16.323 Principles of Optimal Control
Spring 2008

16.323 Lecture 12

Stochastic Optimal Control

- Kwaknernaak 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:**

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B_u(t)\mathbf{u}(t) + B_w(t)\mathbf{w}(t)$$

  - Assume that $\mathbf{w}(t)$ is a white Gaussian noise [20] $\Rightarrow \mathcal{N}(0, R_{\text{ww}})$
  - The initial conditions are random variables too, with

$$E[\mathbf{x}(t_0)] = 0, \text{ and } E[\mathbf{x}(t_0)\mathbf{x}^T(t_0)] = X_0$$

  - Assume that a perfect measure of $\mathbf{x}(t)$ 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

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

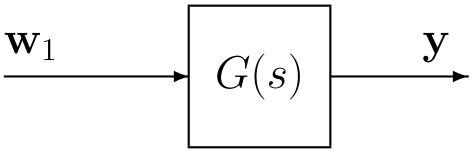
  - Average over all possible realizations of the disturbances.

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

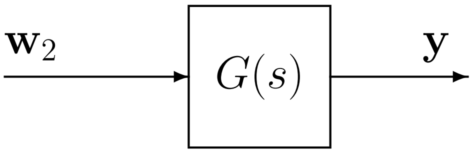
[20] 16.322 Notes

Spectral Factorization

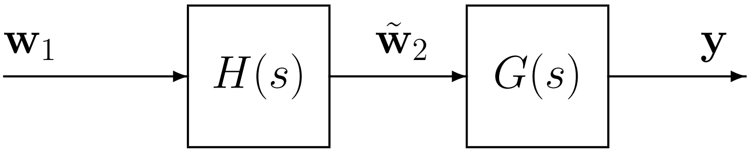
- Had the process noise $\mathbf{w}(t)$ had "color" (i.e., not white), then we need to include a **shaping filter** that captures the spectral content (i.e., temporal correlation) of the noise $\Phi(s)$
  - Previous picture: system is $\mathbf{y} = G(s)\mathbf{w}_1$, with white noise input



  - New picture: system is $\mathbf{y} = G(s)\mathbf{w}_2$, with shaped noise input



- Account for the spectral content using a shaping filter $H(s)$, so that the picture now is of a system $\mathbf{y} = G(s)H(s)\mathbf{w}_1$, with a white noise input



  - Then must design filter $H(s)$ so that the output is a noise $\tilde{\mathbf{w}}_2$ 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 $\mathbf{w}_2$.
  - 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 ω^2, and we factor that into two parts, one of which is stable minimum phase, so if

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

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

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

- More generally, the shaping filter will be

$$\begin{aligned}\dot{\mathbf{x}}_H &= A_H\mathbf{x}_H + B_H\mathbf{w}_1\\ \mathbf{w}_2 &= C_H\mathbf{x}_H\end{aligned}$$

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

$$\begin{aligned}\begin{bmatrix}\dot{\mathbf{x}}\\ \dot{\mathbf{x}}_H\end{bmatrix} &= \begin{bmatrix}A & B_wC_H\\ 0 & A_H\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{x}_H\end{bmatrix} + \begin{bmatrix}B_u\\ 0\end{bmatrix}\mathbf{u} + \begin{bmatrix}0\\ B_H\end{bmatrix}\mathbf{w}_1\\ \mathbf{y} &= \begin{bmatrix}C_y & 0\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{x}_H\end{bmatrix}\end{aligned}$$

where the noise input $\mathbf{w}_1$ 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.

- Now consider the stochastic LQR problem for this case.
  - Modify the state weighting matrix so that

$$\tilde{R}_{\rm xx} = \begin{bmatrix} R_{\rm xx} & 0 \\ 0 & 0 \end{bmatrix}$$

    ⇒ 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)

$$\mathbf{u} = -\begin{bmatrix} K & K_d \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_H \end{bmatrix}$$

  - 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

$$P_{\rm aug} = \left[\begin{array}{c|c} P_{\rm xx} & P_{\rm xx_H} \\ \hline P_{\rm x_Hx} & P_{\rm x_Hx_H} \end{array}\right]$$

    it is easy to show that
    - ◇ $P_{\rm xx}$ 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 ⇒ **disturbance feedforward**.

[21] K+S pg 262

Performance Analysis

- 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 $\mathbf{w}(t) \equiv 0$ 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):

$$\begin{aligned} J_{LQR} &= \frac{1}{2}\mathbf{x}^T(t_0)P(t_0)\mathbf{x}(t_0) \\ \Rightarrow J_s &= E\left\{\frac{1}{2}\mathbf{x}^T(t_0)P(t_0)\mathbf{x}(t_0)\right\} \\ &= \frac{1}{2}E\left\{\texttt{trace}[P(t_0)\mathbf{x}(t_0)\mathbf{x}^T(t_0)]\right\} \\ &= \frac{1}{2}\texttt{trace}[P(t_0)X_0] \end{aligned}$$

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

- Now return to case with $\mathbf{w} \neq 0$ – 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 $\mathbf{z}^T R_{\text{zz}}\mathbf{z} = \mathbf{x}^T R_{\text{xx}}\mathbf{x}$ for the closed-loop system:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= (A(t) - B_u(t)K(t))\mathbf{x}(t) + B_w(t)\mathbf{w}(t) \\ \mathbf{z}(t) &= C_z(t)\mathbf{x}(t) \end{aligned}$$

- This is of the form of a linear time-varying system driven by white Gaussian noise – called a **Gauss-Markov Random process**[22].

[22] Bryson 11.4

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

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

  - Matrix **differential Lyapunov Equation**.

- Can also extract the mean square control values using

$$E[\mathbf{u}(t)\mathbf{u}(t)^T] = K(t)X(t)K(t)^T$$

- Now write performance evaluation as:

$$\begin{aligned}
J_s &= \frac{1}{2}E\left\{\mathbf{x}^T(t_f)P_{t_f}\mathbf{x}(t_f) + \int_{t_0}^{t_f}(\mathbf{x}^T(t)R_{\mathrm{xx}}(t)\mathbf{x}(t) + \mathbf{u}^T(t)R_{\mathrm{uu}}(t)\mathbf{u}(t))dt\right\} \\
&= \frac{1}{2}E\left\{\mathtt{trace}\left[P_{t_f}\mathbf{x}(t_f)\mathbf{x}^T(t_f) + \int_{t_0}^{t_f}(R_{\mathrm{xx}}(t)\mathbf{x}(t)\mathbf{x}^T(t) + R_{\mathrm{uu}}(t)\mathbf{u}(t)\mathbf{u}^T(t))dt\right]\right\} \\
&= \frac{1}{2}\mathtt{trace}\left[P_{t_f}X(t_f) + \int_{t_0}^{t_f}(R_{\mathrm{xx}}(t)X(t) + R_{\mathrm{uu}}(t)K(t)X(t)K(t)^T)dt\right]
\end{aligned}$$

- 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:

$$J_s = \frac{1}{2}\mathtt{trace}\left\{P(t_0)X(t_0) + \int_{t_0}^{t_f}(P(t)B_w R_{\mathrm{ww}} B_w^T)dt\right\}$$

  - First part, $\frac{1}{2}\mathtt{trace}\{P(t_0)X(t_0)\}$ 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

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

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

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

$$\begin{aligned}
-\dot{P}(t)X(t) &= (A - B_uK(t))^T P(t)X(t) + P(t)(A - B_uK(t))X(t) \\
&\quad + R_{\rm xx}X(t) + K(t)^T R_{\rm uu}K(t)X(t)
\end{aligned}$$

$$\begin{aligned}
P(t)\dot{X}(t) &= P(t)(A - B_uK(t))X(t) + P(t)X(t)(A - B_uK(t))^T \\
&\quad + P(t)B_wR_{\rm ww}B_w^T
\end{aligned}$$

- Rearrange terms within the trace and then cancel terms to get final result.

Steady State Values

- 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 $R_{\rm ww}$) and well-posed time-invariant control problem (steady gain $\mathbf{u}(t) = -K_{ss}\mathbf{x}(t)$) mean square value of state settles down to a constant

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

$$0 = (A - B_u K_{ss}) X_{ss} + X_{ss} (A - B_u K_{ss})^T + B_w R_{\rm ww} B_w^T$$

  - Can show that **time-averaged mean square performance** is

$$\begin{aligned} J_a &= \frac{1}{2}\texttt{trace}\left([R_{\rm xx} + K_{ss}^T R_{\rm uu} K_{ss}] X_{ss}\right) \\ &\equiv \frac{1}{2}\texttt{trace}[P_{ss} B_w R_{\rm ww} B_w^T] \end{aligned}$$

- **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}

Missile Example

- Consider a missile roll attitude control system with ω the roll angular velocity, δ the aileron deflection, Q the aileron effectiveness, and ϕ the roll angle, then

$$\dot{\delta} = u \qquad \dot{\omega} = -\frac{1}{\tau}\omega + \frac{Q}{\tau}\delta + n(t) \qquad \dot{\phi} = \omega$$

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

- Then this can be written as:

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

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

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

then solve LQR problem to get feedback gains:

```
K=lqr(A,B,Rxx,Ruu)
```

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

- Then if $n(t)$ has a spectral density of 1000 (deg/sec^2)$^2\cdot$ sec [23]

- Find RMS response of the system from

```
X=lyap(A-B*K,Bw*Rww*Bw')
```

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

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

[23] Process noise input to a derivative of ω, so the units of $n(t)$ must be deg/sec^2, but since $E[n(t)n(\tau)] = R_{\rm ww}\delta(t-\tau)$ and $\int \delta(t)dt = 1$, then the units of $\delta(t)$ are 1/sec and thus the units of $R_{\rm ww}$ are (rad/sec^2)$^2\cdot$ sec=rad^2/sec^3

Full Control Problem

- **Goal:** design an optimal controller for a system with **incomplete and noisy measurements**

- **Setup:** for the system (possibly time-varying)

$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + B_u\mathbf{u} + B_w\mathbf{w} \\ \mathbf{z} &= C_z\mathbf{x} \\ \mathbf{y} &= C_y\mathbf{x} + \mathbf{v}\end{aligned}$$

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

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

- **Cost:**

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

  with $R_{\rm zz} > 0$, $R_{\rm uu} > 0$, $P_{t_f} \geq 0$

- **Stochastic Optimal Output Feedback Problem:** Find

$$\mathbf{u}(t) = \mathbf{f}[\mathbf{y}(\tau), t_0 \leq \tau \leq t] \qquad t_0 \leq t \leq t_f$$

  that minimizes J

- The solution is the Linear Quadratic Gaussian Controller, which uses
  - **LQE** (10–15) to get optimal state estimates $\hat{\mathbf{x}}(t)$ from $\mathbf{y}(t)$ using gain $L(t)$
  - **LQR** to get the optimal feedback control $\mathbf{u}(t) = -K(t)\mathbf{x}$
  - **Separation principle** to implement $\mathbf{u}(t) = -K(t)\hat{\mathbf{x}}(t)$

Solution: LQG

- Regulator: $\mathbf{u}(t) = -K(t)\hat{\mathbf{x}}(t)$

$$
\begin{aligned}
K(t) &= R_{\rm uu}^{-1} B_u^T P(t) \\
-\dot{P}(t) &= A^T P(t) + P(t)A + C_z^T R_{\rm zz} C_z - P(t) B_u R_{\rm uu}^{-1} B_u^T P(t) \\
P(t_f) &= P_{t_f}
\end{aligned}
$$

- Estimator from:

$$
\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}} + B_u \mathbf{u} + L(t)(\mathbf{y}(t) - C_y \hat{\mathbf{x}}(t))
$$

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

$$
\begin{aligned}
\dot{Q}(t) &= AQ(t) + Q(t)A^T + B_w R_{\rm ww} B_w^T - Q(t) C_y^T R_{\rm vv}^{-1} C_y Q(t) \\
L(t) &= Q(t) C_y^T R_{\rm vv}^{-1}
\end{aligned}
$$

- A compact form of the compensator is:

$$
\begin{aligned}
\dot{\mathbf{x}}_c &= A_c \mathbf{x}_c + B_c \mathbf{y} \\
\mathbf{u} &= -C_c \mathbf{x}_c
\end{aligned}
$$

with $\mathbf{x}_c \equiv \hat{\mathbf{x}}$ and

$$
\begin{aligned}
A_c &= A - B_u K(t) - L(t) C_y \\
B_c &= L(t) \\
C_c &= K(t)
\end{aligned}
$$

- 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.

Infinite Horizon LQG

- 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, $t_0 \ll t \ll t_f$

$$\bar{J} = \lim_{\substack{t_f \to \infty \\ t_0 \to -\infty}} \frac{1}{t_f - t_0} J$$

  - Again, this changes the cost, but not the optimality conditions

- Analysis of $\bar{J}$ shows that it can be evaluated as

$$\begin{aligned} \bar{J} &= E[\mathbf{z}^T(t) R_{\mathrm{zz}} \mathbf{z}(t) + \mathbf{u}^T(t) R_{\mathrm{uu}} \mathbf{u}(t)] \\ &= \mathtt{Tr}[P_{ss} L_{ss} R_{\mathrm{vv}} L_{ss}^T + Q_{ss} C_z^T R_{\mathrm{zz}} C_z] \\ &= \mathtt{Tr}[P_{ss} B_w R_{\mathrm{ww}} B_w^T + Q_{ss} K_{ss}^T R_{\mathrm{uu}} K_{ss}] \end{aligned}$$

  where P_{ss} and Q_{ss} are the steady state solutions of

$$\begin{aligned} A^T P_{ss} + P_{ss} A + C_z^T R_{\mathrm{zz}} C_z - P_{ss} B_u R_{\mathrm{uu}}^{-1} B_u^T P_{ss} &= 0 \\ A Q_{ss} + Q_{ss} A^T + B_w R_{\mathrm{ww}} B_w^T - Q_{ss} C_y^T R_{\mathrm{vv}}^{-1} C_y Q_{ss} &= 0 \end{aligned}$$

  with

$$K_{ss} = R_{\mathrm{uu}}^{-1} B_u^T P_{ss} \quad \text{and} \quad L_{ss} = Q_{ss} C_y^T R_{\mathrm{vv}}^{-1}$$

- 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 $\hat{\mathbf{x}}(t) \neq \mathbf{x}(t)$, have two contributions to the cost
    - ◇ Regulation error $\mathbf{x} \neq 0$
    - ◇ Estimation error $\tilde{\mathbf{x}} \neq 0$

Interpretations

- Note that

$$\begin{aligned}\bar{J} &= \texttt{Tr}[P_{ss}L_{ss}R_{\rm vv}L_{ss}^T + Q_{ss}C_z^T R_{\rm zz}C_z] \\ &= \texttt{Tr}[P_{ss}B_w R_{\rm ww}B_w^T + Q_{ss}K_{ss}^T R_{\rm uu}K_{ss}]\end{aligned}$$

  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 $\Rightarrow R_{\rm uu} \to 0$. A full analysis requires that we then determine what happens to P_{ss} and thus $\bar{J}$. But what is clear is that:

$$\lim_{R_{\rm uu}\to 0} \bar{J} \geq \texttt{Tr}[Q_{ss}C_z^T R_{\rm zz}C_z]$$

  which is independent of $R_{\rm uu}$

  – 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 $\lim_{R_{\rm vv}\to 0} \bar{J} \geq \texttt{Tr}[P_{ss}B_w R_{\rm ww}B_w^T]$ 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 $R_{\rm zz}$ and $R_{\rm ww}$, select $R_{\rm uu}$ and $R_{\rm vv}$ so that the performance contributions due to the estimation and regulation error are comparable.

- Now consider what happens when the control $\mathbf{u} = -K\mathbf{x}$ is changed to the new control $\mathbf{u} = -K\hat{\mathbf{x}}$ (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.

$$\begin{aligned} \text{plant :} \qquad \dot{\mathbf{x}} &= A\mathbf{x} + B_u\mathbf{u} + B_w\mathbf{w} \\ \mathbf{z} &= C_z\mathbf{x} \\ \mathbf{y} &= C_y\mathbf{x} + \mathbf{v} \\ \text{compensator :} \qquad \dot{\mathbf{x}}_c &= A_c\mathbf{x}_c + B_c\mathbf{y} \\ \mathbf{u} &= -C_c\mathbf{x}_c \end{aligned}$$

- Which give the closed-loop dynamics

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_c \end{bmatrix} &= \begin{bmatrix} A & -B_uC_c \\ B_cC_y & A_c \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_c \end{bmatrix} + \begin{bmatrix} B_w & 0 \\ 0 & B_c \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix} \\ \mathbf{z} &= \begin{bmatrix} C_z & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_c \end{bmatrix} \\ \mathbf{y} &= \begin{bmatrix} C_y & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_c \end{bmatrix} + \mathbf{v} \end{aligned}$$

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

$$T = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} = T^{-1} \qquad \Rightarrow \qquad \begin{bmatrix} \mathbf{x} \\ \mathbf{n} \end{bmatrix} = T \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_c \end{bmatrix}$$

so that $A_{\text{cl}} \Rightarrow TA_{\text{cl}}T^{-1} \equiv \overline{A_{\text{cl}}}$ and when you work through the math, you get

$$\overline{A_{\text{cl}}} = \begin{bmatrix} A - B_uK & B_uK \\ 0 & A - LC_y \end{bmatrix}$$

- **Absolutely key points:**
  1. $\lambda_i(A_{\rm cl}) \equiv \lambda_i(\overline{A_{\rm cl}})$
  2. $\overline{A_{\rm cl}}$ is block upper triangular, so can find poles by inspection:

$$\det(sI - \overline{A_{\rm cl}}) = \det(sI - (A - B_u K)) \cdot \det(sI - (A - LC_y))$$

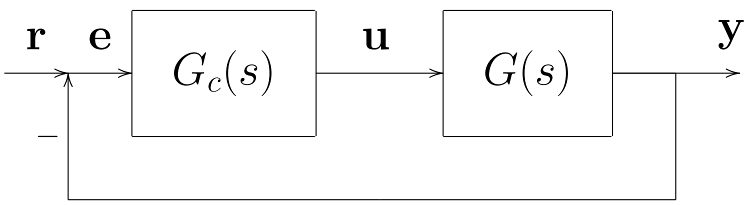
The closed-loop poles of the system consist of the union of the regulator and estimator poles

  - 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

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

  - Reason for this is that when implementing the controller, we often do not just feedback $-\mathbf{y}(t)$, but instead have to include a *reference command* $\mathbf{r}(t)$
  - Use **servo approach** and feed back $\mathbf{e}(t) = \mathbf{r}(t) - \mathbf{y}(t)$ instead



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

- 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

- Performance optimality of this strategy is a little harder to establish
  – Now saying more than just that the separation principle is a "good" idea ⇒ are trying to say that it is the "best" possible solution

- **Approach:**
  – Rewrite cost and system in terms of the estimator states and dynamics ⇒ recall we have access to these
  – Design a stochastic LQR for this revised system ⇒ full state feedback on $\hat{\mathbf{x}}(t)$

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

$$\begin{aligned} E[\mathbf{z}^T R_{zz}\mathbf{z}] &= E[\mathbf{x}^T R_{xx}\mathbf{x}] && \{\pm\hat{\mathbf{x}}\} \\ &= E[(\mathbf{x}-\hat{\mathbf{x}}+\hat{\mathbf{x}})^T R_{xx}(\mathbf{x}-\hat{\mathbf{x}}+\hat{\mathbf{x}})] && \{\tilde{\mathbf{x}}=\mathbf{x}-\hat{\mathbf{x}}\} \\ &= E[\tilde{\mathbf{x}}^T R_{xx}\tilde{\mathbf{x}}] + 2E[\tilde{\mathbf{x}}^T R_{xx}\hat{\mathbf{x}}] + E[\hat{\mathbf{x}}^T R_{xx}\hat{\mathbf{x}}] \end{aligned}$$

- Note that $\hat{\mathbf{x}}(t)$ is the minimum mean square estimate of $\mathbf{x}(t)$ given $\mathbf{y}(\tau)$, $\mathbf{u}(\tau)$, $t_0 \le \tau \le t$.
  – Key property of that estimate is that $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ are uncorrelated[24]

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

- Also,

$$E[\tilde{\mathbf{x}}^T R_{xx}\tilde{\mathbf{x}}] = E[\texttt{trace}(R_{xx}\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T)] = \texttt{trace}(R_{xx}Q)$$

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

- So, in summary we have:

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

[24] Gelb, pg 112

- Now the main part of the cost function can be rewritten as

$$\begin{aligned} J &= E\left\{\frac{1}{2}\int_{t_0}^{t_f}(\mathbf{z}^T(t)R_{\rm zz}\mathbf{z}(t)+\mathbf{u}^T(t)R_{\rm uu}\mathbf{u}(t))dt\right\} \\ &= E\left\{\frac{1}{2}\int_{t_0}^{t_f}(\hat{\mathbf{x}}^T(t)R_{\rm xx}\hat{\mathbf{x}}(t)+\mathbf{u}^T(t)R_{\rm uu}\mathbf{u}(t))dt\right\} \\ &\quad +\frac{1}{2}\int_{t_0}^{t_f}(\texttt{trace}(R_{\rm xx}Q))dt \end{aligned}$$

  - The last term is independent of the control $\mathbf{u}(t)$ $\Rightarrow$ it is only a function of the estimation error
  - Objective now is to choose the control $\mathbf{u}(t)$ to minimize the first term

- But first we need another key fact[25]: If the optimal estimator is

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + B_u\mathbf{u}(t) + L(t)(\mathbf{y}(t) - C_y\hat{\mathbf{x}}(t))$$

then by definition, the innovations process

$$\mathbf{i}(t) \equiv \mathbf{y}(t) - C_y\hat{\mathbf{x}}(t)$$

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

- Then we can rewrite the estimator as

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + B_u\mathbf{u}(t) + L(t)\mathbf{i}(t)$$

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

[25] Gelb, pg 317

- So combining the above, we must pick $\mathbf{u}(t)$ to minimize

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

  subject to the dynamics

  $$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + B_u\mathbf{u}(t) + L(t)\mathbf{i}(t)$$

  – Which is a strange looking Stochastic LQR problem

  – As we saw before, the solution is independent of the driving process noise

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

  – 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.

Simple Example

$$
\begin{aligned}
\dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \\
z &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x} \\
y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x} + v
\end{aligned}
$$

where in the LQG problem we have

$$
R_{\text{zz}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad R_{\text{uu}} = 1 \qquad R_{\text{vv}} = 1 \qquad R_{\text{ww}} = 1
$$

- Solve the SS LQG problem to find that

$$
\begin{aligned}
\texttt{Tr}[P_{ss} L_{ss} R_{\text{vv}} L_{ss}^T] &= 8.0 \quad \texttt{Tr}[Q_{ss} C_z^T R_{\text{zz}} C_z] = 2.8 \\
\texttt{Tr}[P_{ss} B_w R_{\text{ww}} B_w^T] &= 1.7 \quad \texttt{Tr}[Q_{ss} K_{ss}^T R_{\text{uu}} K_{ss}] = 9.1
\end{aligned}
$$

- Suggests to me that we need to improve the estimation error $\Rightarrow$ that R_{vv} is too large. Repeat with

$$
R_{\text{zz}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad R_{\text{uu}} = 1 \qquad R_{\text{vv}} = 0.1 \qquad R_{\text{ww}} = 1
$$

$$
\begin{aligned}
\texttt{Tr}[P_{ss} L_{ss} R_{\text{vv}} L_{ss}^T] &= 4.1 \quad \texttt{Tr}[Q_{ss} C_z^T R_{\text{zz}} C_z] = 1.0 \\
\texttt{Tr}[P_{ss} B_w R_{\text{ww}} B_w^T] &= 1.7 \quad \texttt{Tr}[Q_{ss} K_{ss}^T R_{\text{uu}} K_{ss}] = 3.7
\end{aligned}
$$

and

$$
R_{\text{zz}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad R_{\text{uu}} = 1 \qquad R_{\text{vv}} = 0.01 \qquad R_{\text{ww}} = 1
$$

$$
\begin{aligned}
\texttt{Tr}[P_{ss} L_{ss} R_{\text{vv}} L_{ss}^T] &= 3.0 \quad \texttt{Tr}[Q_{ss} C_z^T R_{\text{zz}} C_z] = 0.5 \\
\texttt{Tr}[P_{ss} B_w R_{\text{ww}} B_w^T] &= 1.7 \quad \texttt{Tr}[Q_{ss} K_{ss}^T R_{\text{uu}} K_{ss}] = 1.7
\end{aligned}
$$

- LQG analysis code

```
A=[0 1;0 0];%
Bu=[0 1]';%
Bw=[0 1]'; %
Cy=[1 0];%
Cz=[1 0;0 1];%
Rww=1;%
Rvv=1;%
Rzz=diag([1 1]);%
Ruu=1;%
[K,P]=lqr(A,Bu,Cz*Rzz*Cz',Ruu);%
[L,Q]=lqr(A',Cy',Bw*Rww*Bw',Rvv);L=L';%
N1=trace(P*(L*Rvv*L'))%
N2=trace(Q*(Cz'*Rzz*Cz))%
N3=trace(P*(Bw*Rww*Bw'))%
N4=trace(Q*(K'*Ruu*K))%
[N1 N2;N3 N4]
```

Stochastic Simulation

- 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:
  - δ (t): tilt angle of rotor thrust vector (radians)

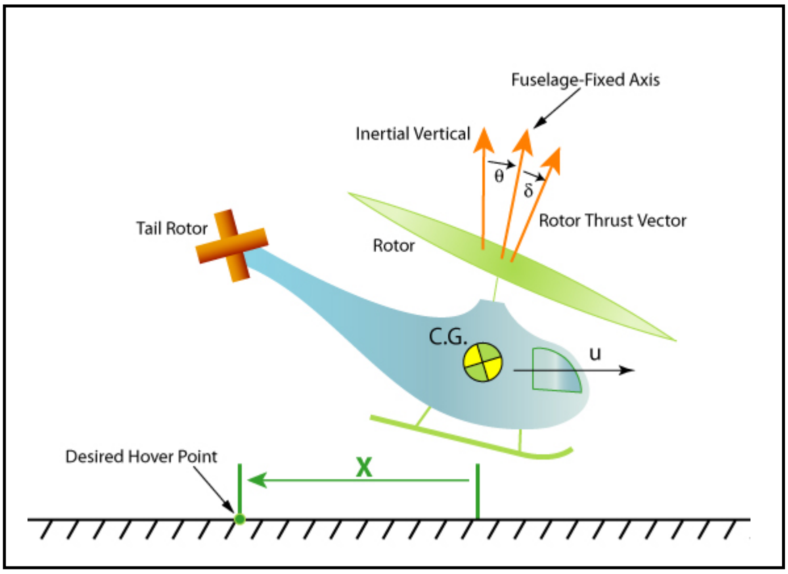


Figure by MIT OpenCourseWare.

Figure 12.1: Helicopter in Hover

- The linearized equation of motion are:

$$
\begin{aligned}
\dot{\theta}(t) &= q(t) \\
\dot{q}(t) &= -0.415q(t) - 0.011u(t) + 6.27\delta(t) - 0.011w(t) \\
\dot{u}(t) &= 9.8\theta(t) - 1.43q(t) - .0198u(t) + 9.8\delta(t) - 0.0198w(t) \\
\dot{x}(t) &= u(t)
\end{aligned}
$$

  - $w(t)$ 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.

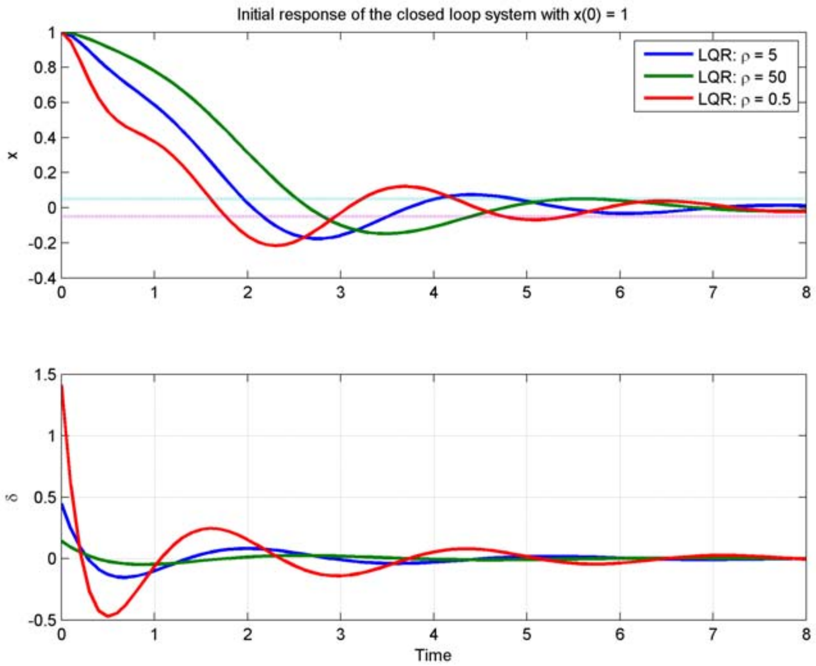


Figure 12.2: Results show that $R_{uu} = 5$ gives reasonable performance.

- 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:

$$\sqrt{E[x^2]} = 0.048 \qquad \sqrt{E[\delta^2]} = 0.017$$

- Then design a LQR for the augmented system and repeat the process.
  – With feedforward:

$$\sqrt{E[x^2]} = 0.0019 \qquad \sqrt{E[\delta^2]} = 0.0168$$

- 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 ζ 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 Δt 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

$$Q_d = R_{\text{ww}}/\Delta t$$

[26] Franklin and Powell, *Digital Control of Dynamic Systems*

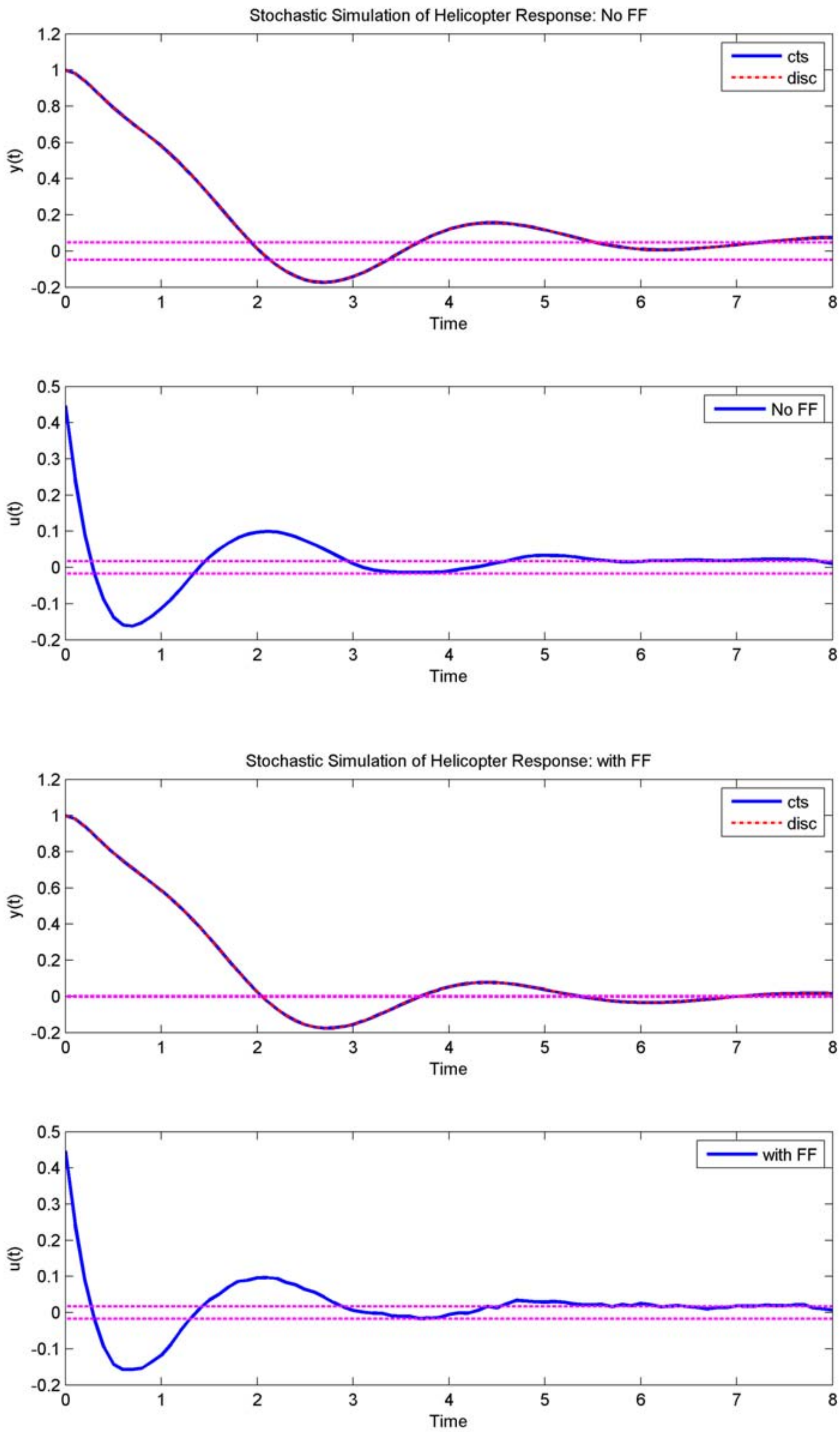


Figure 12.3: Stochastic Simulations with and without disturbance feedforward.

Helicopter stochastic simulation

```
% 16.323 Spring 2008
% Stochastic Simulation of Helicopter LQR
% Jon How
%
clear all, clf, randn('seed',sum(100*clock));
% linearized dynamics of the system
A = [ 0 1 0 0; 0 -0.415 -0.011 0;9.8 -1.43 -0.0198 0;0 0 1 0];
Bw = [0 -0.011 -0.0198 0]';
Bu = [0 6.27 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] = initial(A-Bu*K,zeros(4,1),Cz,0,x0,t);
[y2,x2] = initial(A-Bu*K2,zeros(4,1),Cz,0,x0,t);
[y3,x3] = initial(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

% shaping filter
Ah=-0.2;Bh=6;Ch=1;
% augment the filter dyanmics
Aa = [A Bw*Ch; zeros(1,4) Ah];
Bua = [Bu;0];
Bwa = [zeros(4,1); Bh];
Cza = [Cz 0];
Ka = [K 0]; % i.e. no dist FF
Acla = Aa-Bua*Ka; % close the loop using NO dist FF
Pass = lyap(Acla,Bwa*Rww*Bwa'); % compute SS response to the dist
vx = Cza*Pass*Cza'; % state resp
vd = Ka*Pass*Ka'; % control resp

zeta = sqrt(Rww/Ts)*randn(length(t),1); % discrete equivalent noise
[y,x] = lsim(Acla,Bwa,Cza,0,zeta,t,[x0;0]); % cts closed-loop sim
%
% second simulation approach: discrete time
%
Fa=c2d(ss(Acla,Bwa,Cza,0),Ts); % discretize the closed-loop dynamics
[dy,dx] = lsim(Fa,zeta,[],[x0;0]); % stochastic sim in discrete time
u = Ka*x'; % find control commands given the state response

% disturbance FF
[KK,SS,EE]=lqr(Aa,Bua,Cza'*Cza,rho); % now K will have dist FF
Acl=Aa-Bua*KK;
PP=lyap(Acl,Bwa*Rww*Bwa');
vxa = Cza*PP*Cza';
vda = KK*PP*KK';
[ya,xa] = lsim(Acl,Bwa,Cza,0,zeta,t,[x0;0]); % cts sim
F=c2d(ss(Acl,Bwa,Cza,0),Ts); % discretize the closed-loop dynamics
[dya,dxa] = lsim(F,zeta,[],[x0;0]); % stochastic sim in discrete time
ua = KK*xa'; % find control commands given the state response

```

```
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)
xlabel('Time');ylabel('u(t)');legend('No FF')
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)
xlabel('Time');ylabel('u(t)');legend('with FF')
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
```

LQG for Helicopter

- 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)

$$C_y = [\ I_4 \ \ 0\ ]$$

- Consider only partial vehicle state measurement

$$C_y = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- Set R_{vv} small.

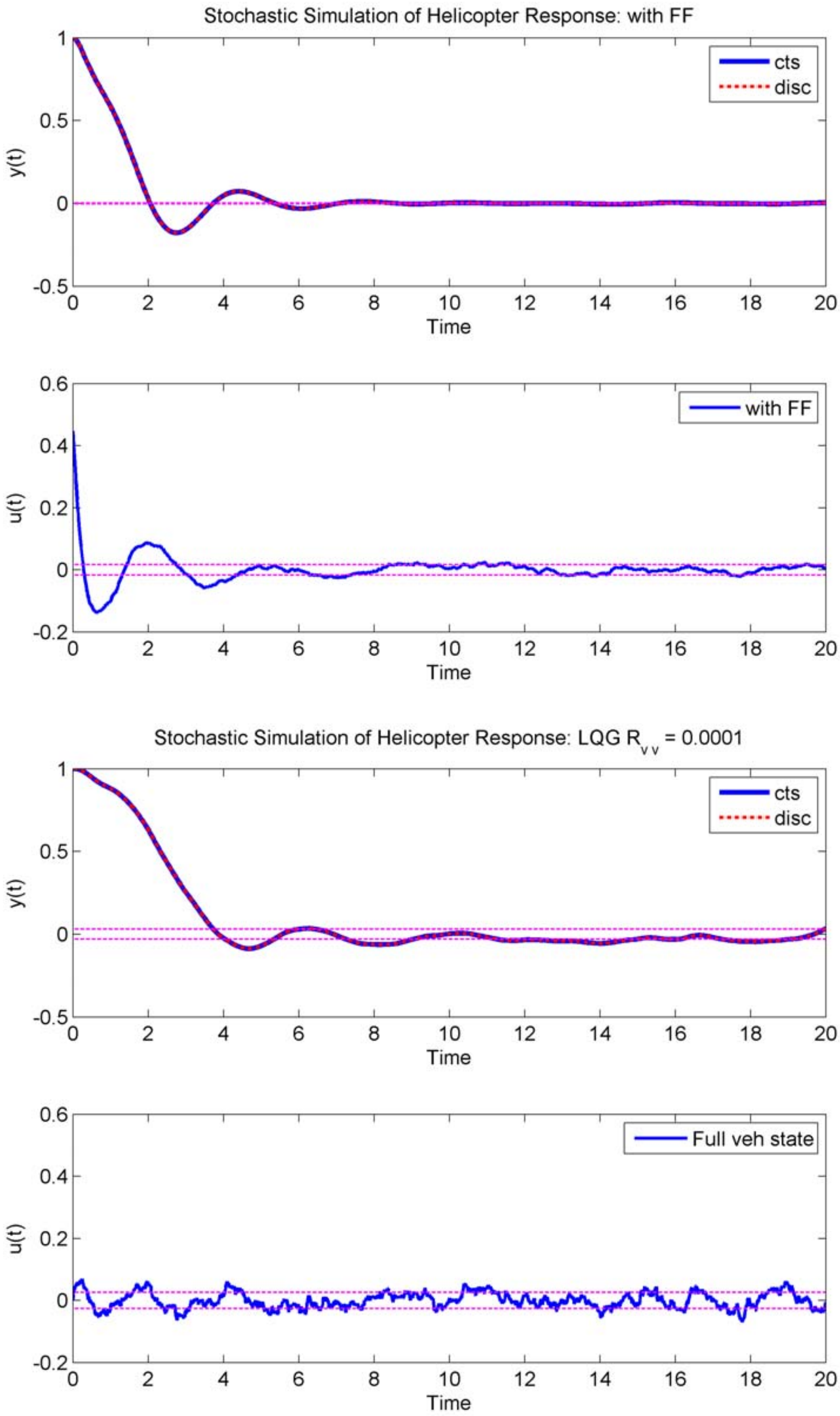


Figure 12.4: LQR with disturbance feedforward compared to LQG

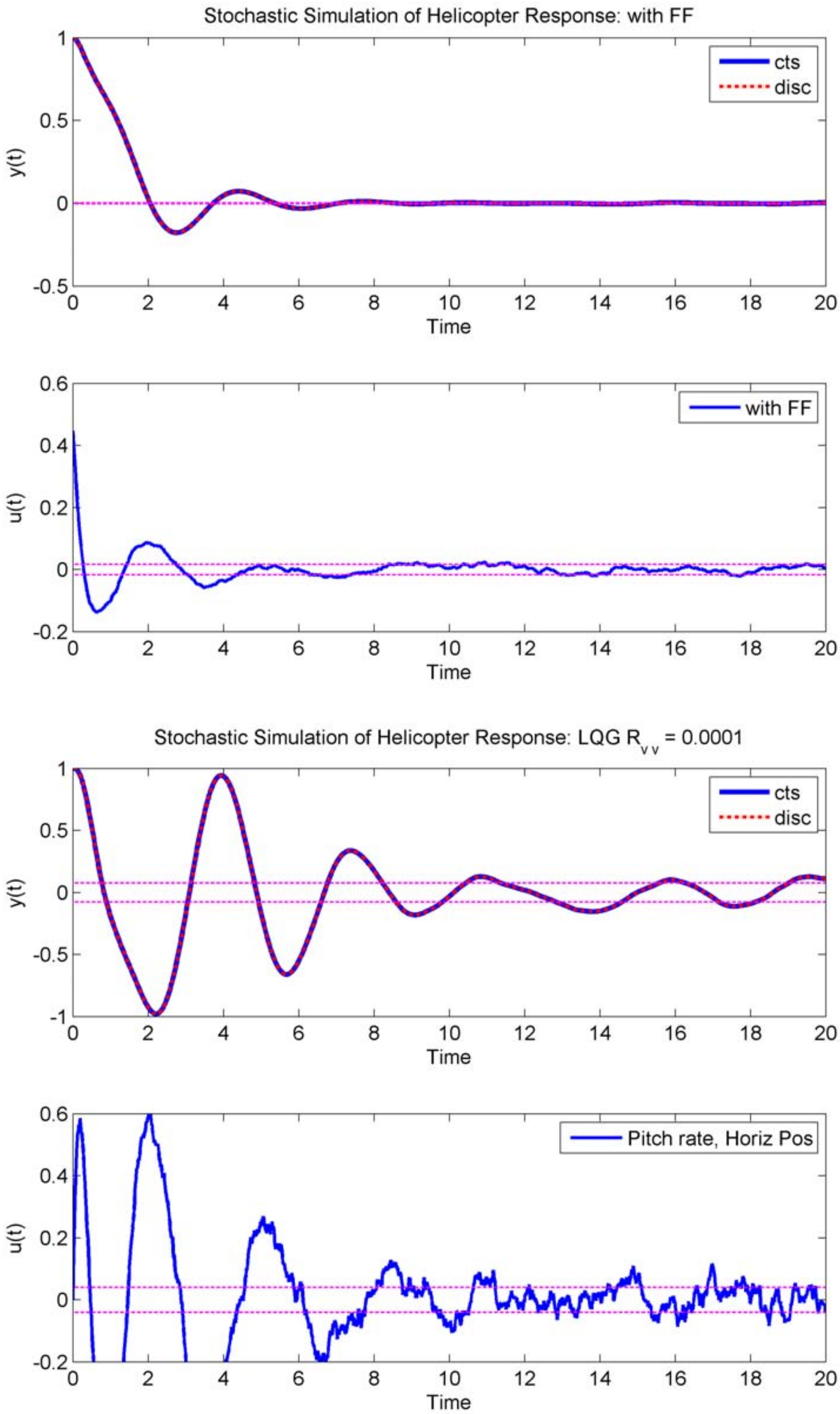


Figure 12.5: Second LQR with disturbance feedforward compared to LQG

Helicopter LQG

```
% 16.323 Spring 2008
% Stochastic Simulation of Helicopter LQR - from Bryson's Book
% Jon How
%
clear all, clf, randn('seed',sum(100*clock));
set(0,'DefaultAxesFontName','arial')
set(0,'DefaultAxesFontSize',12)
set(0,'DefaultTextFontName','arial')
% linearized dynamics of the system state=[theta q dotx x]
A = [ 0 1 0 0; 0 -0.415 -0.011 0;9.8 -1.43 -0.0198 0;0 0 1 0];
Bw = [0 -0.011 -0.0198 0]';
Bu = [0 6.27 9.8 0]';
Cz = [0 0 0 1];
Rxx = Cz'*Cz; Rww=1;
rho = 5;
% lqr control
[K,S,E]=lqr(A,Bu,Rxx,rho);

% initial response with given x0
x0 = [0 0 0 1]';
Ts=0.01; % small discrete step to simulate the cts dynamics
tf=20;t=0:Ts:tf;nt=length(t);
% Now consider shaped noise with shaping filter
Ah=-0.2;Bh=6;Ch=1;
% augment the filter dyanmics
Aa = [A Bw*Ch; zeros(1,4) Ah];
Bua = [Bu;0];
Bwa = [zeros(4,1); Bh];
Cza = [Cz 0];
x0a=[x0;0];
%zeta = Rww/sqrt(Ts)*randn(length(t),1); % discrete equivalent noise
zeta = sqrt(Rww/Ts)*randn(length(t),1); % discrete equivalent noise

%%%% Now consider disturbance FF
[KK,SS,EE]=lqr(Aa,Bua,Cza'*Cza,rho); % now K will have dist FF
Acl=Aa-Bua*KK;
PP=lyap(Acl,Bwa*Rww*Bwa');
vxa = Cza*PP*Cza'; %state
vda = KK*PP*KK'; %control
%
[ya,xa] = lsim(Acl,Bwa,Cza,0,zeta,t,x0a); % cts sim
F=c2d(ss(Acl,Bwa,Cza,0),Ts); % discretize the closed-loop dynamics
[dya,dxa] = lsim(F,zeta,[],x0a); % stochastic sim in discrete time
ua = KK*xa'; % find control commands given the state response

%%%% Now consider Output Feedback Case
% Assume that we can only measure the system states
% and not the dist one
FULL=1;
if FULL
    Cya=eye(4,5); % full veh state
else
    Cy=[0 1 0 0;0 0 0 1]; % only meas some states
    Cya=[Cy [0;0]];
end
Ncy=size(Cya,1);Rvv=(1e-2)^2*eye(Ncy);
[L,Q,FF]=lqr(Aa',Cya',Bwa*Rww*Bwa',Rvv);L=L';% LQE calc
%closed loop dyn
Acl_lqg=[Aa -Bua*KK;L*Cya Aa-Bua*KK-L*Cya];
Bcl_lqg=[Bwa zeros(5,Ncy);zeros(5,1) L];
Ccl_lqg=[Cza zeros(1,5)];Dcl_lqg=zeros(1,1+Ncy);
x0_lqg=[x0a;zeros(5,1)];
zeta_lqg=zeta;
% now just treat this as a system with more sensor noise acting as more
% process noise
for ii=1:Ncy
  zeta_lqg = [zeta_lqg sqrt(Rvv(ii,ii)/Ts)*randn(nt,1)];% discrete equivalent noise
```

```
end
[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
[dya_lqg,dxa_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)
plot([0 max(t)],sqrt(vxa)*[1 1],'m--',[0 max(t)],-sqrt(vxa)*[1 1],'m--','LineWidth',1);
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')
hold on;
plot([0 max(t)],sqrt(vda)*[1 1],'m--',[0 max(t)],-sqrt(vda)*[1 1],'m--','LineWidth',1);
axis([0 tf -0.2 .6])
hold off
print -f3 -r300 -dpng  heli_lqg_1.png;

figure(4);clf
subplot(211)
plot(t,ya_lqg,'LineWidth',3)
hold on;
plot(t,dya_lqg,'r-.','LineWidth',2)
plot([0 max(t)],sqrt(vx_lqg)*[1 1],'m--',[0 max(t)],-sqrt(vx_lqg)*[1 1],'m--','LineWidth',1);
hold off
xlabel('Time');ylabel('y(t)');legend('cts','disc')
title(['Stochastic Simulation of Helicopter Response: LQG R_{v v} = ',num2str(Rvv(1,1))])
subplot(212)
plot(t,ua_lqg,'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([0 max(t)],sqrt(vu_lqg)*[1 1],'m--',[0 max(t)],-sqrt(vu_lqg)*[1 1],'m--','LineWidth',1);
axis([0 tf -0.2 .6])
hold off
if FULL
    print -f4 -r300 -dpng  heli_lqg_2.png;
else
    print -f4 -r300 -dpng  heli_lqg_3.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 $R_{\rm uu}$ and $R_{\rm vv}$ to get a reasonably balanced performance.
   – Took $R_{\rm ww} = 0.1$ and tuned $R_{\rm vv}$

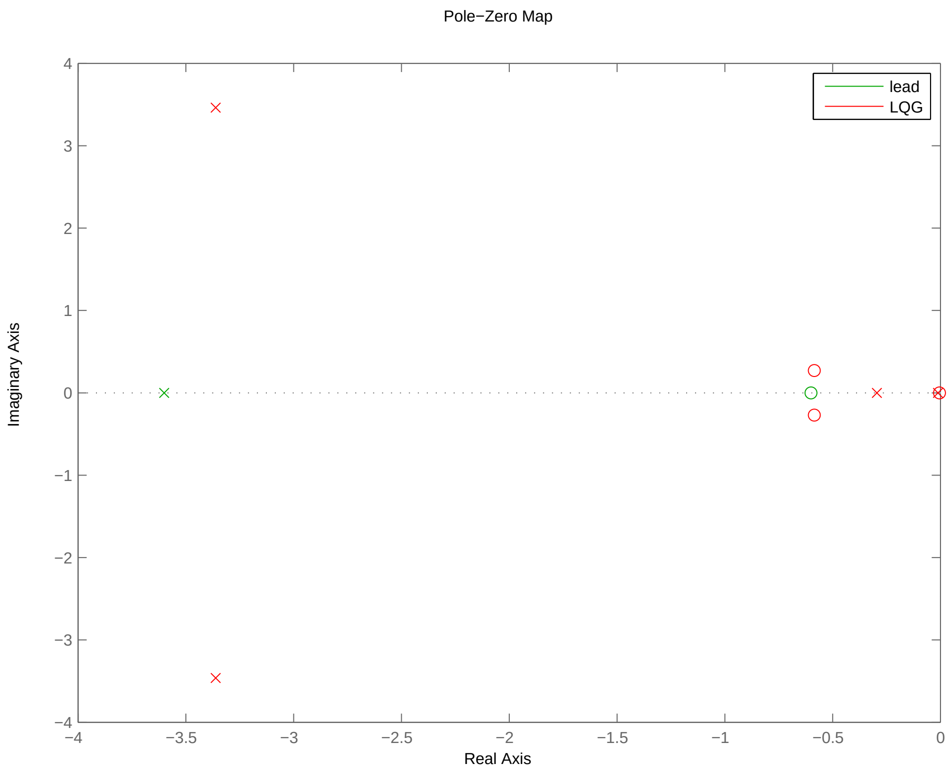


Figure 12.6: B747: Compensators

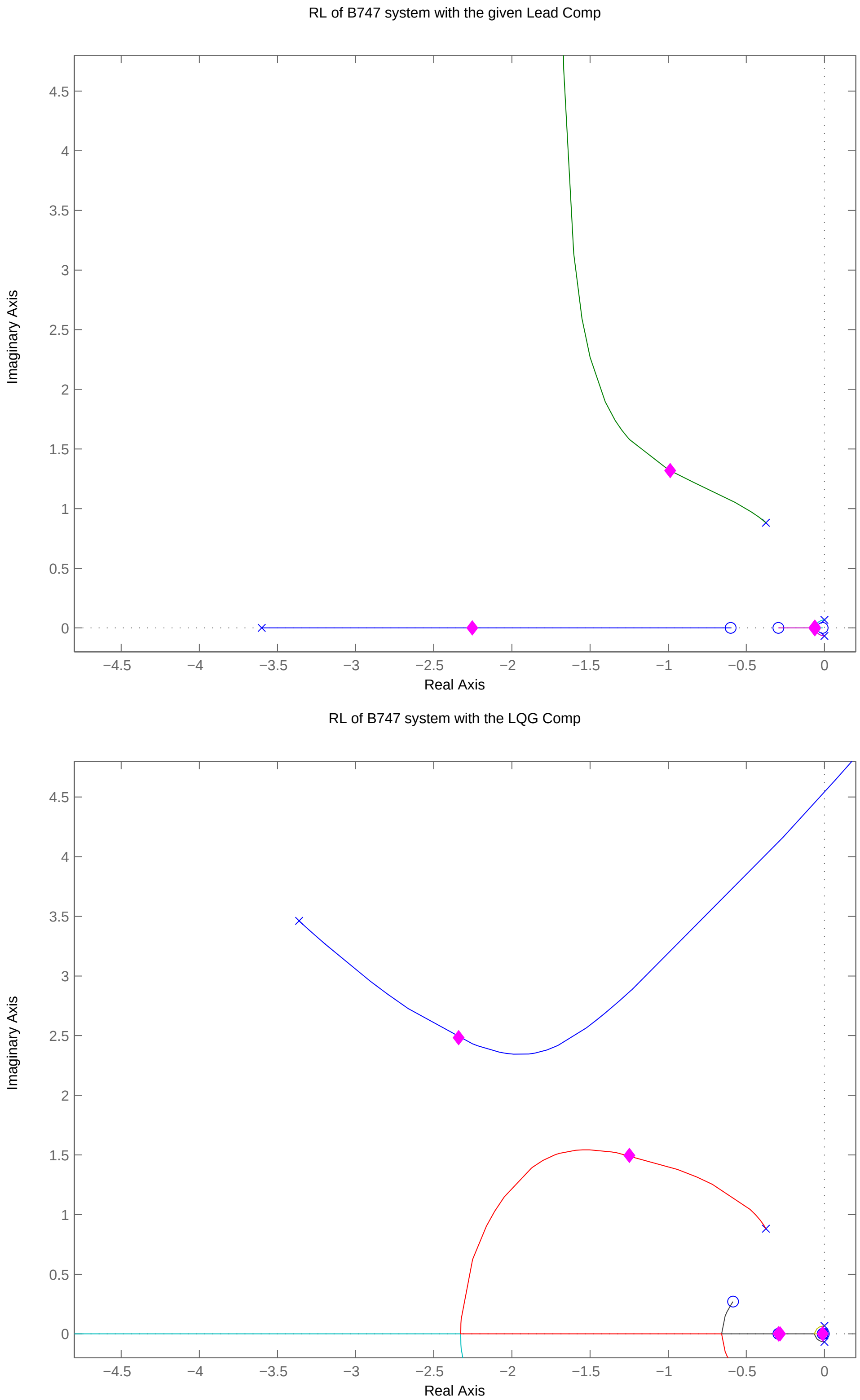


Figure 12.7: B747: root locus (Lead on left, LQG on right shown as a function of the overall compensator gain)

3. Compare the Bode plots of the lead compensator and LQG designs

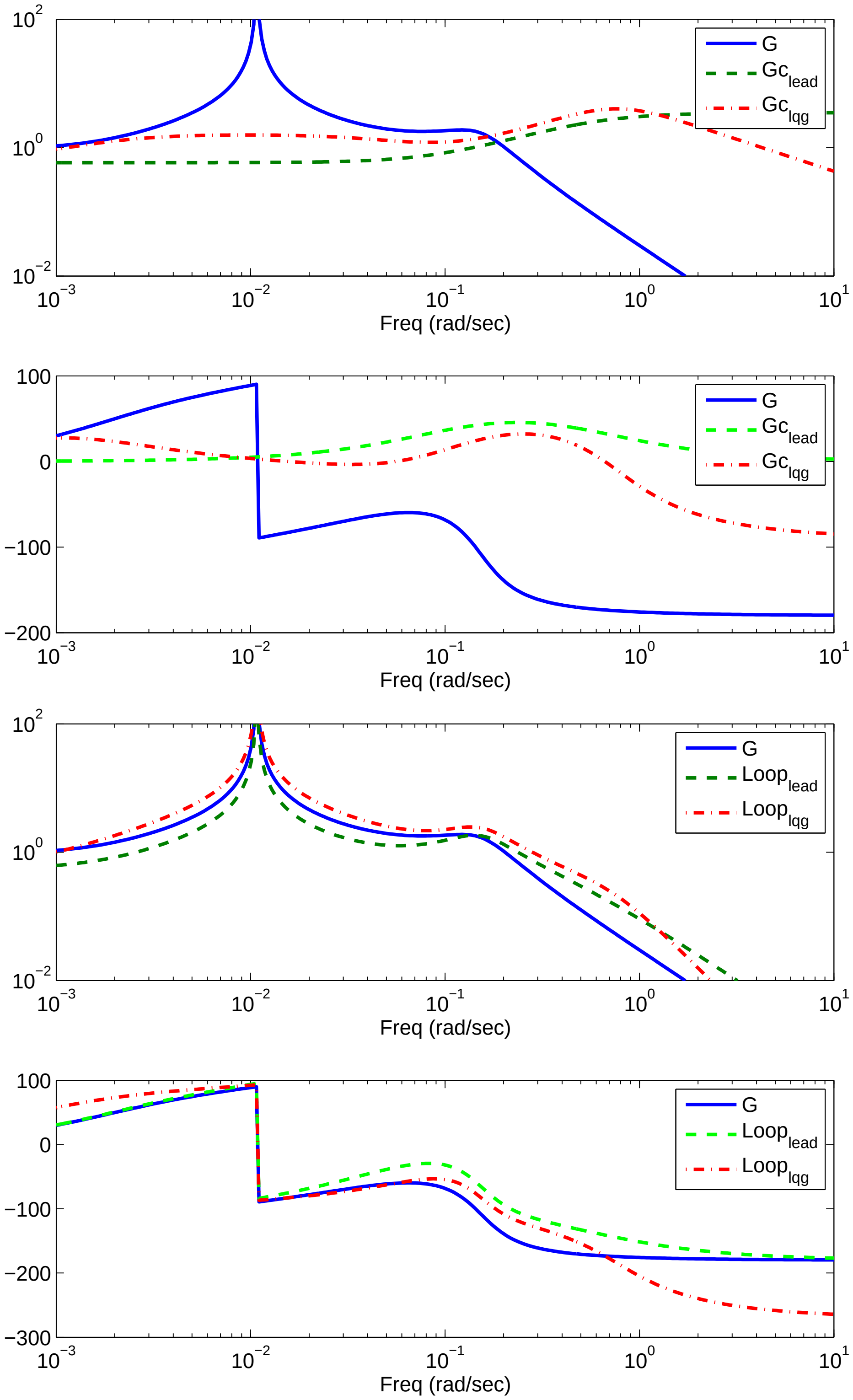


Figure 12.8: B747: Compensators and loop TF

4. Consider the closed-loop TF for the system

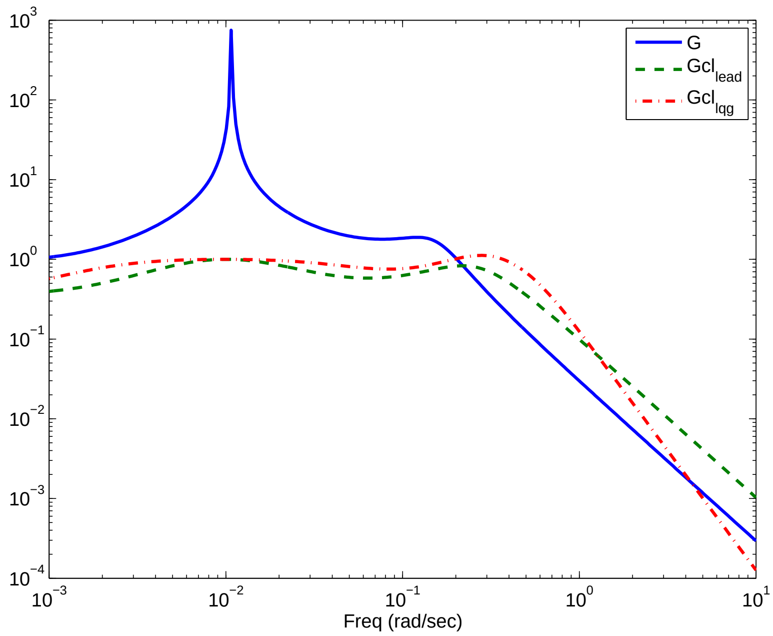


Figure 12.9: B747: closed-loop TF

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

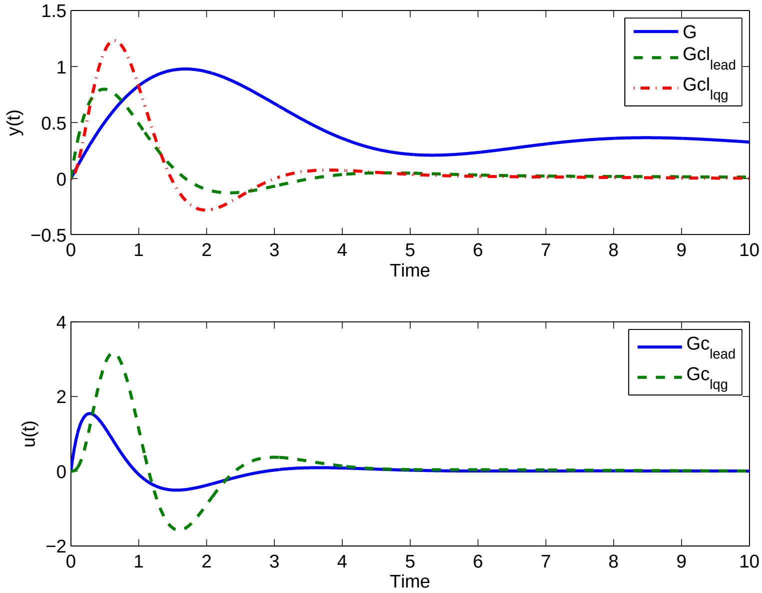


Figure 12.10: B747: Impulse response

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.

B747 LQG

```
% 16.323 B747 example
% Jon How, MIT, Spring 2007
%
clear all
set(0,’DefaultAxesFontName’,’arial’)
set(0,’DefaultAxesFontSize’,12)
set(0,’DefaultTextFontName’,’arial’)

gn=1.16*conv([1 .0113],[1 .295]);
gd=conv([1 0 .0676^2],[1 2*.375 .375^2+.882^2]);
% 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]=cloop(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;Ruu=0.01;Rww=0.1;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*(l*Rvv*l’))%
N2=trace(Q*(c’*Rzz*c))%
N3=trace(P*(b*Rww*b’))%
N4=trace(Q*(k’*Ruu*k))%
N=[N1 N2 N1+N2;N3 N4 N3+N4]

[ncl,dcl]=cloop(conv(gn,knl),conv(gd,kdl)); % CLP with lqg
gcl=freqresp(ncl,dcl,2*pi*f*sqrt(-1)); % CLP with lqg
[[roots(dc);0;0;0] roots(dcl)]
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 compesantors
setlines(2)
legend(’G’,’Gc_{lead}’,’Gc_{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([Gclead])),’g’)
semilogx(f,180/pi*unwrap(phase([Gclqg])),’r’)
xlabel(’Freq (rad/sec)’)
hold off
setlines(2)
legend(’G’,’Gc_{lead}’,’Gc_{lqg}’)

figure(6);clf;
subplot(211)
loglog(f,abs([g g.*Gclead g.*Gclqg])) % Bode of Loop transfer function
```

```
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
title('RL of B747 system with the given Lead Comp')
figure(4);clf;rlocus(conv(gn,knl),conv(gd,kdl));axis(2*[-2.4 0.1 -0.1 2.4])
hold on;plot(roots(dcl)+sqrt(-1)*eps,'md','MarkerFaceColor','m');hold off
title('RL of B747 system with the LQG Comp')

% time simulations
Ts=0.01;
[y1,x,t]=impulse(gn,gd,[0:Ts:10]);
[y2]=impulse(nc,dc,t);
[y3]=impulse(ncl,dcl,t);
[ulead]=lsim(kn,kd,y2,t); % noise free sim
[ulqg]=lsim(knl,kdl,y3,t); % noise free sim

figure(5);clf;
subplot(211)
plot(t,[y1 y2 y3])
xlabel('Time')
ylabel('y(t)')
setlines(2)
legend('G','Gcl_{lead}','Gcl_{lqg}')
subplot(212)
plot(t,[ulead ulqg])
xlabel('Time')
ylabel('u(t)')
setlines(2)
legend('Gc_{lead}','Gc_{lqg}')

figure(7)
pzmap(tf(kn,kd),'g',tf(knl,kdl),'r')
legend('lead','LQG')

print -depsc -f1 b747_1.eps;jpdf('b747_1')
print -depsc -f2 b747_2.eps;jpdf('b747_2')
print -depsc -f3 b747_3.eps;jpdf('b747_3')
print -depsc -f4 b747_4.eps;jpdf('b747_4')
print -depsc -f5 b747_5.eps;jpdf('b747_5')
print -depsc -f6 b747_6.eps;jpdf('b747_6')
print -depsc -f7 b747_7.eps;jpdf('b747_7')
```