performance.
Kalman Filter

We can use the same approach to choose the observer gain. Consider a state space system corrupted by noise:

\[
\dot{x} = Ax + Bu + w \\
y = Cx + Du + \nu
\]

\(w = \text{state noise}, \ \nu = \text{output noise}\)

Both are assumed to be zero mean, white, Gaussian noise with covariance \(W\) and \(V\).

\(w(t)\) and \(\nu(t)\) both are zero mean Gaussian random variables.

\[E[w(t)w^T(\tau)] = \delta(t-\tau)W, \ E[\nu(t)\nu^T(\tau)] = \delta(t-\tau)V\]
We can use the same approach to choose the observer gain.

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

\[
L = \Sigma C^T V^{-1}
\]

where \( \Sigma \) solves the dual Riccati Equation:

\[
A\Sigma + \Sigma A^T + W - \Sigma C^T V^{-1} C \Sigma = 0
\]
LQG Control

Combining LQR and Kalman Filter: Linear Quadratic Gaussian (LQG) controller

\[ \dot{\hat{x}} = (A - LC - (B - LD)F)\hat{x} + Ly \]

where \( F \) is the optimal LQR gain and \( L \) is the Kalman gain

However, there is no longer guaranteed stability margin as in LQR.
MATLAB Command

\[ F = \text{lqr}(A, B, Q, R); \quad \% \text{optimal lqr gain} \]

\[ [F, P, E] = \text{lqr}(A, B, Q, R); \]

\[ L = \text{lqr}(A', C', W, V)'; \quad \% \text{Kalman gain} \]

\% other related commands

\% kalman, lqg