16.323 Lecture 12

Stochastic Optimal Control

- Kwakernaak and Sivan Chapter 3.6, 5
- Bryson Chapter 14
- Stengel Chapter 5
Stochastic Optimal Control

- **Goal:** design optimal compensators for systems with incomplete and noisy measurements
  - Consider this first simplified step: assume that we have noisy system with perfect measurement of the state.

- **System dynamics:**
  \[
  \dot{x}(t) = A(t)x(t) + B_u(t)u(t) + B_w(t)w(t)
  \]
  - Assume that \(w(t)\) is a white Gaussian noise \(\Rightarrow \mathcal{N}(0, R_{ww})\)
  - The initial conditions are random variables too, with
    \[
    E[x(t_0)] = 0, \text{ and } E[x(t_0)x^T(t_0)] = X_0
    \]
    - Assume that a perfect measure of \(x(t)\) is available for feedback.

- Given the noise in the system, need to modify our cost functions from before \(\Rightarrow\) consider the average response of the closed-loop system
  \[
  J_s = E \left\{ \frac{1}{2}x^T(t_f)P_i_jx(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (x^T(t)R_{xx}(t)x(t) + u^T(t)R_{uu}(t)u(t))dt \right\}
  \]
  - Average over all possible realizations of the disturbances.

- **Key observation:** since \(w(t)\) is white, then by definition, the correlation times-scales are very short compared to the system dynamics
  - Impossible to predict \(w(\tau)\) for \(\tau > t\), even with perfect knowledge of the state for \(\tau \leq t\)
  - Furthermore, by definition, the system state \(x(t)\) encapsulates all past information about the system
  - Then the optimal controller for this case is identical to the deterministic one considered before.

---

2016.322 Notes

June 18, 2008
Spectral Factorization

- Had the process noise \( w(t) \) had “color” (i.e., not white), then we need to include a shaping filter that captures the spectral content (i.e., temporal correlation) of the noise \( \Phi(s) \)
  - Previous picture: system is \( y = G(s)w_1 \), with white noise input
  \[
  \begin{array}{c}
  w_1 \\
  \end{array} \xrightarrow{G(s)} \begin{array}{cc}
  & y \\
  \end{array}
  \]
  - New picture: system is \( y = G(s)w_2 \), with shaped noise input
  \[
  \begin{array}{c}
  w_2 \\
  \end{array} \xrightarrow{G(s)} \begin{array}{cc}
  & y \\
  \end{array}
  \]
  - Account for the spectral content using a shaping filter \( H(s) \), so that the picture now is of a system \( y = G(s)H(s)w_1 \), with a white noise input
    \[
    \begin{array}{ccc}
    w_1 & \xrightarrow{H(s)} & \tilde{w}_2 \\
    \end{array} \xrightarrow{G(s)} \begin{array}{cc}
    & y \\
    \end{array}
    \]
    - Then must design filter \( H(s) \) so that the output is a noise \( \tilde{w}_2 \) that has the frequency content that we need

- How design \( H(s) \)? Spectral Factorization – design a stable minimum phase linear transfer function that replicates the desired spectrum of \( w_2 \).
  - Basis of approach: If \( e_2 = H(s)e_1 \) and \( e_1 \) is white, then the spectrum of \( e_2 \) is given by
    \[
    \Phi_{e_2}(j\omega) = H(j\omega)H(-j\omega)\Phi_{e_1}(j\omega)
    \]
    where \( \Phi_{e_1}(j\omega) = 1 \) because it is white.
Typically $\Phi_{w_2}(j\omega)$ will be given as an expression in $\omega^2$, and we factor that into two parts, one of which is stable minimum phase, so if

$$
\Phi_{w_2}(j\omega) = \frac{2\sigma^2\alpha^2}{\omega^2 + \alpha^2}
= \frac{\sqrt{2}\sigma\alpha}{\alpha + j\omega} \cdot \frac{\sqrt{2}\sigma\alpha}{\alpha - j\omega} = H(j\omega)H(-j\omega)
$$

so clearly $H(s) = \frac{\sqrt{2}\sigma\alpha}{s + \alpha}$ which we write in state space form as

$$
\begin{align*}
\dot{x}_H &= -\alpha x_H + \sqrt{2}\alpha\sigma w_1 \\
w_2 &= x_H
\end{align*}
$$

More generally, the shaping filter will be

$$
\begin{align*}
\dot{x}_H &= A_H x_H + B_H w_1 \\
w_2 &= C_H x_H
\end{align*}
$$

which we then augment to the plant dynamics, to get:

$$
\begin{bmatrix}
\dot{x} \\
\dot{x}_H
\end{bmatrix} =
\begin{bmatrix}
A & B_u C_H \\
0 & A_H
\end{bmatrix}
\begin{bmatrix}
x \\
x_H
\end{bmatrix} +
\begin{bmatrix}
B_u \\
0
\end{bmatrix} u +
\begin{bmatrix}
0 \\
B_H
\end{bmatrix} w_1
$$

$$
\begin{align*}
y &= C_y x_H \\
x &= \begin{bmatrix} x \\ x_H \end{bmatrix}
\end{align*}
$$

where the noise input $w_1$ is a white Gaussian noise.

Clearly this augmented system has the same form as the original system that we analyzed - there are just more states to capture the spectral content of the original shaped noise.
Disturbance Feedforward

Now consider the stochastic LQR problem for this case.

- Modify the state weighting matrix so that

\[
\tilde{R}_{xx} = \begin{bmatrix} R_{xx} & 0 \\ 0 & 0 \end{bmatrix}
\]

\( \Rightarrow \) i.e. no weighting on the filter states – Why is that allowed?

- Then, as before, the stochastic LQR solution for the augmented system is the same as the deterministic LQR solution (6–9)

\[
\mathbf{u} = - \left[ \begin{array}{cc} K & K_d \end{array} \right] \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_H \end{bmatrix}
\]

- So the full state feedback controller requires access to the state in the shaping filter, which is fictitious and needs to be estimated

Interesting result is that the gain \( K \) on the system states is completely independent of the properties of the disturbance

- In fact, if the solution of the steady state Riccati equation in this case is partitioned as

\[
P_{\text{aug}} = \begin{bmatrix} P_{xx} & P_{xxH} \\ P_{xHx} & P_{xHxH} \end{bmatrix}
\]

it is easy to show that

- \( P_{xx} \) can be solved for independently, and
- Is the same as it would be in the deterministic case with the disturbances omitted \(^{21}\)

- Of course the control inputs that are also based on \( \mathbf{x}_H \) will improve the performance of the system \( \Rightarrow \) disturbance feedforward.

\(^{21}\text{K+S pg 262}\)
Recall that the specific initial conditions do not effect the LQR controller, but they do impact the cost-to-go from $t_0$.

- Consider the stochastic LQR problem, but with $w(t) \equiv 0$ so that the only uncertainty is in the initial conditions.

- Have already shown that LQR cost can be written in terms of the solution of the Riccati equation (4–7):

$$J_{LQR} = \frac{1}{2} x^T(t_0) P(t_0) x(t_0)$$

$$\Rightarrow J_s = E \left\{ \frac{1}{2} x^T(t_0) P(t_0) x(t_0) \right\}$$

$$= \frac{1}{2} E \{ \text{trace} [P(t_0) x(t_0) x^T(t_0)] \}$$

$$= \frac{1}{2} \text{trace} [P(t_0) X_0]$$

which gives expected cost-to-go with uncertain IC.

- Now return to case with $w \neq 0$ – consider the average performance of the stochastic LQR controller.

- To do this, recognize that if we apply the LQR control, we have a system where the cost is based on $z^T R_z z = x^T R_x x$ for the closed-loop system:

$$\dot{x}(t) = (A(t) - B_u(t) K(t)) x(t) + B_w(t) w(t)$$

$$z(t) = C_z(t) x(t)$$

- This is of the form of a linear time-varying system driven by white Gaussian noise – called a Gauss-Markov Random process\textsuperscript{22}.

\textsuperscript{22}Bryson 11.4
• For a Gauss-Markov system we can predict the **mean square value** of the state \( X(t) = E[x(t)x(t)^T] \) over time using \( X(0) = X_0 \) and

\[
\dot{X}(t) = [(A(t) - B_u(t)K(t)) X(t) + X(t)[A(t) - B_u(t)K(t)]]^T + B_wR_{ww}B_w^T
\]

- **Matrix differential Lyapunov Equation.**

• Can also extract the mean square control values using

\[
E[u(t)u(t)^T] = K(t)X(t)K(t)^T
\]

• Now write performance evaluation as:

\[
J_s = \frac{1}{2} E \left\{ x^T(t_f)P_{t_f}x(t_f) + \int_{t_0}^{t_f} (x^T(t)R_{xx}(t)x(t) + u^T(t)R_{uu}(t)u(t)) dt \right\}
\]

\[
= \frac{1}{2} E \left\{ \text{trace} \left[ P_{t_f}x(t_f)x^T(t_f) + \int_{t_0}^{t_f} (R_{xx}(t)x(t)x^T(t) + R_{uu}(t)u(t)u^T(t)) dt \right] \right\}
\]

\[
= \frac{1}{2} \text{trace} \left[ P_{t_f}X(t_f) + \int_{t_0}^{t_f} (R_{xx}(t)X(t) + R_{uu}(t)K(t)X(t)K(t)^T) dt \right]
\]

• Not too useful in this form, but if \( P(t) \) is the solution of the LQR Riccati equation, then can show that the cost can be written as:

\[
J_s = \frac{1}{2} \text{trace} \left\{ P(t_0)X(t_0) + \int_{t_0}^{t_f} (P(t)B_wR_{ww}B_w^T) dt \right\}
\]

- First part, \( \frac{1}{2} \text{trace} \{ P(t_0)X(t_0) \} \) is the same cost-to-go from the uncertain initial condition that we identified on 11–5

- Second part shows that the cost increases as a result of the process noise acting on the system.
**Sketch of Proof:** first note that

\[ P(t_0)X(t_0) - P_{t_f}X(t_f) + \int_{t_0}^{t_f} \frac{d}{dt}(P(t)X(t))dt = 0 \]

\[
J_s = \frac{1}{2} \text{trace} \left[ P_{t_f}X(t_f) + P(t_0)X(t_0) - P_{t_f}X(t_f) \right] \\
+ \frac{1}{2} \text{trace} \left[ \int_{t_0}^{t_f} \{R_{xx}(t)X(t) + R_{uu}(t)K(t)X(t)K(t)^T\}dt \right] \\
+ \frac{1}{2} \text{trace} \left[ \int_{t_0}^{t_f} \{\dot{P}(t)X(t) + P(t)\dot{X}(t)\}dt \right]
\]

and (first reduces to standard CARE if \( K(t) = R_{uu}^{-1}B_u^TP(t) \))

\[
\dot{P}(t)X(t) = (A - B_uK(t))^TP(t)X(t) + P(t)(A - B_uK(t))X(t) \\
+ R_{xx}X(t) + K(t)^TR_{uu}K(t)X(t)
\]

\[
P(t)\dot{X}(t) = P(t)(A - B_uK(t))X(t) + P(t)X(t)(A - B_uK(t))^T \\
+ P(t)B_wR_{ww}B_w^T
\]

• Rearrange terms within the trace and then cancel terms to get final result.
Problems exist if we set $t_0 = 0$ and $t_f \to \infty$ because performance will be infinite

- Modify the cost to consider the time-average

$$J_a = \lim_{t_f \to \infty} \frac{1}{t_f - t_0} J_s$$

- No impact on necessary conditions since this is still a fixed end-time problem.
- But now the initial conditions become irrelevant, and we only need focus on the integral part of the cost.

For LTI system with stationary process noise (constant $R_{ww}$) and well-posed time-invariant control problem (steady gain $u(t) = -K_{ss}x(t)$) mean square value of state settles down to a constant

$$\lim_{t_f \to \infty} X(t) = X_{ss}$$

$$0 = (A - B_uK_{ss})X_{ss} + X_{ss}(A - B_uK_{ss})^T + B_wR_{ww}B_w^T$$

- Can show that time-averaged mean square performance is

$$J_a = \frac{1}{2} \text{trace} \left( [R_{xx} + K_{ss}^T R_{uu} K_{ss}]X_{ss} \right)$$

$$\equiv \frac{1}{2} \text{trace} \left[ P_{ss} B_w R_{ww} B_w^T \right]$$

- Main point: this gives a direct path to computing the expected performance of a closed-loop system
- Process noise enters into computation of $X_{ss}$
• Consider a missile roll attitude control system with $\omega$ the roll angular velocity, $\delta$ the aileron deflection, $Q$ the aileron effectiveness, and $\phi$ the roll angle, then

$$\begin{align*}
\dot{\delta} &= u \\
\dot{\omega} &= -\frac{1}{\tau} \omega + \frac{Q}{\tau} \delta + n(t) \\
\dot{\phi} &= \omega
\end{align*}$$

where $n(t)$ is a noise input.

• Then this can be written as:

$$\begin{bmatrix}
\dot{\delta} \\
\dot{\omega} \\
\dot{\phi}
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 \\
-1/\tau & Q/\tau & 0 \\
0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
\delta \\
\omega \\
\phi
\end{bmatrix} +
\begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix} u +
\begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix} n$$

• Use $\tau = 1$, $Q = 10$, $R_{uu} = 1/(\pi)^2$ and

$$R_{xx} = \begin{bmatrix}
(\pi/12)^2 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & (\pi/180)^2
\end{bmatrix}$$

then solve LQR problem to get feedback gains:

$$K = lqr(A, B, R_{xx}, R_{uu})$$

$$K = [26.9 \ 29.0 \ 180.0]$$

• Then if $n(t)$ has a spectral density of 1000 $(\text{deg/sec}^2)^2$. sec $^{23}$

• Find RMS response of the system from

$$X = \text{lyap}(A - B*K, Bw*R_{ww}*Bw')$$

$$X = \begin{bmatrix}
95 & -42 & -7 \\
-42 & 73 & 0 \\
-7 & 0 & 0.87
\end{bmatrix}$$

and that $\sqrt{E[\phi^2]} \approx 0.93 \text{deg}$

---

$^{23}$Process noise input to a derivative of $\omega$, so the units of $n(t)$ must be $\text{deg/sec}^2$, but since $E[n(t)n(\tau)] = R_{ww} \delta(t - \tau)$ and $\int \delta(t)dt = 1$, then the units of $\delta(t)$ are $1/\text{sec}$ and thus the units of $R_{ww}$ are $(\text{rad/sec}^2)^2$. sec$=\text{rad}^2/\text{sec}^3$
Goal: design an optimal controller for a system with incomplete and noisy measurements

Setup: for the system (possibly time-varying)

\[
\begin{align*}
adots x &= Ax + Bu + Bw \quad w \\
z &= Cz x \\
y &= Cy x + v 
\end{align*}
\]

with

- White, Gaussian noises \( w \sim \mathcal{N}(0, R_{ww}) \) and \( v \sim \mathcal{N}(0, R_{vv}) \), with \( R_{ww} > 0 \) and \( R_{vv} > 0 \)
- Initial conditions \( x(t_0) \), a stochastic vector with \( E[x(t_0)] = \bar{x}_0 \) and \( E[(x(t_0) - \bar{x}_0)(x(t_0) - \bar{x}_0)^T] = Q_0 \) so that

\( x(t_0) \sim \mathcal{N}(\bar{x}_0, Q_0) \)

Cost:

\[
J = E \left\{ \frac{1}{2} x^T(t_f)P_{tf}x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (z^T(t)R_{zz}z(t) + u^T(t)R_{uu}u(t))dt \right\}
\]

with \( R_{zz} > 0, R_{uu} > 0, P_{tf} \geq 0 \)

Stochastic Optimal Output Feedback Problem: Find

\[
u(t) = f[y(\tau), t_0 \leq \tau \leq t] \quad t_0 \leq t \leq t_f
\]

that minimizes \( J \)

The solution is the Linear Quadratic Gaussian Controller, which uses

- LQE (10–15) to get optimal state estimates \( \hat{x}(t) \) from \( y(t) \) using gain \( L(t) \)
- LQR to get the optimal feedback control \( u(t) = -K(t)x \)
- Separation principle to implement \( u(t) = -K(t)\hat{x}(t) \)
Solution: LQG

- Regulator: \( u(t) = -K(t)\dot{x}(t) \)

\[
K(t) = R_{uu}^{-1}B_u^TP(t)
\]

\[
-\dot{P}(t) = A^TP(t) + P(t)A + C_z^TR_{zz}C_z - P(t)B_uR_{uu}^{-1}B_u^TP(t)
\]

\[
P(t_f) = P_{lf}
\]

- Estimator from:

\[
\dot{\hat{x}}(t) = Ax + B_uu + L(t)(y(t) - C_y\hat{x}(t))
\]

where \( \hat{x}(t_0) = \bar{x}_0 \) and \( Q(t_0) = Q_0 \)

\[
\dot{Q}(t) = AQ(t) + Q(t)A^T + B_wR_{ww}B_w^T - Q(t)C_y^TR_{vv}^{-1}C_yQ(t)
\]

\[
L(t) = Q(t)C_y^TR_{vv}^{-1}
\]

- A compact form of the compensator is:

\[
\begin{align*}
\dot{x}_c &= A_cx_c + B_cy \\
u &= -C_cx_c
\end{align*}
\]

with \( x_c \equiv \dot{x} \) and

\[
\begin{align*}
A_c &= A - B_uK(t) - L(t)C_y \\
B_c &= L(t) \\
C_c &= K(t)
\end{align*}
\]

- Valid for SISO and MIMO systems. Plant dynamics can also be time-varying, but suppressed for simplicity.

  – Obviously compensator is constant if we use the steady state regulator and estimator gains for an LTI system.
• Assuming LTI plant
• As with the stochastic LQR case, use time averaged cost
  – To ensure that estimator settles down, must take $t_0 \to -\infty$ and $t_f \to \infty$, so that for any $t$, $t_0 \ll t \ll t_f$
    \[
    \bar{J} = \lim_{t_f \to \infty, t_0 \to -\infty} \frac{1}{t_f - t_0} J
    \]
  – Again, this changes the cost, but not the optimality conditions

• Analysis of $\bar{J}$ shows that it can be evaluated as
    \[
    \bar{J} = E[z^T(t)R_{zz}z(t) + u^T(t)R_{uu}u(t)] = \text{Tr}[P_{ss}L_{ss}R_{vv}L_{ss}^T + Q_{ss}C_z^T R_{zz}C_z] = \text{Tr}[P_{ss}B_wR_{ww}B_w^T + Q_{ss}K_{ss}^T R_{uu}K_{ss}]
    \]
where $P_{ss}$ and $Q_{ss}$ are the steady state solutions of
    \[
    A^T P_{ss} + P_{ss}A + C_z^T R_{zz}C_z - P_{ss}B_uR_{uu}^{-1}B_u^T P_{ss} = 0
    \]
    \[
    A Q_{ss} + Q_{ss}A^T + B_wR_{ww}B_w^T - Q_{ss}C_y^T R_{vv}^{-1} C_y Q_{ss} = 0
    \]
with
    \[
    K_{ss} = R_{uu}^{-1} B_u^T P_{ss} \quad \text{and} \quad L_{ss} = Q_{ss} C_y^T R_{vv}^{-1}
    \]
• Can evaluate the steady state performance from the solution of 2 Riccati equations
  – More complicated than stochastic LQR because $\bar{J}$ must account for performance degradation associated with estimation error.
  – Since in general $\hat{x}(t) \neq x(t)$, have two contributions to the cost
    ◦ Regulation error $x \neq 0$
    ◦ Estimation error $\hat{x} \neq 0$
Note that
\[
\bar{J} = \text{Tr}[P_{ss} L_{ss} R_{vv} L_{ss}^T + Q_{ss} C_z^T R_{zz} C_z] \\
= \text{Tr}[P_{ss} B_w R_{ww} B_w^T + Q_{ss} K_{ss}^T R_{uu} K_{ss}]
\]
both of which contain terms that are functions of the control and estimation problems.

To see how both terms contribute, let the regulator get very fast \( R_{uu} \rightarrow 0 \). A full analysis requires that we then determine what happens to \( P_{ss} \) and thus \( \bar{J} \). But what is clear is that:
\[
\lim_{R_{uu} \rightarrow 0} \bar{J} \geq \text{Tr}[Q_{ss} C_z^T R_{zz} C_z]
\]
which is independent of \( R_{uu} \)

Thus even in the limit of no control penalty, the performance is lower bounded by term associated with estimation error \( Q_{ss} \).

Similarly, can see that \( \lim_{R_{vv} \rightarrow 0} \bar{J} \geq \text{Tr}[P_{ss} B_w R_{ww} B_w^T] \) which is related to the regulation error and provides a lower bound on the performance with a fast estimator

Note that this is the average cost for the stochastic LQR problem.

Both cases illustrate that it is futile to make either the estimator or regulator much “faster” than the other

The ultimate performance is limited, and you quickly reach the “knee in the curve” for which further increases in the authority of one over the other provide diminishing returns.

Also suggests that it is not obvious that either one of them should be faster than the other.

**Rule of Thumb:** for given \( R_{zz} \) and \( R_{ww} \), select \( R_{uu} \) and \( R_{vv} \) so that the performance contributions due to the estimation and regulation error are comparable.
Separation Theorem

• Now consider what happens when the control \( u = -Kx \) is changed to the new control \( u = -K\dot{x} \) (same \( K \)).
  – Assume steady state values here, but not needed.
  – Previous looks at this would have analyzed the closed-loop stability, as follows, but we also want to analyze performance.

    \[
    \text{plant:} \quad \begin{align*}
    \dot{x} &= Ax + Bu + Bw \\
    z &= Cz x \\
    y &= Cy x + v
    \end{align*}
    \]

    \[
    \text{compensator:} \quad \begin{align*}
    \dot{x}_c &= A_c x_c + B_c y \\
    u &= -C_c x_c
    \end{align*}
    \]

• Which give the closed-loop dynamics

    \[
    \begin{bmatrix}
    \dot{x} \\
    \dot{x}_c
    \end{bmatrix} = \begin{bmatrix}
    A & -Bu C_c \\
    B_c C_y & A_c
    \end{bmatrix} \begin{bmatrix}
    x \\
    x_c
    \end{bmatrix} + \begin{bmatrix}
    B_w \\
    0
    \end{bmatrix} \begin{bmatrix}
    w \\
    v
    \end{bmatrix}
    \]

    \[
    z = \begin{bmatrix}
    C_z & 0
    \end{bmatrix} \begin{bmatrix}
    x \\
    x_c
    \end{bmatrix}
    \]

    \[
    y = \begin{bmatrix}
    C_y & 0
    \end{bmatrix} \begin{bmatrix}
    x \\
    x_c
    \end{bmatrix} + v
    \]

• It is not obvious that this system will even be stable: \( \lambda_i(A_{cl}) < 0? \)
  – To analyze, introduce \( n = x - x_c \), and the similarity transform

    \[
    T = \begin{bmatrix}
    I & 0 \\
    I & -I
    \end{bmatrix} = T^{-1} \quad \Rightarrow \quad \begin{bmatrix}
    x \\
    n
    \end{bmatrix} = T \begin{bmatrix}
    x \\
    x_c
    \end{bmatrix}
    \]

    so that \( A_{cl} \Rightarrow TA_{cl}T^{-1} = \overline{A_{cl}} \) and when you work through the math, you get

    \[
    \overline{A_{cl}} = \begin{bmatrix}
    A - Bu K & B_u K \\
    0 & A - LC_y
    \end{bmatrix}
    \]
• Absolutely key points:
  1. $\lambda_i(A_{cl}) \equiv \lambda_i(A_{cl})$
  2. $A_{cl}$ is block upper triangular, so can find poles by inspection:

$$\det(sI - A_{cl}) = \det(sI - (A - B_uK)) \cdot \det(sI - (A - LC_y))$$

The closed-loop poles of the system consist of the union of the regulator and estimator poles

- This shows that we can design any estimator and regulator separately with confidence that the combination will stabilize the system.
  ◊ Also means that the LQR/LQE problems decouple in terms of being able to predict the stability of the overall closed-loop system.

• Let $G_c(s)$ be the compensator transfer function (matrix) where

$$u = -G_c(sI - A_c)^{-1}B_c y = -G_c(s)y$$

- Reason for this is that when implementing the controller, we often do not just feedback $-y(t)$, but instead have to include a reference command $r(t)$

- Use servo approach and feedback $e(t) = r(t) - y(t)$ instead

- So now $u = G_c e = G_c (r - y)$, and if $r = 0$, then have $u = G_c (-y)$

• Important points:
  - Closed-loop system will be stable, but the compensator dynamics need not be.
  - Often very simple and useful to provide classical interpretations of the compensator dynamics $G_c(s)$. 
• Performance optimality of this strategy is a little harder to establish
  – Now saying more than just that the separation principle is a “good”
    idea \( \Rightarrow \) are trying to say that it is the “best” possible solution

• **Approach:**
  – Rewrite cost and system in terms of the estimator states and dy-
    namics \( \Rightarrow \) recall we have access to these
  – Design a stochastic LQR for this revised system \( \Rightarrow \) full state feed-
    back on \( \hat{x}(t) \)

• Start with the cost (use a similar process for the terminal cost)

\[
E[z^T R_{zz} z] = E[x^T R_{xx} x] \quad \{ \pm \hat{x} \}
\]

\[
= E[(x - \hat{x} + \hat{x})^T R_{xx} (x - \hat{x} + \hat{x})] \quad \{ \tilde{x} = x - \hat{x} \}
\]

\[
= E[\tilde{x}^T R_{xx} \tilde{x}] + 2E[\hat{x}^T R_{xx} \tilde{x}] + E[\hat{x}^T R_{xx} \hat{x}]
\]

• Note that \( \hat{x}(t) \) is the minimum mean square estimate of \( x(t) \) given
  \( y(\tau), u(\tau), t_0 \leq \tau \leq t \).
  – Key property of that estimate is that \( \hat{x} \) and \( \tilde{x} \) are uncorrelated\(^{24}\)

\[
E[\tilde{x}^T R_{xx} \hat{x}] = \text{trace}[E\{\tilde{x} \hat{x}^T\} R_{xx}] = 0
\]

• Also,

\[
E[\tilde{x}^T R_{xx} \tilde{x}] = E[\text{trace}(R_{xx} \hat{x} \tilde{x}^T)] = \text{trace}(R_{xx} Q)
\]

where \( Q \) is the solution of the LQE Riccati equation (11–11)

• So, in summary we have:

\[
E[x^T R_{xx} x] = \text{trace}(R_{xx} Q) + E[\hat{x}^T R_{xx} \hat{x}]
\]

\(^{24}\text{Gelb, pg 112}\)
• Now the main part of the cost function can be rewritten as

\[
J = E \left\{ \frac{1}{2} \int_{t_0}^{t_f} (z^T(t)R_{zz}z(t) + u^T(t)R_{uu}u(t))dt \right\}
\]

\[
= E \left\{ \frac{1}{2} \int_{t_0}^{t_f} (\dot{x}^T(t)R_{xx}\dot{x}(t) + u^T(t)R_{uu}u(t))dt \right\}
\]

\[
+ \frac{1}{2} \int_{t_0}^{t_f} \text{trace}(R_{xx}Q)dt
\]

– The last term is independent of the control \( u(t) \) \( \Rightarrow \) it is only a function of the estimation error

– Objective now is to choose the control \( u(t) \) to minimize the first term

• But first we need another key fact\textsuperscript{25}: If the optimal estimator is

\[
\dot{x}(t) = A\dot{x}(t) + B_u u(t) + L(t)(y(t) - C_y \dot{x}(t))
\]

then by definition, the innovations process

\[
i(t) \equiv y(t) - C_y \dot{x}(t)
\]

is a white Gaussian process, so that \( i(t) \sim \mathcal{N}(0, R_{vv} + C_y QC_y^T) \)

• Then we can rewrite the estimator as

\[
\dot{x}(t) = A\dot{x}(t) + B_u u(t) + L(t)i(t)
\]

which is an LTI system with \( i(t) \) acting as the process noise through a computable \( L(t) \).

\textsuperscript{25}Gelb, pg 317

June 18, 2008
• So combining the above, we must pick $u(t)$ to minimize

$$J = E \left\{ \frac{1}{2} \int_{t_0}^{t_f} (\dot{x}(t)R_{xx}\dot{x}(t) + u^T(t)R_{uu}u(t))dt \right\} + \text{term ind. of } u(t)$$

subject to the dynamics

$$\dot{x}(t) = Ax(t) + Bu(t) + L(t)i(t)$$

– Which is a strange looking Stochastic LQR problem
– As we saw before, the solution is independent of the driving process noise

$$u(t) = -K(t)\dot{x}(t)$$

– Where $K(t)$ is found from the LQR with the data $A$, $B_u$, $R_{xx}$, and $R_{uu}$, and thus will be identical to the original problem.

• Combination of LQE/LQR gives performance optimal result.
\[
\begin{align*}
\dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \\
z &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \\
y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x + v
\end{align*}
\]
where in the LQG problem we have

\[
R_{zz} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad R_{uu} = 1 \quad R_{vv} = 1 \quad R_{ww} = 1
\]

- Solve the SS LQG problem to find that

\[
\begin{align*}
\text{Tr}[P_{ss} L_{ss} R_{vv} L_{ss}^T] &= 8.0 \\
\text{Tr}[Q_{ss} C_z^T R_{zz} C_z] &= 2.8 \\
\text{Tr}[P_{ss} B_w R_{ww} B_{w}^T] &= 1.7 \\
\text{Tr}[Q_{ss} K_{ss}^T R_{uu} K_{ss}] &= 9.1
\end{align*}
\]

- Suggests to me that we need to improve the estimation error \( \Rightarrow \) that 

\( R_{vv} \) is too large. Repeat with

\[
R_{zz} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad R_{uu} = 1 \quad R_{vv} = 0.1 \quad R_{ww} = 1
\]

\[
\begin{align*}
\text{Tr}[P_{ss} L_{ss} R_{vv} L_{ss}^T] &= 4.1 \\
\text{Tr}[Q_{ss} C_z^T R_{zz} C_z] &= 1.0 \\
\text{Tr}[P_{ss} B_w R_{ww} B_{w}^T] &= 1.7 \\
\text{Tr}[Q_{ss} K_{ss}^T R_{uu} K_{ss}] &= 3.7
\end{align*}
\]

and

\[
R_{zz} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad R_{uu} = 1 \quad R_{vv} = 0.01 \quad R_{ww} = 1
\]

\[
\begin{align*}
\text{Tr}[P_{ss} L_{ss} R_{vv} L_{ss}^T] &= 3.0 \\
\text{Tr}[Q_{ss} C_z^T R_{zz} C_z] &= 0.5 \\
\text{Tr}[P_{ss} B_w R_{ww} B_{w}^T] &= 1.7 \\
\text{Tr}[Q_{ss} K_{ss}^T R_{uu} K_{ss}] &= 1.7
\end{align*}
\]
LQG analysis code

\[ A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; \%
\]

\[ B_u = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \%
\]

\[ B_w = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \%
\]

\[ C_y = \begin{bmatrix} 1 & 0 \end{bmatrix}; \%
\]

\[ C_z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \%
\]

\[ R_{ww} = 1; \%
\]

\[ R_{vv} = 1; \%
\]

\[ R_{zz} = \text{diag}([1 \ 1]); \%
\]

\[ R_{uu} = 1; \%
\]

\[ [K, P] = \text{lqr}(A, B_u, C_z \cdot R_{zz} \cdot C_z', R_{uu}); \%
\]

\[ [L, Q] = \text{lqr}(A', C_y', B_w \cdot R_{ww} \cdot B_w', R_{vv}); L = L'; \%
\]

\[ N_1 = \text{trace}(P \cdot (L \cdot R_{vv} \cdot L')); \%
\]

\[ N_2 = \text{trace}(Q \cdot (C_z' \cdot R_{zz} \cdot C_z)); \%
\]

\[ N_3 = \text{trace}(P \cdot (B_w \cdot R_{ww} \cdot B_w')); \%
\]

\[ N_4 = \text{trace}(Q \cdot (K' \cdot R_{uu} \cdot K)); \%
\]

\[ [N_1 \ N_2; N_3 \ N_4] \]
Consider the linearized longitudinal dynamics of a hypothetical helicopter. The model of the helicopter requires four state variables:
- $\theta(t)$: fuselage pitch angle (radians)
- $q(t)$: pitch rate (radians/second)
- $u(t)$: horizontal velocity of CG (meters/second)
- $x(t)$: horizontal distance of CG from desired hover (meters)

The control variable is:
- $\delta(t)$: tilt angle of rotor thrust vector (radians)

![Helicopter in Hover](https://opencourseware.mit.edu)

Figure 12.1: Helicopter in Hover

The linearized equation of motion are:

\[
\begin{align*}
\dot{\theta}(t) &= q(t) \\
\dot{q}(t) &= -0.415q(t) - 0.011u(t) + 6.27\delta(t) - 0.011w(t) \\
\dot{u}(t) &= 9.8\theta(t) - 1.43q(t) - 0.198u(t) + 9.8\delta(t) - 0.0198w(t) \\
\dot{x}(t) &= u(t)
\end{align*}
\]

- $w(t)$ represents a horizontal wind disturbance
- Model $w(t)$ as the output of a first order system driven by zero mean, continuous time, unit intensity Gaussian white noise $\xi(t)$:

\[
\dot{w}(t) = -0.2w(t) + 6\xi(t)
\]
• First, treat original (non-augmented) plant dynamics.
  – Design LQR controller so that an initial hover position error, \( x(0) = 1 \) m is reduced to zero (to within 5\%) in approximately 4 sec.

![Initial response of the closed loop system with \( x(0) = 1 \)](image)

Figure 12.2: Results show that \( R_{uu} = 5 \) gives reasonable performance.

• Augment the noise model, and using the same control gains, form the closed-loop system which includes the wind disturbance \( w(t) \) as part of the state vector.

• Solve necessary Lyapunov equations to determine the (steady-state) variance of the position hover error, \( x(t) \) and rotor angle \( \delta(t) \).
  – Without feedforward:
    \[
    \sqrt{E[x^2]} = 0.048 \quad \sqrt{E[\delta^2]} = 0.017
    \]

• Then design a LQR for the augmented system and repeat the process.
  – With feedforward:
    \[
    \sqrt{E[x^2]} = 0.0019 \quad \sqrt{E[\delta^2]} = 0.0168
    \]
Now do stochastic simulation of closed-loop system using $\Delta t = 0.1$.

- Note the subtly here that the design was for a continuous system, but the simulation will be discrete.
- Are assuming that the integration step is constant.
- Need to create $\zeta$ using the `randn` function, which gives zero mean unit variance Gaussian noise.
- To scale it correctly for a discrete simulation, multiply the output of `randn` by $1/\sqrt{\Delta t}$, where $\Delta t$ is the integration step size.\(^{26}\)
- Could also just convert the entire system to its discrete time equivalent, and then use a process noise that has a covariance

\[
Q_d = \frac{R_{ww}}{\Delta t}
\]

\(^{26}\text{Franklin and Powell, Digital Control of Dynamic Systems}\)
Figure 12.3: Stochastic Simulations with and without disturbance feedforward.
clear all, clf; randn('seed', sum(100*clock));

% linearized dynamics of the system
A = [ 0 0 0 0; 0 -0.415 -0.011 0; 9.8 -1.43 -0.0198 0; 0 0 0 1];
Bu = [0 -0.011 -0.0198 0];
Bu = [0 0.27 9.8 0];
Cz = [0 0 0 1];
Rxx = Cz'*Cz;

rho = 5;
Rww = 1;

% lqr control
[K,S,E] = lqr(A,Bu,Rxx,rho);

[K2,S,E] = lqr(A,Bu,Rxx,10*rho);

[K3,S,E] = lqr(A,Bu,Rxx,rho/10);

% initial response with given x0
x0 = [0 0 0 1];
Ts = 0.1;

% small discrete step to simulate the cts dynamics
tf = 8; t = 0:Ts:tf;

[y, x] = lsim(A-Bu*K, zeros(4, 1), Cz, 0, x0, t);
[y2,x2] = lsim(A-Bu*K2, zeros(4, 1), Cz, 0, x0, t);
[y3,x3] = lsim(A-Bu*K3, zeros(4, 1), Cz, 0, x0, t);

subplot(211), plot(t, [y y2 y3], [0 8], .05*[1 1], ':', [0 8], .05*[-1 -1], ':', 'LineWidth', 2)
ylabel('x'); title('Initial response of the closed loop system with x(0) = 1');

h = legend(['LQR: \rho = ', num2str(rho) ], ['LQR: \rho = ', num2str(rho*10) ], ['LQR: \rho = ', num2str(rho/10) ];
axes(h)

subplot(212), plot(t, [(K*x')' (K2*x2')' (K3*x3')' ], 'LineWidth', 2); grid on
xlabel('Time'), ylabel('\delta')
print -r300 -dpng heli1.png
figure(2);
subplot(211)
plot(t,y,'LineWidth',2)
hold on;
plot(t,dy,'r-.','LineWidth',1.5)
plot([0 max(t)],sqrt(vx)*[1 1],'m--',[0 max(t)],-sqrt(vx)*[1 1],'m--','LineWidth',1.5);
hold off
xlabel('Time');ylabel('y(t)');legend('cts','disc')
title('Stochastic Simulation of Helicopter Response: No FF')
subplot(212)
plot(t,u,'LineWidth',2)
hold on;
plot([0 max(t)],sqrt(vd)*[1 1],'m--',[0 max(t)],-sqrt(vd)*[1 1],'m--','LineWidth',1.5);
hold off
figure(3);
subplot(211)
plot(t,ya,'LineWidth',2)
hold on;
plot(t,dya,'r-.','LineWidth',1.5)
plot([0 max(t)],sqrt(vxa)*[1 1],'m--',[0 max(t)],-sqrt(vxa)*[1 1],'m--','LineWidth',1.5);
hold off
xlabel('Time');ylabel('y(t)');legend('cts','disc')
title('Stochastic Simulation of Helicopter Response: with FF')
subplot(212)
plot(t,ua,'LineWidth',2)
hold on;
plot([0 max(t)],sqrt(vda)*[1 1],'m--',[0 max(t)],-sqrt(vda)*[1 1],'m--','LineWidth',1.5);
hold off
print -f2 -r300 -dpng heli2.png
print -f3 -r300 -dpng heli3.png
Now consider what happens if we reduce the measurable states and use LQG for the helicopter control/simulation.

Consider full vehicle state measurement (i.e., not the disturbance state)

\[ C_y = \begin{bmatrix} I_4 & 0 \end{bmatrix} \]

Consider only partial vehicle state measurement

\[ C_y = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \]

Set \( R_{\text{vv}} \) small.
Figure 12.4: LQR with disturbance feedforward compared to LQG
Figure 12.5: Second LQR with disturbance feedforward compared to LQG
Spr 2008

Helicopter LQG

1  % 16.323 Spring 2008
2  % Stochastic Simulation of Helicopter LQR - from Bryson’s Book
3  % Jon How
4  %
5 clear all, clf, randn('seed',sum(100*clock));
6 set(0,'DefaultAxesFontName','arial')
7 set(0,'DefaultAxesFontSize',12)
8 set(0,'DefaultTextFontName','arial')
% linearized dynamics of the system state=[\theta q \dot{x} x]
9 A = [ 0 1 0 0; 0 -0.615 -0.011 0; 9.8 -1.43 -0.0198 0; 0 0 1 0];
10 Bw = [0 -0.011 -0.0198 0.0 0 0 1];
11 Bu = [0 6.27 9.8 0]';
12 Cz = [0 0 0 1];
13 Rxx = Cz'*Cz;
14 Rww=1;
15 rho = 5;
% lqr control
16 [K,S,E]=lqr(A,Bu,Rxx,rho);
17 % initial response with given x0
18 x0 = [0 0 0 1]';
19 Ts=0.01;
% augment the filter dynamics
20 Aa = [A Bw*Ch; zeros(1,4) Ah];
21 Bua = [Bu;0];
22 Bwa = [zeros(4,1); Bh];
23 Cza = [Cz 0];
24 x0a=[x0;0];
% state
25 vxa = Cza*PP*Cza';
% control
26 vda = KK*PP*KK';
% state
27 [ya,xa] = lsim(Acl,Bwa,Cza,0,zeta,t,x0a);
% control commands given the state response
28 ua = KK*xa';
% now just treat this as a system with more sensor noise acting as more
% process noise
29 for ii=1:Ncy
30 zeta_lqg = [zeta_lqg sqrt(Rvv(ii,ii)/Ts)*randn(nt,1)];
31 end
32 % now consider Output Feedback Case
33 % Assume that we can only measure the system states
34 % and not the dist one
35 FULL=1;
36 if FULL
37 Cya=eye(4,5); % full veh state
38 else
39 Cy=[0 1 0 0;0 0 0 0]; % only meas some states
40 Cya=[Cy [0;0]]; %
41 end
42 Ncy=size(Cya,1);Rvv=(1e-2)^2*eye(Ncy);
43 [LQ,FF]=lqr(Aa',Cya',Bwa*Rww*Bwa',Rvv);L=L'; % LQE calc
44 ACL_lqg=[Aa -Bua*KK;L*Cya Aa-Bua*KK-L*Cya];
45 BCL_lqg=[Bwa zeros(5,Ncy);zeros(5,1) L];
46 CCL_lqg=[Cza zeros(1,5)];DCL_lqg=zeros(1,1+Ncy);
47 x0_lqg=[x0a;zeros(5,1)];
48 zeta_lqg=zeta;
49 zeta_lqg = [zeta_lqg sqrt(Rvv(ii,ii)/Ts)*randn(nt,1)]; % discrete equivalent noise

June 18, 2008
Spr 2008

[ya_lqg,xa_lqg] = lsim(Acl_lqg,Bcl_lqg,Ccl_lqg,Dcl_lqg,zeta_lqg,t,x0_lqg); \% cts sim
F_lqg=c2d(ss(Acl_lqg,Bcl_lqg,Ccl_lqg,Dcl_lqg),Ts); \% discretize the closed-loop dynamics
[dxyla_lqg,txa_lqg] = lsim(F_lqg,zeta_lqg,[],x0_lqg); \% stochastic sim in discrete time
ua_lqg = [zeros(1,5) KK]*xa_lqg'; \% find control commands given the state estimate

% LQG State Perf Prediction
X_lqg=lyap(Acl_lqg,Bcl_lqg*[Rww zeros(1,Ncy);zeros(Ncy,1) Rvv]*Bcl_lqg');
vx_lqg=Ccl_lqg*X_lqg*Ccl_lqg';
vu_lqg=[zeros(1,5) KK]*X_lqg*[zeros(1,5) KK]';

figure(3);clf
subplot(211)
plot(t,ya,'LineWidth',3)
hold on;
plot(t,dya,'r-.','LineWidth',2)
hold off
xlabel('Time');ylabel('y(t)');legend('cts','disc')
title('Stochastic Simulation of Helicopter Response: with FF')
subplot(212)
plot(t,ua,'LineWidth',2)
xlabel('Time');ylabel('u(t)');%legend('with FF')
if FULL
legend('Full veh state')
else
legend('Pitch rate, Horiz Pos')
end
hold on;
plot(t,dvxvla_lqg,'r-.','LineWidth',2)
hold off
xlabel('Time');ylabel('\nu(t)');%legend('with FF')
if FULL
legend('Full veh state')
else
legend('Pitch rate, Horiz Pos')
end
hold on;
plot(t,dvu_lqg,'r-.','LineWidth',2)
hold off
if FULL
print -f4 -r300 -dpng helilqg_1.png;
else
print -f4 -r300 -dpng helilqg_2.png;
end
Example: 747

- Bryson, page 209 Consider the stabilization of a 747 at 40,000 ft and Mach number of 0.80. The perturbation dynamics from elevator angle to pitch angle are given by

\[
\frac{\theta(s)}{\delta_e(s)} = G(s) = \frac{1.16(s + 0.0113)(s + 0.295)}{[s^2 + (0.0676)^2][(s + 0.375)^2 + (0.882)^2]} 
\]

1. Note that these aircraft dynamics can be stabilized with a simple lead compensator

\[
\frac{\delta_e(s)}{\theta(s)} = 3.50 \frac{s + 0.6}{s + 3.6} 
\]

2. Can also design an LQG controller for this system by assuming that \( B_w = B_u \) and \( C_z = C_y \), and then tuning \( R_{uu} \) and \( R_{vv} \) to get a reasonably balanced performance.
- Took \( R_{ww} = 0.1 \) and tuned \( R_{vv} \)

Figure 12.6: B747: Compensators
Figure 12.7: B747: root locus (Lead on left, LQG on right shown as a function of the overall compensator gain)
3. Compare the Bode plots of the lead compensator and LQG designs.

![Bode plots of lead compensator and LQG designs](image)

Figure 12.8: B747: Compensators and loop TF
4. Consider the closed-loop TF for the system

![Graph showing B747: closed-loop TF](image)

**Figure 12.9: B747: closed-loop TF**

5. Compare impulse response of two closed-loop systems.

![Graph showing B747: Impulse response](image)

**Figure 12.10: B747: Impulse response**

6. So while LQG controllers might appear to be glamorous, they are actually quite ordinary for SISO systems.
   - Where they really shine is that it this simple to design a MIMO controller.
% 16.323 B747 example
% Jon How, MIT, Spring 2007
%
clear all

set(0,'DefaultAxesFontName','arial')
set(0,'DefaultAxesFontSize',12)
set(0,'DefaultTextFontName','arial')

% lead comp given
kn=3.5*[1 .6];kd=[1 3.6];

f=logspace(-3,1,300);
g=freqresp(gn,gd,2*pi*f*sqrt(-1));

[nc,dc]=clooep(conv(gn,kn),conv(gd,kd)); % CLP with lead
gc=freqresp(nc,dc,2*pi*f*sqrt(-1)); % CLP with lead

roots(dc)

loglog(f,abs([g gc]))

% get state space model
[a,b,c,d]=tf2ss(gn,gd);

% assume that Bu and Bw are the same
% take y=z
Rzz=1;Ru0=0.01;Rww=0.01;Rvv=0.01;
[k,P,e1]=lqr(a,b,c'*Rzz*c,Ruu);
[l,Q,e2]=lqe(a,b,c,Rww,Rvv);
[ac,bc,cc,tdc]=reg(a,b,c,d,k,l);

[knl,kdl]=ss2tf(ac,bc,cc,tdc);

N1=trace(P*(1*Rvv*1'))
N2=trace(Q*(c'*Rzz*c))
N3=trace(P*(b*Rww*b'))
N4=trace(Q*(k'*Ruu*k'))
N=[N1 N2 N3 N4 N3+N4]

[nc,dc]=clooep(conv(gn,knl),conv(gd,kdl)); % CLP with lqg
gcl=freqresp(nc,dc,2*pi*f*sqrt(-1)); % CLP with lqg

figure(2);clf;
loglog(f,abs([g gc gcl])) % mag plot of closed loop system
setlines(2)
legend('G','Gcl_{lead}','Gcl_{lqg}')
xlabel('Freq (rad/sec)')

Gclead=freqresp(kn,kd,2*pi*f*sqrt(-1));
Gclqg=freqresp(knl,kdl,2*pi*f*sqrt(-1));

figure(3);clf;

subplot(211)
loglog(f,abs([g Gclead Gclqg])) % Bode of compensators
setlines(2)
legend('G','Gc_{lead}','Gc_{lqg}')
xlabel('Freq (rad/sec)')

axis([1e-3 10 1e-2 1e2])

subplot(212)
semilogx(f,unwrap(phase([g])));hold on
semilogx(f,180/pi*unwrap(phase([Gclead])),'g')
semilogx(f,180/pi*unwrap(phase([Gclqg])),'r')
xlabel('Freq (rad/sec)')
hold off

subplot(211)
semilogx(f,abs([g Gclead Gclqg])) % Bode of Loop transfer function

June 18, 2008
setlines(2)
legend('G','Loop_{lead}','Loop_{lqg}')
xlabel('Freq (rad/sec)')
axis([1e-3 10 1e-2 1e2])
subplot(212)
semilogx(f,180/pi*unwrap(phase([g]))) hold on
semilogx(f,180/pi*unwrap(phase([g.*Gclead])), 'g')
semilogx(f,180/pi*unwrap(phase([g.*Gclqg])), 'r')
xlabel('Freq (rad/sec)') hold off
setlines(2)
legend('G','Loop_{lead}','Loop_{lqg}')

RL of 2 closed-loop systems
figure(1); clf; rlocus(conv(gn,kn), conv(gd,kd)); axis(2*[-2.4 0.1 -0.1 2.4])
hold on; plot(roots(dc)+sqrt(-1)*eps,'md','MarkerFaceColor','m'); hold off
figure(5); clf; pzmap(tf(kn,kd), 'g', tf(knl,kdl), 'r')
legend('lead','LQG')

% time simulations
Ts=0.01;
[y1,x,t]=impulse(gn,gd,[0:Ts:10]);
[y2]=impulse(nc,dc,t);
[y3]=impulse(nc1,dcl,t);
[ulead]=lsim(kn,kd,y2,t);
[ulqg]=lsim(knl,kdl,y3,t);
figure(5); clf;
plot(t,[y1 y2 y3])
setlines(2)
legend('G','Gc_{lead}','Gc_{lqg}')
subplot(212)
plot(t,[ulqg])
setlines(2)
legend('Gc_{lead}','Gc_{lqg}')