Lectures 12-13

State Estimation, H_2 and H_∞ Optimal Control

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

ECE 6095-4121
Dynamic Modeling and Control of Mechatronic Systems

State Estimation, H_2 and H_∞ Optimal Control

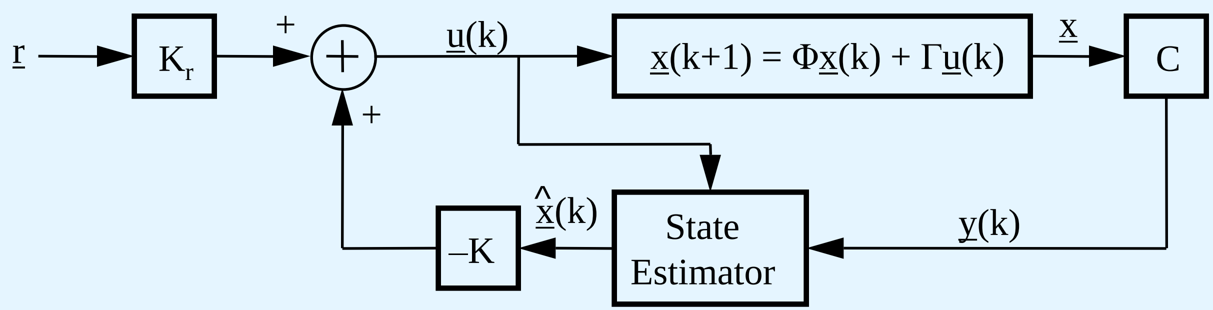
1. **State Estimation: Introduction**
   - Observability requirement
2. **State Observer**
   - Structure of observer
   - Properties of estimation error
   - Observer pole placement (SO, MO cases)
   - Deadbeat Observer
   - Kalman Filter (KF)
   - Reduced order observers
   - Time delay modifications
3. **Implementation Considerations**
   - Composite CL observer and controller (LQG when observer = KF)
   - Poles and zeros of composite system
   - Loop transfer recovery (LTR)
4. **Examples**
   - Antenna positioning
   - Satellite control
5. **H_2 Output Feedback Optimal Controller**
   - General Control Problem Formulation
   - LQG: a special H_2 output feedback optimal controller
6. **H_∞ Output Feedback Optimal Controller**
   - Two Riccati equation solution: continuous design with gain transfomation and direct digital design
   - H_∞ Loop shaping

State Estimation

- State $\underline{x}(k)$ is often not measurable directly
  - Measure $\underline{y}(k) = C\underline{x}(k)$, a linear combination of states
- Assume measurements made with no noise/error
- Objective:
  - Develop an estimate $\hat{\underline{x}}(k)$ of the state $\underline{x}(k)$ suitable for use in SVFB or for other purposes
  - Use available information about system input and output

$$\{\underline{u}(j),\ j < k\},\ \{\underline{y}(j),\ j \leq k\}$$

  - Need to generate state estimate on-line



  - Will using $\hat{\underline{x}}(k)$ as a substitute for $\underline{x}(k)$ work?
- Design issues
  - Desired properties of state estimator
  - Expect/force a linear estimator (system is linear, so why not the estimator?)
  - How fast must $\hat{\underline{x}}(k) \rightarrow \underline{x}(k)$?
- State estimate is useful even in non-control applications (e.g., decisionmaking using $\underline{x}$).

"Observation" of System State

- What can be done to estimate $\underline{x}(k)$?
  - Consider $u(k) = 0$:

$$\underline{x}(k+1) = \Phi\,\underline{x}(k)$$
$$\underline{y}(k) = C\,\underline{x}(k)$$
$$\underline{x}(0) = \text{unknown initial condition}$$

- Estimate $\underline{x}(0) = [x_1(0), \ldots, x_n(0)]'$ from the output measurements $\{\underline{y}(0), \underline{y}(1), \ldots, \underline{y}(n-1)\}$

$$\underline{y}(0) = C\,\underline{x}(0)$$
$$\underline{y}(1) = C\,\underline{x}(1) = C\Phi\,\underline{x}(0)$$
$$\underline{y}(2) = C\,\underline{x}(2) = C\Phi^2\,\underline{x}(0)$$
$$\vdots$$
$$\underline{y}(n-1) = C\,\underline{x}(n-1) = C\Phi^{n-1}\,\underline{x}(0)$$

$$\begin{bmatrix} \underline{y}(0) \\ \underline{y}(1) \\ \vdots \\ \underline{y}(n-1) \end{bmatrix} = \underbrace{\begin{bmatrix} \text{----}\ C\ \text{----} \\ \text{---}\ C\Phi\ \text{---} \\ \vdots \\ \text{--}\ C\Phi^{n-1}\ \text{--} \end{bmatrix}}_{H_0'} \underbrace{\begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_n(0) \end{bmatrix}}_{\underline{x}(0)}$$

  - If H_0' has full column rank (invertible in SO case), it is possible to find $\underline{x}(0)$
  - Obtain $\underline{x}(0)$ after n independent measurements at step $k = n - 1$ (for SO case)
- Once $\underline{x}(0)$ is obtained, $\underline{x}(k) = \Phi^k\,\underline{x}(0)$ for $k > 0$.
  - Eventually, we would like to obtain state estimates recursively.

Observability

- A discrete system is completely observable if

$$\det\begin{bmatrix} | & | & & | \\ C^T & \Phi^T C^T \ldots & & (\Phi^T)^{n-1} C^T \\ | & | & & | \end{bmatrix} = \det(H_0) \neq 0$$

- Physical interpretation: All modes show up in the output, either directly or indirectly.
- Observability is a property of only $\{\Phi, C\}$. Actually, all you need is detectability (unobserved modes are stable).
- Controllability-observability duality:

$$\Phi \rightarrow \Phi', \quad \Gamma \rightarrow C'$$

- Continuous-discrete relationship

If original continuous system was observable,

$$\det\begin{bmatrix} | & | & & | \\ C' & A'C' & \cdots & (A')^{n-1}C' \\ | & | & & | \end{bmatrix} \neq 0$$

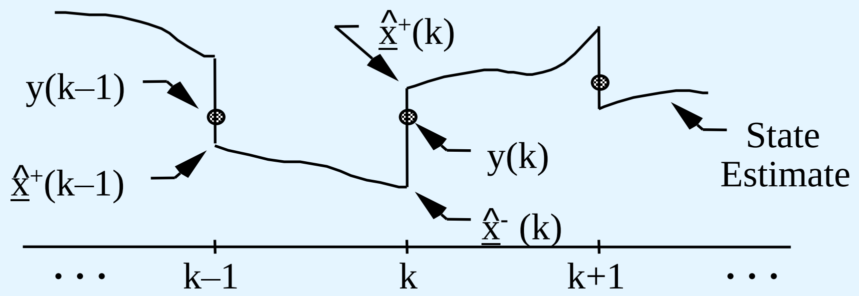
then equivalent discrete system is observable provided $h \neq M(2\pi/\omega_{c0})$, M = integer, where

ω_{c0} = imaginary part of any eigenvalue of A that is on $j\omega$-axis

- Observability will be a necessary condition for state estimation
  - det (H_0) and/or det $(H_0'H_0)$ is often used as a "measure of observability"

A Question of Notation

- Must consider state estimate in 2 parts
  - Estimate will undergo a discontinuity at a measurement point k
  - Need to distinguish between the estimate $\underline{\hat{x}}^-(k)$ prior to making the measurement of y(k), and the estimate $\underline{\hat{x}}^+(k)$ after making the measurement y(k).



<u>Define</u>:

- $\underline{\hat{x}}(k \mid k-1)$ = estimate of $\underline{x}(k)$ prior to obtaining the measurement y(k) at time t = kh
  = estimate of $\underline{x}(k)$ from $\{\underline{y}(k–1), \underline{y}(k–2), \ldots\} = \underline{\hat{x}}^-(k)$
- $\underline{\hat{x}}(k \mid k)$ = estimate of $\underline{x}(k)$ after obtaining and processing the mesurement y(k) at t = kh
  = estimate of $\underline{x}(k)$ from $\{\underline{y}(k), \underline{y}(k–1), \ldots\} = \underline{\hat{x}}^+(k)$
  - Obviously $\underline{\hat{x}}(k \mid k)$ is the better estimate of $\underline{x}(k)$
  - Desire a recursive estimation scheme:

time k–1 time k

$$\cdots \quad \underset{\underline{\hat{x}}^+(k-1)}{\underline{\hat{x}}(k-1 \mid k-1)} \longrightarrow \underset{\underline{u}(k-1)}{\oplus} \longrightarrow \underset{\hat{x}^-(k)}{\underline{\hat{x}}(k \mid k-1)} \longrightarrow \underset{\underline{y}(k)}{\oplus} \longrightarrow \underset{\underline{\hat{x}}^+(k)}{\underline{\hat{x}}(k \mid k)} \quad \cdots$$

Structure of the Estimator

- "Prediction" estimate, $\hat{\underline{x}}(k \mid k-1)$ from $\hat{\underline{x}}(k-1 \mid k-1)$
  - Since no measurements are made over (k–1, k) the only way to estimate $\hat{\underline{x}}(k \mid k-1)$ is via the state equation

known input over (k–1, k]

$$\underline{x}(k) = \Phi\, \underline{x}(k-1) + \Gamma \underline{u}(k-1)$$

$$\Rightarrow \quad \hat{\underline{x}}(k \mid k-1) = \Phi\, \hat{\underline{x}}(k-1 \mid k-1) + \Gamma \underline{u}(k-1)$$

  - Alternate notation $\hat{\underline{x}}^{-}(k) = \Phi\, \hat{\underline{x}}^{+}(k-1) + \Gamma \underline{u}(k-1)$

- "Update" estimate, $\hat{\underline{x}}(k \mid k)$ from $\hat{\underline{x}}(k \mid k-1)$
  - How to include the measurement y(k) ?

$$\left.\begin{matrix} \hat{\underline{x}}(k \mid k-1) \\ \underline{y}(k) \end{matrix}\right\} \xrightarrow[\text{ALGORITHM}]{} \hat{\underline{x}}(k \mid k)$$

$$\Rightarrow \quad \hat{\underline{x}}(k \mid k) = \hat{\underline{x}}(k \mid k-1) + L[\underline{y}(k) - \underbrace{C\, \hat{\underline{x}}(k \mid k-1)}_{\hat{\underline{y}}(k \mid k-1)}]$$

where

$$\hat{\underline{y}}(k \mid k-1) \triangleq C\, \hat{\underline{x}}(k \mid k-1)$$

= best prediction of what the measurement at step k should be

$$\underline{\nu}(k) \triangleq \underline{y}(k) - \hat{\underline{y}}(k \mid k-1)$$

= difference between what is actually measured at step k and what we expect to measure (innovation)

L = n x m arbitrary gain matrix, to be determined

  - Alternate notation $\hat{\underline{x}}^{+}(k) = \hat{\underline{x}}^{-}(k) + L[\underline{y}(k) - C\, \hat{\underline{x}}^{-}(k)]$

These relations are called a dynamic "observer" - Requires a model of system: (Φ, Γ, C)

The Estimation Error

- Observer "starts" at $k = 0$ with $\hat{\underline{x}}(0\,|\,-1)$
  - $\hat{\underline{x}}(0\,|\,-1) \triangleq$ estimate of initial state $\underline{x}(0)$ based on all prior information
  - Usually $\hat{\underline{x}}(0\,|\,-1) = \underline{0}$
- Obtain evolution of prediction error, $\tilde{\underline{e}}(k\,|\,k-1) \triangleq \underline{x}(k) - \hat{\underline{x}}(k\,|\,k-1)$

$$\hat{\underline{x}}(k+1|\,k) = \Phi\,\hat{\underline{x}}(k\,|\,k) + \Gamma\underline{u}(k)$$

$$\hat{\underline{x}}(k\,|\,k-1) + L[\underline{y}(k) - C\,\hat{\underline{x}}(k\,|\,k-1)]$$

$$\Rightarrow\quad \hat{\underline{x}}(k+1|\,k) = \Phi\,\hat{\underline{x}}(k\,|\,k-1) + \Gamma\underline{u}(k) + \Phi L[\underline{y}(k) - C\,\hat{\underline{x}}(k\,|\,k-1)]$$

  - Subtract from system equation $\underline{x}(k+1) = \Phi\,\underline{x}(k) + \Gamma\underline{u}(k)$

$$\tilde{\underline{e}}(k+1|\,k) = \Phi\tilde{\underline{e}}(k\,|\,k-1) - \Phi L[\underbrace{\underline{y}(k) - C\,\hat{\underline{x}}(k\,|\,k-1)}_{C\tilde{\underline{e}}\,(k\,|\,k-1)}]$$

$$\Rightarrow\quad \tilde{\underline{e}}(k+1|\,k) = (\Phi - \Phi LC)\tilde{\underline{e}}(k\,|\,k-1)$$

  - Initial condition $\tilde{\underline{e}}(0\,|\,-1) = \underline{x}(0) - \hat{\underline{x}}(0\,|\,-1) = \underline{x}(0)$, [if $\hat{\underline{x}}(0\,|\,-1) = 0$]

$$\tilde{\underline{e}}(k|\,k-1) = (\Phi - \Phi LC)^k\,\underline{x}(0)$$

- Selection of observer gain L
  - Want $\tilde{\underline{e}} \rightarrow \underline{0}$ rapidly
  - Rate at which $\tilde{\underline{e}} \rightarrow \underline{0}$ depends on eigenvalues of $\Phi - \Phi LC$
  - Choose L so that eigenvalues of $\Phi - \Phi LC$ are within unit circle
  - Since $\Phi - \Phi LC = \Phi(\Phi - LC\,\Phi)\Phi^{-1}$, eigenvalues of $\Phi - \Phi LC \equiv$ eigenvalues of $\Phi - LC\Phi$
- Update error, $\tilde{\underline{e}}(k\,|\,k) \triangleq \underline{x}(k) - \hat{\underline{x}}(k\,|\,k) = (\Phi - LC\Phi)\tilde{\underline{e}}(k-1\,|\,k-1)$

Observer Pole Placement Problem

- Select L so that eigenvalues of $\Phi - LC\Phi$ are at preselected locations within unit circle

  $\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_n \rightarrow p_e(z) = z^n + \tilde{d}_1 z^{n-1} + \cdots + \tilde{d}_n =$ estimator desired characteristic polynomial

  $= | zI - (\Phi - LC\Phi) |$

- Re-formulate as a "control" problem
  - Select L' so that eigenvalues of $\Phi^T - [\Phi^T C^T] L^T$ are at desired locations
  - Like pole placement for $\Phi - \Gamma K$ with associations

  $$\Phi \Longleftrightarrow \Phi^T, \quad \Gamma \Longleftrightarrow \Phi^T C^T, \quad K \Longleftrightarrow L^T$$

- Ackermann formula (for single output state stimation)

$$L' = [\,0 \quad 0 \quad \cdots \quad 1\,] \underbrace{\begin{bmatrix} | & | & & | \\ \Phi^T C^T & (\Phi^T)^2 C^T & \cdots & (\Phi^T)^n C^T \\ | & | & & | \end{bmatrix}^{-1}}_{\Phi' \cdot H_0} p_e(\Phi')$$

$$p_e(\Phi) = \Phi^n + \tilde{d}_1 \Phi^{n-1} + \cdots + \tilde{d}_n I$$

- Multi-output case
  - Kautsky's robust eigen structure assignment algorithm for multi-output case
  - Sylvester Equation: Let $\hat{\underline{x}}(k \mid k-1) = X^{-1} \underline{z}(k)$ and find requirements on X

  State: $\underline{x}(k) = \Phi\, \underline{x}(k-1) + \Gamma \underline{u}(k-1)$ Estimate equation: $\underline{z}(k+1) = \Phi_e\, \underline{z}(k) + \Gamma_e \underline{u}(k) + \Delta \underline{y}(k)$

  $X \underline{e}(k+1|k) = \underline{z}(k+1) - X\underline{x}(k+1) = \Phi_e \underline{z}(k) + \Gamma_e \underline{u}(k) + \Delta \underline{y}(k) - X\Phi \underline{x}(k) - X\Gamma \underline{u}(k)$

  Use $\underline{y}(k) = C\underline{x}(k)$ while adding and subtracting $\Phi_e X \underline{x}(k)$, we get

  $= \Phi_e X e(k) + (\Gamma_e - X\Gamma)\underline{u}(k) + (\Phi_e X - X\Phi + \Delta C)\underline{x}(k)$

  $\Rightarrow X\Phi - \Phi_e X = \Delta C$ *Sylevester observer equation* and $\Gamma_e = X\Gamma$ *and* Φ_e stable

Selection of Observer CL Poles

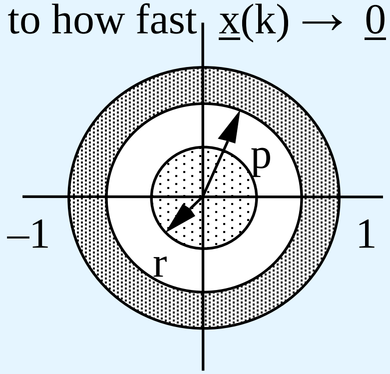
- Depends on what you will need to do with the estimate. Will it be used for SVFB or not ?

  (1) Only interested in a good state estimate, $\hat{\underline{x}} \to \underline{x}$
  - No tie-ins or constraints imposed by SVFB
  - Place poles within unit circle depending on how fast desire $\hat{\underline{x}} \to \underline{x}$
  - E.g., if $|\tilde{\lambda}_i| \leq r < 1$ then error $\to 0$ as r^k (if $r = 0.5$, error decreases by 50% each step, with ~ 12% error after 3 steps)

  (2) Anticipate using $\hat{\underline{x}}$ for $\underline{x}$ in SVFB control
  - What matters is how fast $\underline{\tilde{e}}(k) \to \underline{0}$ compared to how fast $\underline{x}(k) \to \underline{0}$
  - Desire $\underline{\tilde{e}}(k) \to \underline{0}$ faster by ~ 2 to 3 times
  - E.g., if primary poles of $\Phi - \Gamma K$ satisfy

    $$p \leq |\lambda_i| < 1$$

    then place observer poles $\tilde{\lambda}_i$ inside circle of radius $r = p^2$ to $r = p^3$ (p = magnitude of primary control poles)



- Best to "uniformly" space $\tilde{\lambda}_i$ on semi-circle of radius r, with $\mathrm{Re}\{\tilde{\lambda}_i\} > 0$
- Deadbeat observer
  - Special case when $r = 0$ => all observer poles @ $z = 0$
  - Any initial error $\underline{\tilde{e}}(0\,|\,-1) \to \underline{0}$ in n steps

  => obtain perfect estimate after n measurements $y(0), y(1), \ldots, y(n-1)$

  => $\hat{\underline{x}}(n-1|\,n-1) = \underline{x}(n-1)$, and all subsequent estimates are exact

Example of State Estimation

- Satellite model, $G(s) = 1/s^2$

$$u \longrightarrow \boxed{\frac{1}{s}} \xrightarrow{x_2} \boxed{\frac{1}{s}} \longrightarrow x_1 = y \qquad \begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned}$$

  - Can only measure $y(kh) = x_1(kh)$; build estimator for $\underline{x}(k)$
- Equivalent discrete system

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} h^2/2 \\ h \end{bmatrix} u(k); \quad y(k) = \underbrace{[\,1 \;\; 0\,]}_{C}\, \underline{x}(k)$$

- Check observability

$$H_o = [\, C' \;\; \Phi'C' \,] = \begin{bmatrix} 1 & 1 \\ 0 & h \end{bmatrix} \Rightarrow \text{ observable (as long as } h \neq 0)$$

Design observer

- Desired characteristic polynomial, $p_e(z) = z^2 + \tilde{d}_1 z + \tilde{d}_2$
- Observer gains, $L = p_e(\Phi)\begin{bmatrix} -\ C\Phi\ - \\ -\ C\Phi^2\ - \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = p_e(\Phi)\begin{bmatrix} 1 & h \\ 1 & 2h \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 - \tilde{d}_2 \\ (1 + \tilde{d}_1 + \tilde{d}_2)/h \end{bmatrix}$
- Deadbeat observer

$$\tilde{d}_1 = \tilde{d}_2 = 0 \text{ (poles @ } z = 0)$$

$$L = \begin{bmatrix} 1 \\ 1/h \end{bmatrix} \quad \text{(as } h \to 0\text{, need large L)}$$

Mechanics of Observer Dynamics

- Observer algorithm; initialize $\hat{x}^-(0) = \underline{0}$ (usually)
  - (1) measure y(k), form $v(k) = y(k) - \hat{y}(k)$
  - (2) update $\underline{\hat{x}}^+(k) = \underline{\hat{x}}^-(k) + Lv(k)$
  - (3) propagate $\underline{\hat{x}}^-(k+1) = \Phi\, \underline{\hat{x}}^+(k) + \Gamma u(k)$
  - (4) $\hat{y}(k+1) = C\, \underline{\hat{x}}^-(k+1)$
  - (5) $k = k + 1$
- In previous example, $h = 1$, $L = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} = \begin{bmatrix} 1 - \tilde{d}_2 \\ 1 + \tilde{d}_1 + \tilde{d}_2 \end{bmatrix}$

$$v(k) = y(k) - \hat{x}_1^-(k)$$

$$\left.\begin{aligned} \hat{x}_1^+(k) &= \hat{x}_1^-(k) + L_1 v(k) \\ \hat{x}_2^+(k) &= \hat{x}_2^-(k) + L_2 v(k) \end{aligned}\right\} \text{update}$$

$$\left.\begin{aligned} \hat{x}_1^-(k+1) &= \hat{x}_1^+(k) + x_2^+(k) + \tfrac{1}{2} u(k) \\ \hat{x}_2^-(k+1) &= \hat{x}_2^+(k) + u(k) \end{aligned}\right\} \text{propagate}$$

- Deadbeat case, $L_1 = 1$, $L_2 = 1$. Open-loop u = {–1, 0.5, –0.3, 0.4, ...}

	Actual System				Observer				
k	x_1	x_2	u	y	$\hat{x}_1^-$	$\hat{x}_2^-$	v	$\hat{x}_1^+$	$\hat{x}_2^+$
0	1.0	-0.3	-1.0	1.0	0.0	0.0	1.0	1.0	1.0
1	0.2	-1.3	0.5	0.2	1.5	0.0	-1.3	0.2	-1.3
2	-0.85	-0.8	-0.3	-0.85	-0.85	-0.8	0.0	-0.85	-0.8
3	-1.8	-1.1		-1.8	-1.8	-1.1	0.0	-1.8	-1.1
⋮ etc.							⋮		

- u(k) = control input over time interval (k, k+1]
- Need only 2 measurements to obtain $\underline{x}$ exactly => $\underline{\hat{x}}^+(1) = \underline{x}(1)$
- Subsequent $\underline{\hat{x}}^+$, $\underline{\hat{x}}^-$ are correct as long as we know state equations and system inputs

Discrete-time Steady State Kalman Filter

- Estimation in presence of noise

$$\underline{x}(k+1) = \Phi\, \underline{x}(k) + \Gamma \underline{u}(k) + E\underline{w}(k)$$

white noise, zero mean, cov W

$$\underline{y}(k) = C\, \underline{x}(k) + \underline{v}(k)$$

white noise, zero mean, cov V

- Results in Kalman filter for $\hat{\underline{x}}(k \mid k)$, $\hat{\underline{x}}(k \mid k-1)$
- Identical to observer, but with a different scheme to find steady state gains L
- "Prediction" => $\hat{\underline{x}}(k \mid k-1) = \Phi\, \hat{\underline{x}}(k-1 \mid k-1) + \Gamma \underline{u}(k-1)$
- "Update" => $\hat{\underline{x}}(k \mid k) = \hat{\underline{x}}(k \mid k-1) + L\,[\underline{y}(k) - C\hat{\underline{x}}(k \mid k-1)]$

  L = n x m Kalman gain matrix = $\Sigma C^T (C\Sigma C^T + V)^{-1}$
- Σ is the steady state error covariance matrix (of prediction error) given by

$$\Sigma = \Phi\Sigma\Phi^T + EWE^T - \Phi\Sigma C^T (C\Sigma C^T + V)^{-1} C\Sigma\Phi^T = \Phi(\Sigma^{-1} + C^T V^{-1} C)^{-1}\Phi^T + EWE^T = \Phi(I_n + \Sigma C^T V^{-1} C)^{-1}\Sigma\Phi^T + EWE^T$$

- Suppose want to minimize

$$J = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N} [\underline{x}^T(k) Q \underline{x}(k) + \underline{u}^T(k) R \underline{u}(k)] = E\{\underline{x}^T(k) Q \underline{x}(k) + \underline{u}^T(k) R \underline{u}(k)\}$$

Recalling $\underline{x}(k) = \hat{\underline{x}}(k \mid k-1) + \underline{e}(k \mid k-1)$ and that $\hat{\underline{x}}(k \mid k-1) \perp \underline{e}(k \mid k-1)$, we have

$$J = E\{\hat{\underline{x}}^T(k \mid k-1) Q \hat{\underline{x}}(k \mid k-1) + \underline{u}^T(k) R \underline{u}(k)\} + Trace(Q\Sigma) = Trace(PEWE^T + Q\Sigma)$$

where P is the solution of control DARE:

$$P = \Phi^T P \Phi + Q - \Phi^T P \Gamma (\Gamma^T P \Gamma + R)^{-1} \Gamma^T P \Phi = \Phi^T P (I_n + \Gamma R^{-1} \Gamma^T P)^{-1} \Phi + Q$$

and $\underline{u}(k) = -K \hat{\underline{x}}(k \mid k-1) = -(\Gamma^T P \Gamma + R)^{-1} \Gamma^T P \Phi \hat{\underline{x}}(k \mid k-1)$

Separation Principle:
Controller and Estimator gains
can be computed separately

Implementation of Observer – Controller Pair

- Use feedback control $\underline{u}(k) = K_r\underline{r}(k) - \underline{\underline{K\,\hat{\underline{x}}(k \mid k)}}$
  - $\hat{\underline{x}}(k \mid k)$ is best estimate of $\underline{x}(k)$ at step k
  - includes the latest information y(k)
  - $\hat{\underline{x}}(k \mid k-1)$ is not as good as $\hat{\underline{x}}(k \mid k)$
  - K_r, K obtained via usual SVFB control design
- Algorithm at any particular k

0	Initialize at k = 0 $\hat{\underline{x}}^-$, $\hat{\underline{y}} = C\,\hat{\underline{x}}^-$	
1	Sample: $\underline{y}(k)$, $\underline{r}(k)$	
2	$\hat{\underline{x}}^+ = \hat{\underline{x}}^- + L[\underline{y}(k) - \hat{\underline{y}}]$	Update: $\hat{\underline{x}}(k \mid k)$
3	$\underline{u} = K_r\underline{r}(k) - K\,\hat{\underline{x}}^+$	Control
4	Return $\underline{u}(k)$	
5	$\hat{\underline{x}}^- = \Phi\,\hat{\underline{x}}^+ + \Gamma\underline{u}(k)$	Propagate: $\hat{\underline{x}}(k+1 \mid k)$
6	$\hat{\underline{y}} = C\,\hat{\underline{x}}^-$ $\rightarrow$ Return	

- Steps 2 and 3 require np+nm MADDS to obtain new control
- Steps 5 and 6 set up $\hat{\underline{x}}^-$ and $\hat{\underline{y}}$ for the next cycle.
  This shifts $n^2 + n\,(m+p)$ MADDS to "wait" portion of cycle

Some Practical Considerations

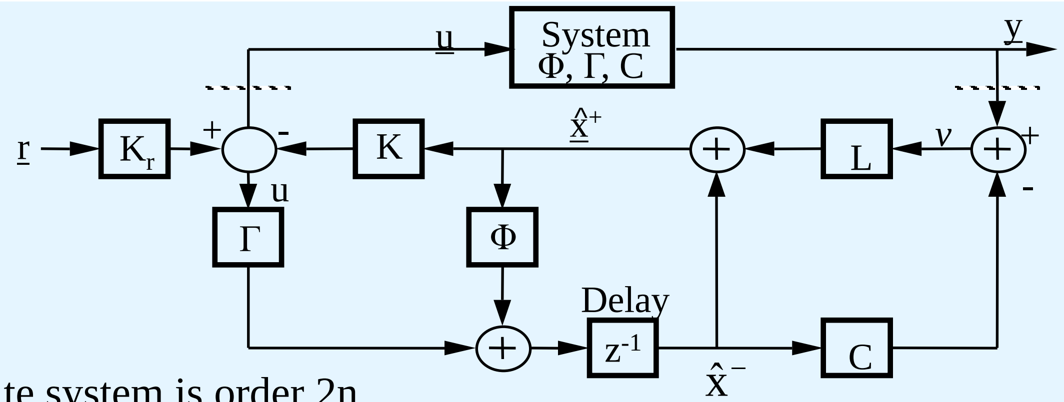
- Propagate step 5 assumes that the $\underline{u}(k)$ computed will actually be applied to the system
  - Apply software limits to u, Δu, etc., to match any system or hardware constraints/nonlinearities, or else
  - Modify algorithm to use <u>actual</u> control:

$$
\begin{array}{llll}
 & & 1. & \text{Sample } y(k),\ r(k),\ \boxed{\underline{u}(k-1)} \\
5 & \rightarrow & 1a. & \hat{\underline{x}}^- = \Phi\, \hat{\underline{x}}^+ + \Gamma \underline{u} \quad \leftarrow \text{obtains } \hat{\underline{x}}(k \mid k-1) \text{ at step } k \\
6 & \rightarrow & 1b. & \hat{\underline{y}} = C\, \hat{\underline{x}}^- \\
 & & 2. & \hat{\underline{x}}^+ = \hat{\underline{x}}^- + L[\underline{y}(k) - \hat{\underline{y}}] \\
 & & 3. & \underline{u} = K_r \underline{r}(k) - K\, \hat{\underline{x}}^+ \\
 & & 4. & \text{Return } \underline{u}(k)
\end{array}
$$

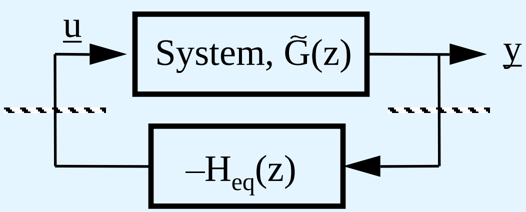
  - Requires significantly more computation before $\underline{u}(k)$ is obtained
    => larger computational delay
- Any system time delay must be modeled in step 5: $\hat{\underline{x}}^- = \Phi\, \hat{\underline{x}}^+ + \Gamma_1 \underline{u}(k-M-1) + \Gamma_0 \underline{u}(k-M)$
- Try to keep observer gains with $| L_i |$ small
  - Minimize amplification of $\underline{y}(k)$ measurement error
  - $\| L \|$ increases as observer poles → 0
- Observer requires a "model" of system $\{\Phi, \Gamma, C\}$
  - Mismatch will yield estimation error
  - In a CL application, the feedback will reduce some of the effects of mismatch between “model” and system
  - Large modeling errors can cause the estimate to divergence
- If $\underline{u}(k)$ must be returned prior to sampling y(k) use

$$\underline{u}(k) = K_r \underline{r}(k) - K\, \hat{\underline{x}}(k \mid k-1)$$

Composite CL Observer and Controller



- Composite system is order $\underline{2n}$
  - n-th order system and n-th order observer
- Alternative representation ($\underline{r} = 0$, $\underline{\hat{x}}^-(0) = \underline{0}$)
  - Input $\underline{y}$ to observer-controller and output $\underline{u}$



=> can compute

$$\underline{u}(z) = -H_{eq}(z)\,\underline{y}(z)$$

- Develop equation for $\underline{\hat{x}}(k \mid k)$ in terms of $\underline{y}(k)$

$$\underline{\hat{x}}(k+1 \mid k+1) = (I - LC)\,\underline{\hat{x}}(k+1 \mid k) + L\underline{y}(k+1)$$

$$\Phi\,\underline{\hat{x}}(k \mid k) + \Gamma\underline{u}(k) = (\Phi - \Gamma K)\,\underline{\hat{x}}(k \mid k)$$

$$\Rightarrow \underline{\hat{x}}(k+1 \mid k+1) = \tilde{\Phi}\,\underline{\hat{x}}(k \mid k) + L\underline{y}(k+1); \quad \tilde{\Phi} = (I - LC)(\Phi - \Gamma K); \quad -\underline{u}(k) = K\,\underline{\hat{x}}(k \mid k)$$

- Take z-transform $[y(k+1) \rightarrow zy(z)]$ => $H_{eq}(z) = zK(zI - \tilde{\Phi})^{-1}L$

- Provides a "modern" control approach to the design of "classical“ series/FB compensators.
  - Use $LG_{ain}(z) = \tilde{G}(z)H_{eq}(z)$ for stability analysis (ϕ_m, ω_c)

Why LQG/Loop Transfer Recovery ?

- Recall Loop gain $LG_{ain}(z) = \tilde{G}(z)H_{eq}(z) = zC(zI_n - \Phi)^{-1}\Gamma K(zI_n - \tilde{\Phi})^{-1}L; \tilde{\Phi} = (I - LC)(\Phi - \Gamma K)$
- Double integrator with h=0.1 sec. Select $Q = \text{Diag}(1,0)$; R=0.1
- SVFB gain vector: $K = [2.7889\ 2.3617]$. Controller poles: [0.8749 - 0.1107i 0.8749 + 0.1107i]

$$\text{SVFB } LG_{ain}, L_1 = K(zI-\Phi)^{-1}\Gamma = \frac{0.25z - 0.222}{z^2 - 2z + 1};\ \phi_m = 58.2^0$$

- $W = 10^{-3}I, V = 10^{-2}I \Rightarrow L = [0.3575\ 0.2585]^T$; filter poles: [0.6873 - 0.1654i 0.6873 + 0.1654i]

$$\text{Filter } LG_{ain}, L_2 = C(zI-\Phi)^{-1}L = \frac{0.3575z - 0.3316}{z^2 - 2z + 1};\phi_m = 68^0$$

- Controller-Kalman filter combination gives $LG_{ain}, L_3 = zC(zI_n - \Phi)^{-1}\Gamma K(zI_n - \tilde{\Phi})^{-1}L$

$$L_3 = \frac{0.0080376\ (z+1)\ (z-0.864)}{(z-1)^2\ (z^2\ - 1.375z + 0.4997)};\phi_m = 11^0\quad \text{(what happened?)}$$

- Loop transfer recovery (LTR)

SVFB $LG_{ain}(z)$, *an mxm matrix*, $L_1(z) = K(zI_n - \Phi)^{-1}\Gamma$ has good phase margin properties

so does, Kalman filter LG_{ain}, *a pxp matrix*, $L_2(z) = C(zI_n - \Phi)^{-1}L$ by duality

But, *LQG* $LG_{ain}(z)$, *a pxp matrix*, $L_3(z) = zC(zI_n - \Phi)^{-1}\Gamma K(zI_n - \tilde{\Phi})^{-1}L$ need not have good ϕ_m

Alternately, *LQG* $LG_{ain}(z)$, *an mxm matrix*, $L_4(z) = zK(zI_n - \tilde{\Phi})^{-1}LC(zI_n - \Phi)^{-1}\Gamma$ need not

Can we make $L_3(z)$ or $L_4(z)$ equal to $L_1(z)$ or $L_2(z)$? Yes for minimum phase and $p = m$ systems

LQG/Loop Transfer Recovery

- How to make $zC(zI_n-\Phi)^{-1}\Gamma K(zI_n-\tilde{\Phi})^{-1}L = K(zI_n-\Phi)^{-1}\Gamma$?

  A dual approach to make $zC(zI_n-\Phi)^{-1}\Gamma K(zI_n-\tilde{\Phi})^{-1}L = C(zI_n-\Phi)^{-1}L$
- Procedure:
  - Design SVFB by choosing K by setting $Q = C^TC$ and $R=\rho I \ni \underline{\sigma}\,[K(zI_n-\Phi)^{-1}\Gamma]$ is large at low frequencies and small at high frequencies for robust stability.
  - Then, design a Kalman filter gain, L with $V = CC^T$ and $W = \gamma\Gamma\Gamma^T$.

    As $\gamma \to 0$, $zC(zI_n-\Phi)^{-1}\Gamma K(zI_n-\tilde{\Phi})^{-1}L = K(zI_n-\Phi)^{-1}\Gamma$
- Loop transfer recovery (LTR) applied to Satellite example

  For $Q = C^TC$ and R=0.5, SVFB $\text{LG}_{ain} = K(zI-\Phi)^{-1}\Gamma = \dfrac{0.1678\text{ z - }0.1548}{z^2-2z+1} \Rightarrow \phi_m = 60.5^0$

  Choose W=10^{-8} $\Gamma\Gamma^T, V = CC^T \Rightarrow$ Filter $L_2(z) = C(zI_n-\Phi)^{-1}L$ has $\phi_m = 65.5^0$

  This choice gives $LQG\ LG_{ain}(z) = \dfrac{5.424\text{e-}05\ z^3 + 2.743\text{e-}007\ z^2 - 5.397\text{e-}05\ z}{z^4 - 3.824\ z^3 + 5.487\ z^2 - 3.501\ z + 0.8385}; \phi_m = 59^0$

- Disadvantages of LTR procedure
  - Applicable to minimum phase systems only (it is basically cancelling zeros)
  - Lightly damped zeros cause problems as well
  - Typically results in high gains
  - Ad hoc process

Transfer Function of Composite CL Observer and Controller

- 2n-th order system (=> 2n poles)
- Examine transfer function $\underline{y}(z) = T(z)\,\underline{u}(z)$
  - Obtain via state equations (2n)
- State $\underline{x}(k)$ evolution with $\underline{u}(k) = K_r\underline{r}(k) - K\,\hat{\underline{x}}(k\,|\,k)$

$$\underline{x}(k+1) = \Phi\underline{x}(k) - \Gamma K\,\hat{\underline{x}}(k\,|\,k) + K_r\Gamma\underline{r}(k)$$

$$\hat{\underline{x}}(k\,|\,k) = (I - LC)\,\hat{\underline{x}}(k\,|\,k-1) + L\overbrace{C\underline{x}(k)}^{\underline{y}(k)}$$

$$=> \quad \underline{x}(k+1) = (\Phi - \Gamma KLC)\,\underline{x}(k) - \Gamma K(I - LC)\,\hat{\underline{x}}(k\,|\,k-1) + K_r\Gamma\,\underline{r}(k)$$

- State estimate $\hat{\underline{x}}(k\,|\,k\text{-}1)$ evolution

$$\hat{\underline{x}}(k+1|\,k) = \Phi\,\hat{\underline{x}}(k\,|\,k) + \Gamma\underline{u}(k)$$

$$\hat{\underline{x}}(k+1|\,k) = (\Phi - \Gamma K)\,\hat{\underline{x}}(k\,|\,k) + K_r\Gamma\underline{r}(k)$$

$$\hat{\underline{x}}(k\,|\,k) = (I - LC)\,\hat{\underline{x}}(k\,|\,k-1) + LC\,\underline{x}(k)$$

$$=> \quad \hat{\underline{x}}(k+1|\,k) = (\Phi - \Gamma K)(I - LC)\,\hat{\underline{x}}(k\,|\,k-1) + (\Phi - \Gamma K)LC\,\underline{x}(k) + K_r\Gamma\,\underline{r}(k)$$

- Augmented system, $\underline{x}_a(k) \triangleq [\,\underline{x}(k),\ \ \hat{\underline{x}}(k\,|\,k-1)]^T$

$$\begin{bmatrix} \underline{x}(k+1) \\ \hat{\underline{x}}(k+1|\,k) \end{bmatrix} = \underbrace{\left[\begin{array}{c|c} \Phi - \Gamma KLC & -\Gamma K(I-LC) \\ \hline (\Phi - \Gamma K)LC & (\Phi - \Gamma K)(I - LC) \end{array}\right]}_{\Phi^*} \underbrace{\begin{bmatrix} \underline{x}(k) \\ \hat{\underline{x}}(k\,|\,k\text{-}1) \end{bmatrix}}_{\underline{x}_a(k)} + \underbrace{\begin{bmatrix} K_r\Gamma \\ K_r\Gamma \end{bmatrix}}_{\Gamma^*} \underline{r}(k)$$

$$y(k) = \underbrace{[\,C \;\vdots\; 0\,]}_{C^*}\,\underline{x}_a(k)$$

$$T(z) = C^{*\prime}(zI - \Phi^*)^{-1}\Gamma^* = \frac{N(z)}{D(z)}$$

- Messy! Easier to obtain D(z) first, then get N(z)

Poles and Zeros of Composite T(z)

- $D(z) = |\, zI - \Phi^* \,| =$ characteristic polynomial of CL system

$$\left[\begin{array}{c|c} zI-\Phi+\Gamma KLC & \Gamma K(I-LC) \\ \hline -\Phi LC+\Gamma KLC & \underbrace{zI-(\Phi-\Gamma K)(I-LC)}_{-\Phi(I-LC)+\Gamma K(I-LC)} \end{array}\right] \rightarrow \left[\begin{array}{c|c} zI-\Phi+\Gamma K & \Gamma K(I-LC) \\ \hline 0 & zI-\Phi(I-LC) \end{array}\right]$$

$$\Rightarrow \; D(z) = |\, zI - \Phi+\Gamma K \,| \cdot |\, zI - \Phi(I-LC) \,| = p_d(z) \cdot p_e(z)$$

- Poles of composite system are those of the controller $\{\lambda_i, \cdots, \lambda_n\}$ and the observer $\{\tilde{\lambda}_i, \cdots, \tilde{\lambda}_n\}$

- $N(z) = \left|\begin{array}{c|c} zI-\Phi^* & -\Gamma^* \\ \hline C^* & 0 \end{array}\right|$

$$\left|\begin{array}{c|c|c} zI-\Phi^* & -K_r\Gamma & \\ & -K_r\Gamma & \\ \hline C & 0 & 0 \end{array}\right| \xrightarrow{\text{Via row and column manipulation}} \left|\begin{array}{c|c|c} zI-\Phi & -K_r\Gamma & \Gamma K(I-LC) \\ \hline C & 0 & 0 \\ \hline 0 & 0 & zI-\Phi(I-LC) \end{array}\right|$$

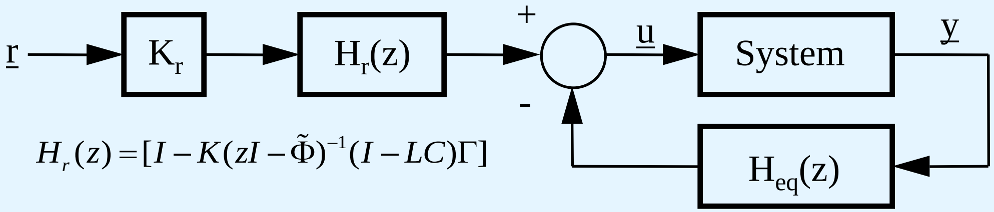
$$\Rightarrow \; N(z) = \underbrace{\left|\begin{array}{c|c} zI-\Phi & -K_r\Gamma \\ \hline C & 0 \end{array}\right|}_{K_r \,*\, \text{numerator of open-loop system}} \cdot \underbrace{|\, zI-\Phi(I-LC) \,|}_{p_e(z)}$$

$N(z) = K_r\, B(z)\, p_e(z)$, $\quad B(z) =$ open-loop numerator

- Transfer function $\quad \Rightarrow \; T(z) = \dfrac{K_r B(z)\, p_e(z)}{p_d(z)\, p_e(z)} = \dfrac{K_r\, B(z)}{p_d(z)}$
  - Same as SVFB case! True in MIMO case as well.
  - Observer dynamics are "transparent" in steady-state (after initial transient in state estimate $\underline{\tilde{e}} \rightarrow \underline{0}$)
  - SVFB with observer does not modify system zeros

Command Inputs to Observer-Controller System

- Alternate representation in output feedback form (when r ≠ 0)



$$H_r(z) = [I - K(zI - \tilde{\Phi})^{-1}(I - LC)\Gamma]$$

$$H_{eq}(z) = zK(zI - \tilde{\Phi})^{-1}L$$

$$\tilde{\Phi} = (I - LC)(\Phi - \Gamma K)$$

Includes a feedforward compensator $H_r(z)$ on $\underline{r}(k)$

- Observer-controller implementation generally preferable

- Recall, for a SVFB loop, $\underline{u}(k) = K_r\,\underline{r}(k) - K\,\underline{x}(k)$
  - K_r chosen so that DC gain $\underline{r} \rightarrow \underline{y} = 1$ in steady-state

$$K_r = \left[C(zI - \Phi + \Gamma K)^{-1}\Gamma\right]^{-1}_{z=1}$$

In observer-controller implementation, $\underline{u}(k) = K_r\,\underline{r}(k) - K\,\hat{\underline{x}}(k\,|\,k)$

- In steady-state, $\hat{\underline{x}}(k\,|\,k) \rightarrow \underline{x}(k)$ => K_r same as in SVFB case

- A common situation (consider SO case for simplicity):

$$y = x_j = \text{some position variable}$$

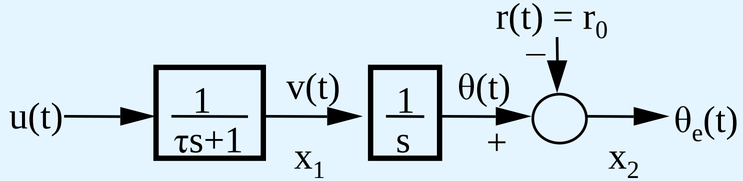
then generally $K_r = K_j$ = gain on x_j, i.e.,

$$u = -K_1\hat{x}_1 - \cdots - K_j(\hat{x}_j - r) - \cdots - K_n\hat{x}_n$$

Redefine $x_j - r \triangleq x_{je}$ = error in x_j and we measure only x_{je} => error model

- All previous results (with r = 0) applicable to error model

Example – Radar Positioning System



$$\dot{x}_1(t) = -\frac{1}{\tau}x_1(t) + \frac{1}{\tau}u(t)$$

$$\dot{x}_2(t) = x_1(t) \qquad \tau = 10 \text{ sec}$$

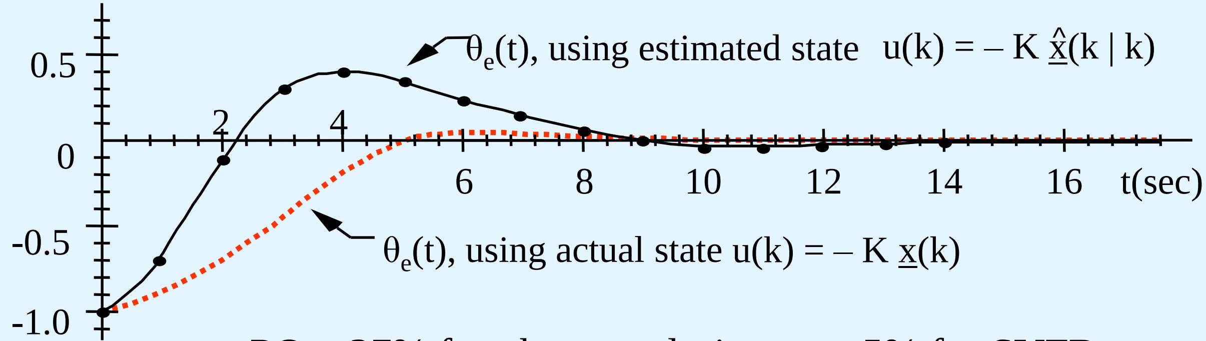
- Design digital control (h = 1) so that an initial offset $\theta_e(0) \rightarrow 0$ with $t_s|_{1\%}$ ~ 10 sec and PO ~ 10%. Only $\theta_e = x_2$ is measurable. $x_1(0) = 0$, $x_2(0) = -r_0$
  - Equivalent to an input command system when only the error $e(t) \triangleq \theta_e(t) = \theta(t) - r$ is measurable and $\theta(0) = 0$
- Digital control design $\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-0.1 & 0\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0.1\\ 0\end{bmatrix}u$
  - Equivalent discrete system with h = 1

$$\underline{x}(k+1) = \begin{bmatrix}0.905 & 0\\ 0.952 & 1\end{bmatrix}\underline{x}(k) + \begin{bmatrix}0.095\\ 0.048\end{bmatrix}u(k)$$

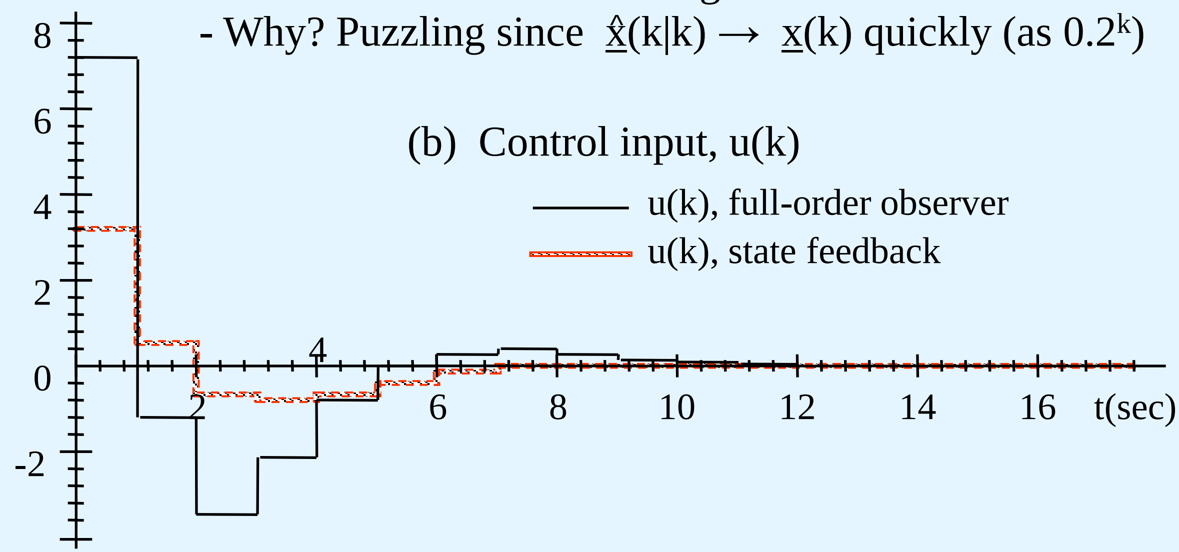
  - Pick desired CL poles $\lambda_i = e^{s_i h}$, $s_i = -0.5 \pm j0.5$; $p_d(z) = z^2 - 1.06z + 0.367$ $(\lambda_i = 0.53 \pm j0.29)$
  - SVFB gains K = [7.21 3.19]
- Only the shaft offset $\theta_e(k) = x_2(k)$ available for measurement
  - Design observer to estimate $\underline{x}(k)$
  - With C = [0 1], system is observable
  - Observer poles $\tilde{\lambda}_i \le |\lambda_{prim}|^3 = 0.6^3 \approx 0.22$; pick $\tilde{\lambda}_i = 0.2 \pm j0.1$ for good damping on $\tilde{e}$
  - Observer gain L = [0.588 0.945]′
- Simulate CL response: $\underline{x}(0) = [\,0 \;\; -1\,]'$; $\hat{\underline{x}}(0|-1) = [\,0 \;\; 0\,]'$; r = unit step

Simulation Results

(a) Shaft offset angle, $\theta_e(t)$, $\theta_e(kh)$



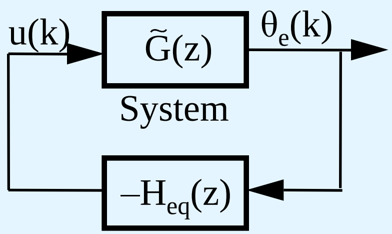
- PO ≈ 37% for observer design vs ≈ 5% for SVFB.
- Why? Puzzling since $\hat{\underline{x}}(k|k) \rightarrow \underline{x}(k)$ quickly (as 0.2^k)



- Observer design uses significantly more control than does SVFB, especially at k = 0
- After first measurement $y(0) = \theta_e(0) = -1$, examine observer:

$\hat{x}_1(0\,|\,0) = -0.588$, $\hat{x}_2(0\,|\,0) = -0.945$ ==> $-K\,\hat{\underline{x}}(0\,|\,0) = 7.25$

Alternate Representation in Output Feedback Form



- Open-loop dynamics

$$\tilde{G}(z) = C(zI - \Phi)^{-1}\Gamma = 0.048\,\frac{z + 0.967}{(z-1)(z-0.905)}$$

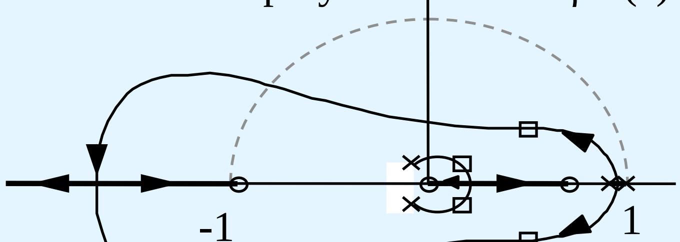
- Feedback loop dynamics

$$H_{eq}(z) = zK(zI - \tilde{\Phi})^{-1}L = \frac{\text{2nd order}}{\text{2nd order}}\ ;\ L = \begin{bmatrix} 0.588 \\ 0.945 \end{bmatrix};\ K = [\ 7.21 \quad 3.19\ ]$$

$$\tilde{\Phi} = (I - LC)(\Phi - \Gamma K) = \begin{bmatrix} -0.136 & -0.801 \\ 0.0333 & 0.0466 \end{bmatrix}$$

$$H_{eq}(z) = \frac{7.25z^2 - 5.18z}{z^2 + 0.0894z + 0.0204} \qquad \text{(zeros @ 0.713, 0 ; poles @ } -0.045 \pm j0.134)$$

- Can examine ϕ_m, ω_c via $LG_{ain}(z) = \tilde{G}(z)\,H_{eq}(z)$;
  - gives $\omega_c = 0.62$ rad/sec, $\phi_m = 40^o$
- Examine root locus of CL system with $H_{eq} \rightarrow \beta H_{eq}$
  CL characteristic polynomial = $1 + \beta\tilde{G}(z)\,H_{eq}(z)$



Root locus wr to β

- When $\beta = 1$, CL poles are at
  $z_i = 0.2 \pm j0.1$ (observer poles)
  and $0.53 \pm j0.29$ (controller poles)

Possible Modification to Improve Response

- Problems when $\underline{x}(0) \neq 0$, or if a sudden large $\Delta \underline{x}$ $y(0)$

  if $\hat{\underline{x}}(0 \mid -1) = \underline{0}$, $u(0) = -K\,\hat{\underline{x}}(0 \mid 0) = -KLC\,\underline{x}(0)$

  - Initial $u(0)$ can be far from $-K\,\underline{x}(0)$ in such cases since $\hat{\underline{x}}(0 \mid 0)$ is off.
- These problems are typical of command input systems when we only measure the error $e(k) = y(t) - r(t)$, i.e., is a change in e due to a change in $\underline{x}$ or a change in r ?

(1) Initialize $\hat{x}_2(0 \mid -1) = -r_0 = -1$

- Resulting $\theta_e(t) \equiv \theta_e(t)$ with SVFB (provided $x_1(0) = 0$)
- Plausible to do in an input command system when r(k) is known

  ($x_2 \sim \theta - r$ or $\theta \sim x_2 + r$, θ = shaft angle)

$$\hat{x}_2(k^+ \mid k-1) = x_2(k^- \mid k-1) - r(k) + r(k-1) \quad = 0 \ @\ k = 0$$

- Not possible in general if $\underline{x}(0)$ has unknown structure

(2) Slow down observer at $k = 0$

- Phase in observer gain $L(k)$: $0 \rightarrow L$
- Use slower observer poles, e.g., $\leq p^2 \Rightarrow \tilde{\lambda}_i = 0.3 \pm 0.2j$

  (results in smaller L, but slower CL response)

(3) Slow down the control $u(k) = \alpha\, u(k-1) - (1-\alpha) K\,\hat{\underline{x}}(k \mid k)$

- Best obtained by using $\Delta(k) = u(k) - u(k-1)$ as the "control"

  => a very popular scheme in practice

(4) Slow down the input command

- Use r(k) = a sequence of smaller changes => bound $|\Delta r(k)|$

(5) Reformulate as a command input problem; $u(k) = K_r r(k) - K\,\hat{\underline{x}}(k \mid k)$ where we measure $\theta(k)$

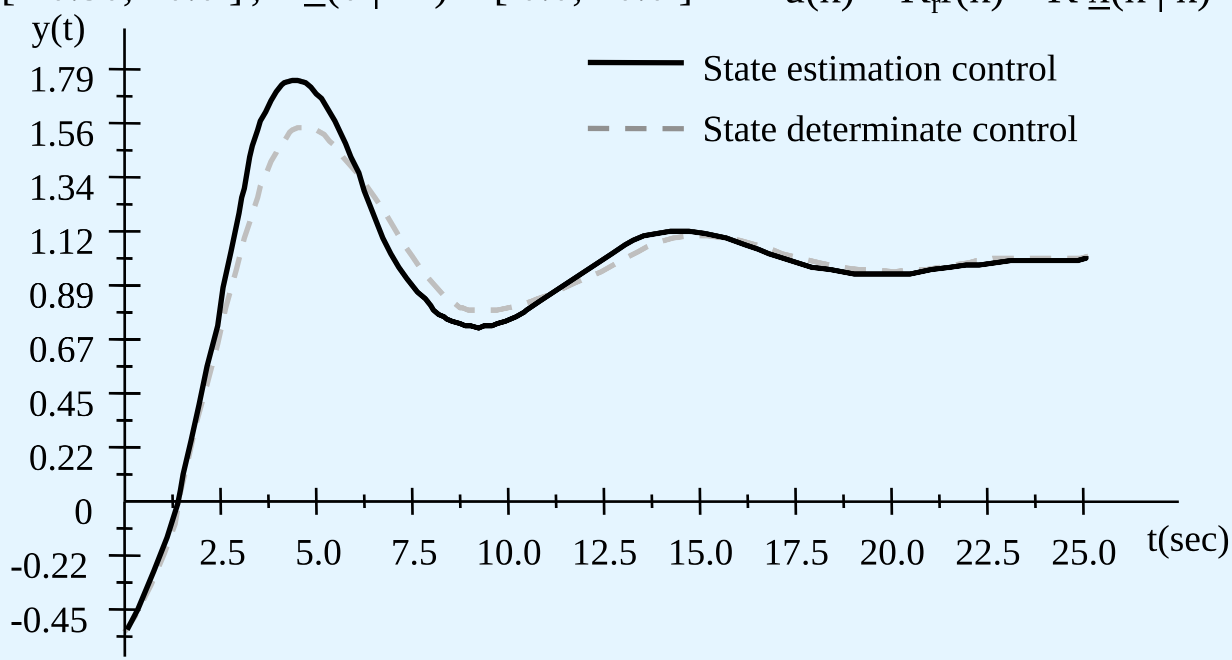
Example – Satellite Control with Command Input

- Formulation as a command input structure

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = u, \quad \text{measure } y(k) = x_1(k)$$

$$h = 0.5, \quad r(k) = 1 \text{ (unit step)}$$

- Select SVFB gains to place control poles at $z_i = 0.85 \pm j0.3$ (gives $\omega_c \approx 0.78$ rad/sec, $\phi_m \approx 32.5^o$)
- Select L to place observer poles at $z_i = 0.55 \pm j0.15$
- $\underline{x}(0) = [\,-0.50, \;\; 0.0\,]'$, $\quad \underline{\hat{x}}(0\,|\,-1) = [\,0.0, \;\; 0.0\,]'$ $\qquad u(k) = K_r r(k) - K\,\underline{\hat{x}}(k\,|\,k)$



- If $\underline{x}(0) = \underline{0}$, response of system using observer is identical to response of system using actual state.

Reduced-Order Observers

- Redefine states so that $y(k) = x_1(k)$
  - Use standard observable form, or
  - Use SV transformation $\underline{v} = T^{-1}\,\underline{x}$ with $T^{-1} = \begin{bmatrix} -\ C\ - \\ T_{n-1} \end{bmatrix} \}\ n\text{-}1$

    $T_{n-1} = (n-1)\cdot n$, arbitrary (need only $CT = \underline{e}_1'$)
- Idea: if measure $y(k) = x_1(k)$ need only to estimate $\underline{x}_b = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$

  $\Rightarrow$ need only build an $(n-1)^{st}$ order estimator

  [in general if p measurements $\Rightarrow$ $(n-p)^{th}$ order observer]
- Decompose state equation: $\underline{x}(k) = \Phi\,\underline{x}(k-1) + \Gamma u(k-1)$; $y(k) = x_1(k)$

$$\begin{array}{r} 1\ \{ \\ n\text{-}1\ \{ \end{array}\begin{bmatrix} x_1(k) \\ \underline{x}_b(k) \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{1b} \\ \underbrace{\Phi_{b1}}_{1} & \underbrace{\Phi_{bb}}_{n-1} \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ \underline{x}_b(k-1) \end{bmatrix} + \begin{bmatrix} \Gamma_1 \\ \Gamma_b \end{bmatrix} \underline{u}(k-1)$$

(1) $\underline{x}_b(k) = \Phi_{bb}\,\underline{x}_b(k-1) + \Phi_{b1}y(k-1) + \Gamma_b\underline{u}(k-1)$ $\rightarrow$ $\underline{u}^*(k-1)$, known at time $k-1$

(2) $y(k) = \Phi_{11}y(k-1) + \Phi_{1b}\,\underline{x}_b(k-1) + \Gamma_1\underline{u}(k-1)$

$$\underbrace{y(k) - \Phi_{11}y(k-1) - \Gamma_1 u(k-1)}_{y^*(k),\ \text{known at time } k} = \Phi_{1b}\,\underline{x}_b(k-1)$$

- Build an observer for $\underline{x}_b$: $\underline{x}_b(k) = \Phi_{bb}\,\underline{x}_b(k-1) + \underline{u}^*(k-1)$

  $y^*(k) = \Phi_{1b}\,\underline{x}_b(k-1)$
- Observable? - If original $\{\Phi, C\}$ is observable, then $\{\Phi_{bb}, \Phi_{1b}\}$ is also

Reduced-Order Observer Design for $\underline{x}_b$

$\hat{\underline{x}}_b(k \mid k-1)$ = estimate of $\underline{x}_b$ based on $\{y(k-1), \cdots\} \triangleq \hat{\underline{x}}_b^-(k)$

$\hat{\underline{x}}_b(k \mid k)$ = estimate of $\underline{x}_b$ based on $\{y(k), y(k-1), \cdots\} \triangleq \hat{\underline{x}}_b^+(k)$

- Propagation/prediction step $k-1 \rightarrow k$

  $$\Rightarrow \quad \hat{\underline{x}}_b(k \mid k-1) = \Phi_{bb}\, \hat{\underline{x}}_b(k-1 \mid k-1) + \Phi_{b1} y(k-1) + \Gamma_b \underline{u}(k-1)$$

- Update step $\left.\begin{array}{c} \hat{\underline{x}}_b(k \mid k-1) \\ y^*(k) \end{array}\right\} \rightarrow \hat{\underline{x}}_b(k \mid k)$
  - follow same basic approach as full-order case

  $$\hat{\underline{x}}_b(k \mid k) = \hat{\underline{x}}_b(k \mid k-1) + L_b[y^*(k) - \Phi_{1b}\, \hat{\underline{x}}_b(k-1 \mid k-1)]$$

  best prediction of $y^*(k) = \Phi_{1b}\, \underline{x}_b(k-1)$ at time k, given $\{y(k-1), \cdots\}$

  $$\Rightarrow \quad \hat{\underline{x}}_b(k \mid k) = \hat{\underline{x}}_b(k \mid k-1) + L_b[y(k) - \underbrace{\Phi_{11} y(k-1) - \Gamma_1 \underline{u}(k-1) - \Phi_{1b}\, \hat{\underline{x}}_b(k-1 \mid k-1)}_{\hat{y}(k)}]$$

- Selection of $L_b = (n-1)$ gain vector
  - Can obtain equation for $\tilde{e}(k \mid k) \triangleq \underline{x}_b(k) - \hat{\underline{x}}_b(k \mid k)$:

    $$\underline{\tilde{e}}(k \mid k) = (\Phi_{bb} - L_b \Phi_{1b})\, \underline{\tilde{e}}(k-1 \mid k-1)$$

    Inverse exists if $\{\Phi_{bb}, \Phi_{1b}\}$ = observable

  - Pick L_b so that eigenvalues of $\Phi_{bb} - L_b \Phi_{1b}$ are in unit circle

    $$p_{er}(z) = z^{n-1} + \tilde{d}_1 z^{n-2} + \cdots + \tilde{d}_{n-1}$$

  - Analogous to earlier result with $\boxed{\Phi \rightarrow \Phi_{bb},\ C\Phi \rightarrow \Phi_{1b}}$: $L_b = p_{er}(\Phi_{bb}) \overbrace{\begin{bmatrix} \text{---} & \Phi_{1b} & \text{---} \\ \text{--} & \Phi_{1b}\Phi_{bb} & \text{--} \\ & \vdots & \\ \text{-} & \Phi_{1b}\Phi_{bb}^{n-2} & \text{-} \end{bmatrix}}^{}{}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$

    $H_{0,r}'$ = observability matrix for R-O system

Compute L_b by using **Place** command $\Phi \rightarrow \Phi_{bb}'$, $\Gamma \rightarrow \Phi_{1b}'$; obtain $K \rightarrow L_b'$

Implementation of Reduced-Order Observer/Controller

- Note, if original system in standard observable form

$$\Phi = \left[\begin{array}{c|cccc} -a_1 & 1 & 0 & \cdots & 0 \\ \hline -a_2 & 0 & 1 & & \\ \vdots & \vdots & & \ddots & \\ \vdots & \vdots & & & 1 \\ -a_n & 0 & \cdot & \cdot \; \cdot & 0 \end{array}\right] \Longrightarrow \quad H_{0,r}' = I_{(n-1)\cdot(n-1)}$$

Φ_{1b}

Φ_{bb}

- Algorithm at any particular k
  - Initialization: at step k = 0 we do not have $y(-1)$, $\underline{u}(-1)$ or $\hat{\underline{x}}_b(-1|-1)$ => $\hat{y} = 0$
  - Let $\hat{\underline{x}}_b(0|-1)$ = best prior estimate of $\hat{\underline{x}}_b(0)$, usually = $\underline{0}$

0 Initialize at k = 0: $\hat{\underline{x}}_b^-$, $\hat{y} = 0$

1 Sample: $y(k)$, $\underline{r}(k)$

2 $\hat{\underline{x}}_b^+ = \hat{\underline{x}}_b^- + L_b[y(k) - \hat{y}]$ — Update: $\hat{\underline{x}}_b(k|k)$

3 $\underline{u} = K_r\, r(k) - K \begin{bmatrix} y(k) \\ \hat{\underline{x}}_b^+ \end{bmatrix}$ — Control, using most current information

4 Return $\underline{u}(k)$

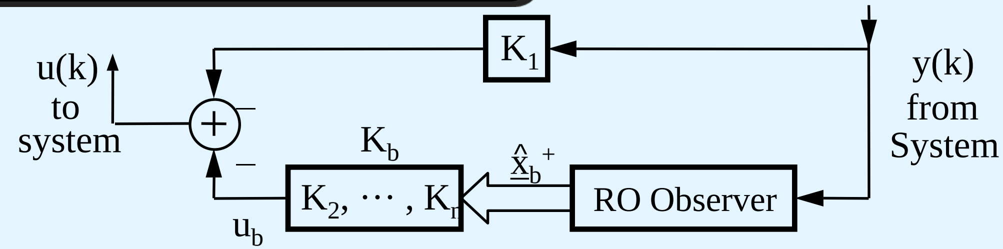
5 $\hat{y} = \Phi_{11} y(k) + \Gamma_1 u(k) + \Phi_{1b}\hat{\underline{x}}_b^+$ — Get $\hat{y}$ for next cycle

6 $\hat{\underline{x}}_b^- = \Phi_{bb}\hat{\underline{x}}_b^+ + \Phi_{b1} y(k) + \Gamma_b \underline{u}(k)$ → Return

Propagate $\hat{\underline{x}}_b(k+1|k)$ for next cycle

Loop Gain Analysis of R-O Observer/Controller

- Assume $\hat{\underline{x}}_b^-(0) = 0$, $r(k) = 0$
  => $u(k) = -K_1 y(k) - K_b \hat{\underline{x}}_b(k \mid k)$



- Obtain transfer function from y to $u = -H_{eq}(z)$
  - (n–1)st order feedback loop $u(z) = -H_{eq}(z)y(z) = -\left[K_1 + \dfrac{u_b(z)}{y(z)}\right]y(z)$
- Obtain dynamic equation for $\hat{\underline{x}}_b(k \mid k)$

$$\hat{\underline{x}}_b(k \mid k) = \hat{\underline{x}}_b(k \mid k-1) + L_b[y(k) - \Phi_{11}y(k-1) - \Gamma_1 u(k-1) - \Phi_{1b}\,\hat{\underline{x}}_b(k-1 \mid k-1)]$$

$$\Phi_{bb}\,\hat{\underline{x}}_b(k-1 \mid k-1) + \Phi_{b1}y(k-1) + \Gamma_b u(k-1)$$

$$\Rightarrow \hat{\underline{x}}_b(k \mid k) = \underbrace{[\Phi_{bb} - L_b\Phi_{1b} - (\Gamma_b - L_b\Gamma_1)K_b]}_{\tilde{\Phi}_{bb}}\,\hat{\underline{x}}_b(k-1 \mid k-1) + L_b y(k) + \underbrace{[\Phi_{b1} - L_b\Phi_{11} - K_1(\Gamma_b - L_b\Gamma_1)]}_{\tilde{L}_b}y(k-1)$$

  - $u(k) = K_b\,\hat{\underline{x}}_b(k \mid k) + K_1 y(k)$

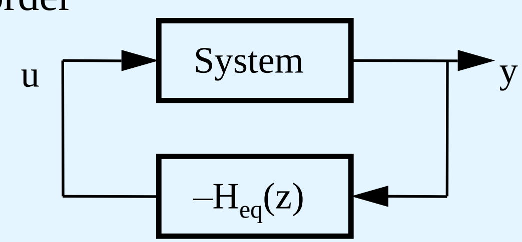
- Take z-transforms

$$H_{eq}(z) = [K_b(zI - \tilde{\Phi}_{bb})^{-1}(zL_b + \tilde{L}_b) + K_1] = \frac{(n-1)\text{ order}}{(n-1)\text{ order}}$$

$$\tilde{\Phi}_{bb} = \Phi_{bb} - L_b\Phi_{1b} - (\Gamma_b - L_b K_1)K_b$$

$$\tilde{L}_b = \Phi_{b1} - L_b\Phi_{11} - K_1(\Gamma_b - L_b\Gamma_1)$$

If measure p outputs, will obtain an (n–p)th order feedback compensator.



- Treat command inputs similarly to full-order observer case.

Example – Radar Positioning Problem

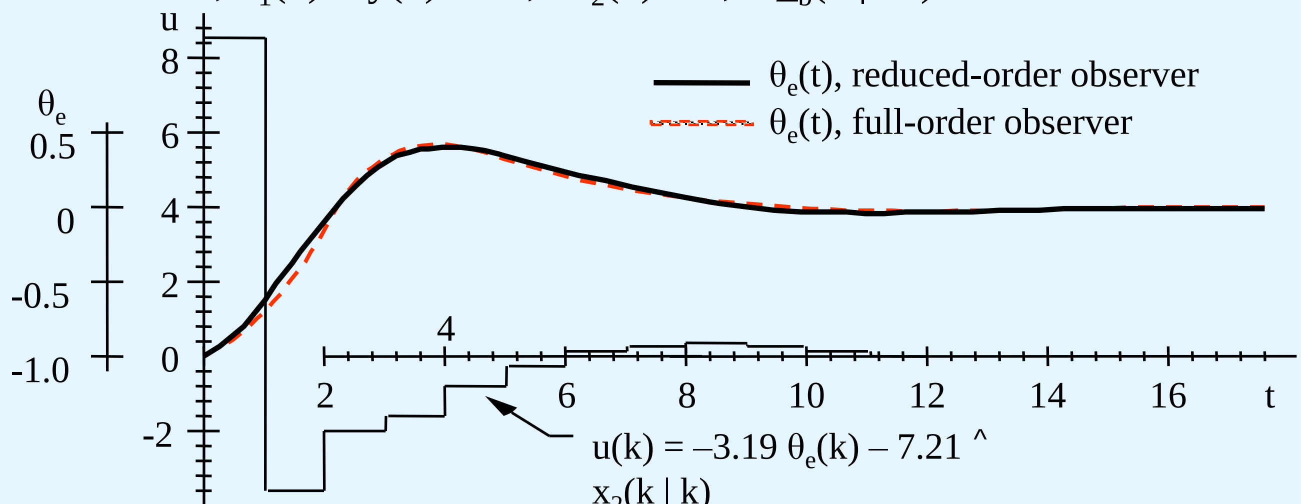
$$\underline{x}(k+1) = \begin{bmatrix} 0.905 & 0 \\ 0.952 & 1 \end{bmatrix} \underline{x}(k) + \begin{bmatrix} 0.095 \\ 0.048 \end{bmatrix} u(k)\,; \quad y(k) = x_2(k) = \theta_e(k)$$

- Redefine states so that $x_1 = y(k) \;\Rightarrow\; T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$; $T^{-1} = T$

$$\underline{x}(k+1) = \underbrace{\left[\begin{array}{c:c} 1 & 0.952 \\ \hdashline 0 & 0.905 \end{array}\right]}_{T^{-1}\Phi T} \underline{x}(k) + \underbrace{\left[\begin{array}{c} 0.048 \\ \hdashline 0.095 \end{array}\right]}_{T^{-1}\Gamma} u(k)\,; \quad y(k) = x_1(k) = \theta_e(k)$$

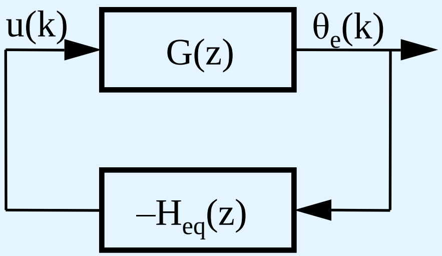
K = [3.19 7.21] from previous analysis

- Select observer gain L_b ($\Phi_{bb} = 0.905$, $\Phi_{1b} = 0.952$)
  - Place observer pole at $\tilde{\lambda}_i = 0.2 \;\Rightarrow\; \Phi_{bb} - L_b\Phi_{1b} = 0.2 = 0.905 - L_b\, 0.952 \;\Rightarrow\; \underline{L_b = 0.740}$
- Simulation results, $x_1(0) = y(0) = -1$, $x_2(0) = 0$; $\hat{\underline{x}}_b(0\,|\,-1) = 0$



  - Results highly analogous to full-order observer case
    $\hat{x}_2(0\,|\,0) = 0 + L_b\{y(0) - 0\} = -0.74 \;\Rightarrow\; u(0) = 8.52$

Alternate Implementation and LG Analysis



$$\underline{x}(k+1) = \begin{array}{cc} \Phi_{11} & \Phi_{1b} \\ \left[\begin{array}{c|c} 1 & 0.952 \\ \hline 0 & 0.905 \end{array}\right] & \\ \Phi_{b1} & \Phi_{bb} \end{array} x(k) + \begin{array}{c} \Gamma_1 \\ \left[\begin{array}{c} 0.048 \\ \hline 0.095 \end{array}\right] \\ \Gamma_b \end{array} u(k)$$

$$y(k) = x_1(k)$$

- Feedback loop dynamics (reduced-order observer, $L_b = 0.74$)

  $H_{eq}(z) = [K_b(zI - \tilde{\Phi}_{bb})^{-1}(zL_b + \tilde{L}_b) + K_1]$; $K_1 = 3.19$; $K_b = K_2 = 7.21$

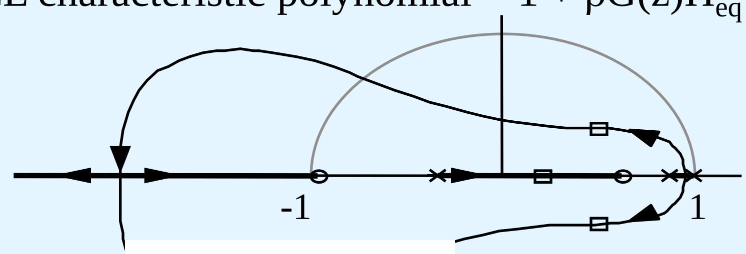
  $\tilde{\Phi}_{bb} = [\Phi_{bb} - L_b\Phi_{1b} - (\Gamma_b - L_b\Gamma_1)K_b] = -0.36$

  $\tilde{L}_b = \Phi_{b1} - L_b\Phi_{11} - K_1(\Gamma_b - L_b\Gamma_1) = -0.93$

  $\Rightarrow H_{eq}(z) = \dfrac{8.52z - 5.55}{z + 0.362} = 8.52\dfrac{z - 0.650}{z + 0.362}$

- Examine root locus of CL system with $H_{eq} \to \beta H_{eq}$

  CL characteristic polynomial = $1 + \beta\tilde{G}(z)H_{eq}(z)$; $\tilde{G}(z) = \dfrac{0.048(z+0.967)}{(z-1)(z-0.905)}$



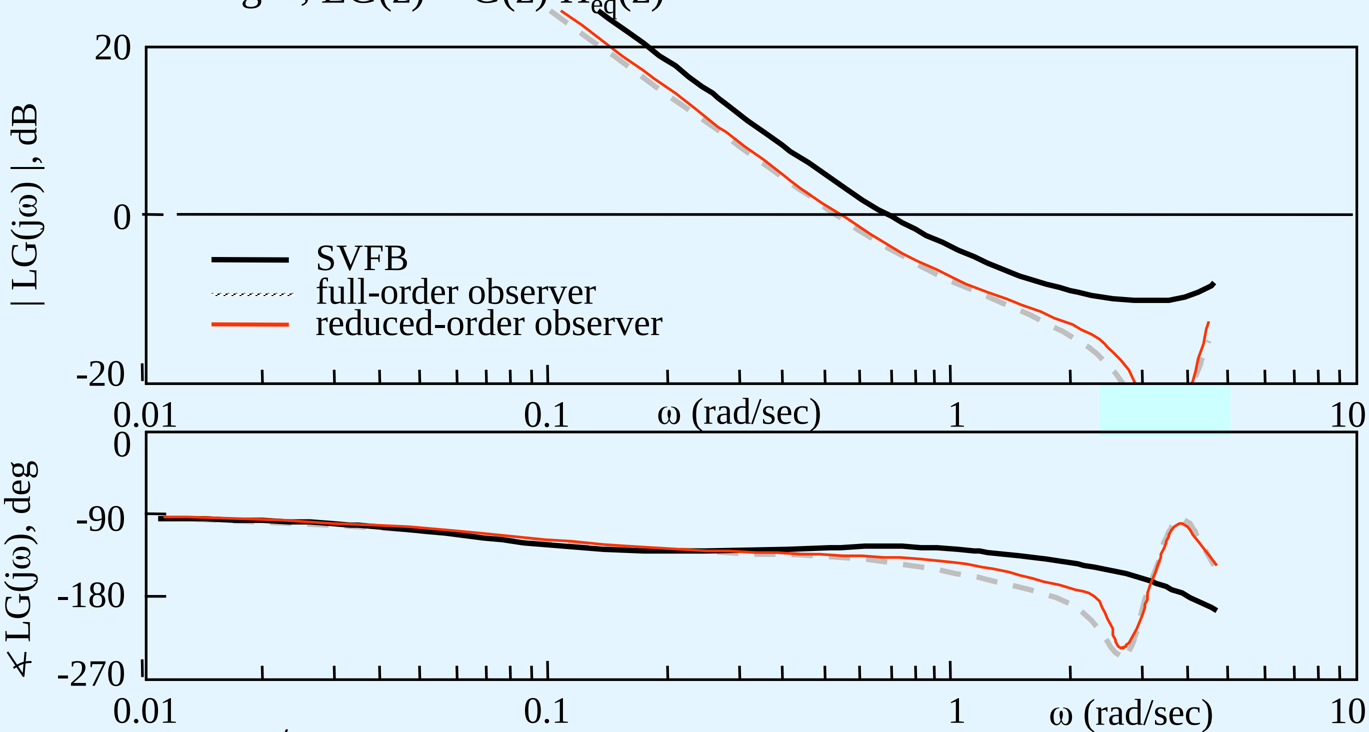
Root locus wr to β

- When $\beta = 1$, CL poles are at
  $z_i = 0.53 \pm j0.29$
  and 0.2

- In general, for RO observer-controller
  - CL poles are observer poles plus controller poles
  - $T(z) = \dfrac{y(z)}{r(z)} = \dfrac{K_r B(z) p_{er}(z)}{p_d(z) p_{er}(z)} = \dfrac{K_r B(z)}{p_d(z)}$ $\quad B(z)$ = Numerator of open-loop $\tilde{G}(z)$

Phase Margin and Loop Gain Comparisons; Radar Positioning

- For SVFB, $LG(z) = K(zI - \Phi)^{-1}\Gamma = \dfrac{0.84z - 0.573}{z^2 - 1.905z + 0.905}$
- For observer designs, $LG(z) = \tilde{G}(z)\, H_{eq}(z)$



- SVFB: $\omega_c = 0.83$, $\phi_m = 46^o$
- Full-order observer: $\omega_c = 0.616$, $\phi_m = 40^o$
- Reduced-order observer: $\omega_c = 0.614$, $\phi_m = 39.3^o$

- To compensate for the phase-shift (filtering) that they introduce, observer designs inherently reduce ω_c.

Summary of Observer Design

- Full-order observer offers an excellent method to estimate system states from output measurement(s)
  - Can specify how fast $\underline{\tilde{e}}(k \mid k) \rightarrow \underline{0}$ via $\tilde{\lambda}_i$
  - Faster estimation => higher gains L and more sensitivity to errors
  - Calculate estimator gains via use of **Place** or **Acker** (SISO) commnds
- Can obtain estimate of $\underline{x}$ between samples,

  $$\hat{\underline{x}}(kh+\delta) = \Phi(\delta)\,\hat{\underline{x}}(k \mid k) + \Gamma(\delta)u(k) \quad \text{where } \Phi(\delta) = e^{A\delta}\ ; \quad \Gamma(\delta) = \int_0^{\delta} e^{A\sigma} d\sigma B$$

- Use of $\hat{\underline{x}}(k \mid k)$ in place of $\underline{x}(k)$ in feedback
  - Need to place observer poles closer to origin ($z = 0$) than primary control poles $r = p^2$ to p^3; $p = |\lambda_{dom}|$ = magnitude of CL poles
- Implementation
  - Requires additional computation/storage
  - Includes a "model" of system in its structure
  - Can be implemented as an n-th order FB compensator (when r = 0)
- Reduced-order observer
  - Can implement an (n–1)-order observer when $x_1 = y$ by setting $\hat{x}_1 \triangleq y$
- Poles of CL observer/controller = $\{\lambda_i, \cdots, \lambda_n, \ \tilde{\lambda}_1, \cdots, \tilde{\lambda}_n\}$
- CL transfer function from r to y same as SVFB using actual states
  - Observer is "transparent" in steady state
- Observer: excellent for systems that have good quality measurements, and state is subject to occasional random/deterministic changes, Δx.

Modification for Time Delay $\tau = Mh + \varepsilon$

$$\underline{x}(k+1) = \Phi\,\underline{x}(k) + \Gamma_0 u(k-M) + \Gamma_1 u(k-1-M)$$

- Control law modifications using τ-sec ahead prediction:

$$u(k) = K_r\, r(k) - K\,\hat{\underline{x}}(kh+\tau \mid k)$$

prediction of state at time $t + Mh + \varepsilon$
from $\hat{\underline{x}}(k \mid k)$ and $u(k-1), \cdots, u(k-M), u(k-M-1)$

(1) Obtain $\hat{\underline{x}}(kh+\varepsilon) = e^{A\varepsilon}\,\hat{\underline{x}}(k \mid k) + \int_0^{\varepsilon} e^{A\sigma} d\sigma B u(k-M-1)$

(2) M-step propagation from time $kh + \varepsilon$ to $kh + \varepsilon + Mh$

$$\hat{\underline{x}}(kh+\tau) = \Phi^M\,\hat{\underline{x}}(kh+\varepsilon) + \sum_{i=1}^{M} \Phi^{i-1}\Gamma u(k-i)$$

- Store $\{u(k-M-1), u(k-M), \cdots, u(k-1)\}$ in an M+1 stack
- Controller is exactly as for full state FB, but with $\underline{x}(k) \rightarrow \hat{\underline{x}}(k \mid k)$

- Observer modifications - Propagate step only

$$\hat{\underline{x}}(k+1 \mid k) = \Phi\,\hat{\underline{x}}(k \mid k) + \Gamma_0 u(k-M) + \Gamma_1 u(k-1-M)$$

= prediction of state at next sample time

- Since initial estimates are incorrect, estimation error will be propagated forward in time

=> future FB control may not be very good until $\hat{\underline{x}}(\cdot) \rightarrow \underline{x}(\cdot)$.

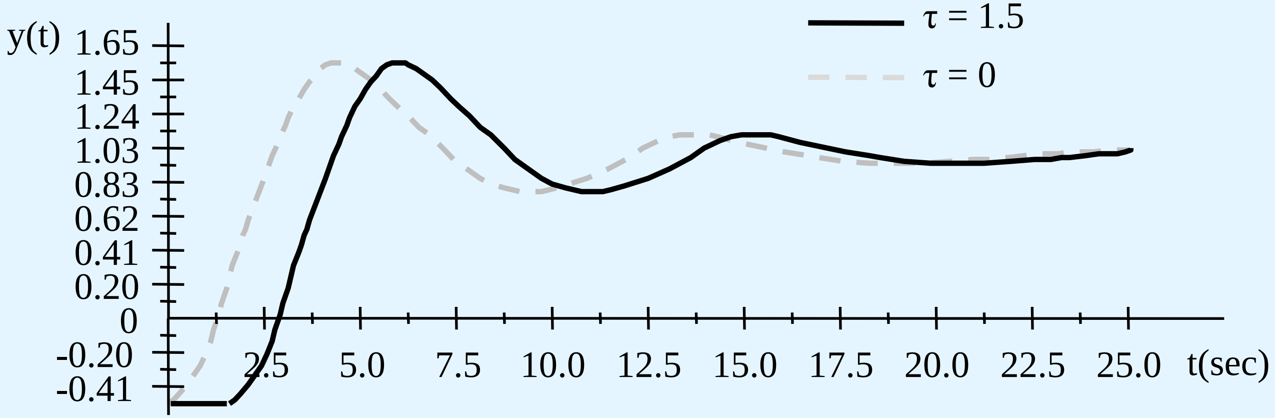
- Response $\neq$ time shifted response with initial $\underline{x} = e^{A\tau}\,\underline{x}(0)$

Example – Satellite Control with Time Delay

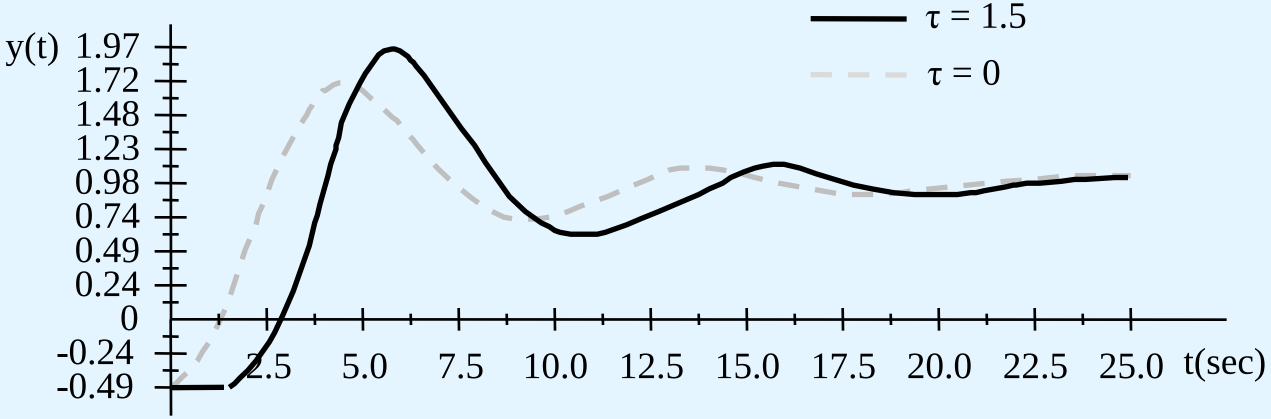
- Same problem as examined previously but with M = 3 step delay in loop (τ = 1.5 sec)

$$\underline{x}(0) = [\ -0.50 \quad 0.0\ ]', \quad u(k) = K_r r(k) - K\,\hat{\underline{x}}(k+M \mid k)$$

- Full state FB, with and without delay

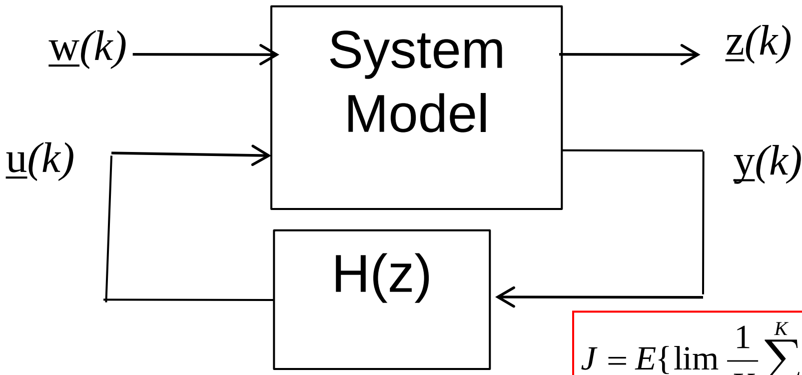


- State estimation FB, with and without delay



H_2 Controller = Generlized LQG

- Standard Controller Design Framework. H_2 assumes disturbances to be white noise procsses.



$\underline{x}(k+1) = \Phi\,\underline{x}(k) + \Gamma\underline{u}(k) + E\,\underline{w}(k)$
$\underline{z}(k) = C_1\,\underline{x}(k) + D_1\,\underline{u}(k)$
$\underline{y}(k) = C\,\underline{x}(k) + \underline{v}(k)$
$\mathrm{Cov}[\underline{w}(k)] = W$, $\mathrm{Cov}[\underline{v}(k)] = V$
$\mathrm{Cov}[\underline{w}(k), \underline{v}(k)] = N$

$$J = E\{\lim_{K\to\infty}\frac{1}{K}\sum_{k=0}^{K}\underline{z}^T(k)\underline{z}(k)\} = E\{\underline{x}^T(k)Q\,\underline{x}(k) + 2\underline{x}^T(k)M\,\underline{u}(k) + \underline{u}^T(k)R\underline{u}(k)\}$$

where $Q = C_1^T C_1; M = C_1^T D_1; R = D_1^T D_1$

- H_2 controller = Generalized LQG $\Rightarrow$ can select W, V, N, Q, M and R arbitrarily. Also, can employ frequency weighting of cost terms as in LQR

If $\underline{w}(k)$ and $\underline{v}$ (k-1) are Correlated, formulas change . See Bar-Shalom's book Ch. 8, section 3, Page 326

- Kalman Filter: Define $\bar{\Phi} = \Phi - ENV^{-1}C; \bar{W} = W - NV^{-1}N^T$

  - "Prediction" $\Rightarrow \hat{\underline{x}}(k|k-1) = \bar{\Phi}\hat{\underline{x}}(k-1|k-1) + \Gamma\underline{u}(k-1) + ENV^{-1}\underline{y}(k-1)$

  - "Update" => $\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + L[\underline{y}(k) - C\hat{\underline{x}}(k|k-1)]$

    L = n x m Kalman gain matrix = $\Sigma C^T (C\Sigma C^T + V)^{-1}$

  - Σ is the steady state prediction error covariance matrix given by

$$\Sigma = \bar{\Phi}\Sigma\bar{\Phi}^T + E\bar{W}E^T - \bar{\Phi}\Sigma C^T(C\Sigma C^T + V)^{-1}C\Sigma\bar{\Phi}^T$$

H_2 Controller = Generlized LQG

- Controller

$$\underline{u}(k) = -(R + \Gamma^T \tilde{P}^* \Gamma)^{-1} \Gamma^T \tilde{P}^* \Phi \hat{\underline{x}}(k|k-1) - R^{-1} M^T \hat{\underline{x}}(k|k-1)$$

where $\tilde{P}^*$ satisfies the DARE

$$\tilde{P}^* = \tilde{\Phi}^T [\tilde{P}^* - \tilde{P}^* \Gamma (R + \Gamma^T \tilde{P}^* \Gamma)^{-1} \Gamma^T \tilde{P}^*] \tilde{\Phi} + \tilde{Q}$$

$$\tilde{\Phi} = \Phi - \Gamma R^{-1} M^T ; \tilde{Q} = Q - M R^{-1} M^T \geq 0$$

- For sampled data systems, there is a technique called *lifting* that takes into account intra-sample behavior of the continuous system. Fairly complicated process. Suggest that you design in continuous domain and use **Tustin** or **average gain** method.

$$\dot{\underline{x}} = A\underline{x}(t) + B\underline{u}(t) + E\underline{w}(t)$$

$$\underline{z}(t) = C_1 \underline{x}(t) + D_1 \underline{u}(t)$$

$$\underline{y}(t) = C\underline{x}(t) + D\underline{u}(t) + \underline{v}(t)$$

$\underline{x}$ ~ state

$\underline{z}$ ~ defines cost function

$\underline{y}$ ~ measurements (outputs)

Assumptions:

(i) $[A\ B]$ controllable (or stabilizable)

(ii) $[A\ C]$ observable (or detectable)

(iii) $\text{Cov}[E\underline{w}(t)\ \underline{v}(t)] = \begin{bmatrix} EWE^T & EN \\ N^T E^T & V \end{bmatrix}$

(iv) $Q = C_1^T C_1 ; M = C_1^T D_1 ; R = D_1^T D_1 > 0 \ \& \ Q - M R^{-1} M^T \geq 0$

H_2 controller minimizes the 2-norm of $T_{zw}(s)$

What is $T_{zw}(s)$?

Continuous H_2 Controller

- What is $T_{zw}(s)$?

$$observer-controller\ gives: \underline{u}(s) = -K\hat{\underline{x}}(s) = -H(s)\underline{y}(s)$$

$$\underline{z}(s) = [C_1(sI-A)^{-1}B + D_1]\underline{u}(s) + C_1(sI-A)^{-1}E\underline{w}(s) = G_1(s)\underline{u}(s) + G_2(s)\underline{w}(s)$$

$$\underline{y}(s) = [C(sI-A)^{-1}B + D]\underline{u}(s) + C(sI-A)^{-1}E\underline{w}(s) + \underline{v}(s) = G(s)\underline{u}(s) + G_3(s)\underline{w}(s) + \underline{v}(s)$$

$$\Rightarrow \underline{y}(s) = [I_p + G(s)H(s)]^{-1}[G_3(s)\underline{w}(s) + \underline{v}(s)]$$

$$\underline{u}(s) = -H(s)G(s)\underline{u}(s) - H(s)G_3(s)\underline{w}(s) - H(s)\underline{v}(s)$$

$$\Rightarrow \underline{u}(s) = -[I_m + H(s)G(s)]^{-1}[H(s)G_3(s)\underline{w}(s) + H(s)\underline{v}(s)]$$

$$T_{zw}(s)|_{\underline{v}=0} = G_2(s) - G_1(s)[I_m + H(s)G(s)]^{-1}H(s)G_3(s)$$

You can shape disturbance to minimize sensitivity
and shape measurement noise to minimize effects of noise and model uncertainties
at high frequencies

- What is H(s) ?

$$\dot{\hat{\underline{x}}} = A\hat{\underline{x}} + B\underline{u} + L(\underline{y} - C\hat{\underline{x}} - D\underline{u}) = (A - BK - LC + LDK)\hat{\underline{x}} + L\underline{y}$$

$$\underline{u}(s) = -K\hat{\underline{x}}(s) = -K(sI - A + BK + LC - LDK)^{-1}L\underline{y}(s)$$

- How to get the gains *K* and *L*? Via control and estimation Riccati equations

$$K = R^{-1}B^TP + R^{-1}M^T;\ \tilde{A} = A - BR^{-1}M^T; \tilde{Q} = Q - MR^{-1}M^T$$

$$L = (\Sigma C^T + EN)V^{-1}; \overline{A} = A - ENV^{-1}C; \overline{W} = W - NV^{-1}N^T$$

Control CARE: $\tilde{P}\tilde{A} + \tilde{A}^T\tilde{P} + \tilde{Q} - \tilde{P}BR^{-1}B^T\tilde{P} = 0$

Estimation CARE: $\overline{A}\Sigma + \Sigma\overline{A}^T + E\overline{W}E^T - \Sigma C^TV^{-1}C\Sigma = 0$

Application to F8 Aircraft

- H_2 Design for F8 Aircraft (Lublin, Grocott and Athans, Chapter 40 of Control Handbook)

a) *continuous system model*

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1.5 & -1.5 & 0 & 0.0057 & 1.5 \\ -12 & 12 & -0.6 & -0.0344 & -12 \\ -0.852 & 0.290 & 0 & -0.014 & -0.29 \\ 0 & 0 & 0 & 0 & -0.730 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 & 0 \\ 0.16 & 0.80 \\ -19 & -3 \\ -0.015 & -0.0087 \\ 0 & 0 \end{bmatrix} \underline{u} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1.1459 \end{bmatrix} d; \underline{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \underline{x} + \underline{v}(t)$$

Specs : $V = 0.01 \deg^2/\sec$; Magnitude of each output to be less than 0.25 degrees for $\omega \le 1$rad/sec for d.

Recall $\underline{y}(s) = S(s)[G_d(s)\underline{d}(s) + \underline{v}(s)] \Rightarrow want\ \sigma_{\max}[S(j\omega)G_d(j\omega)] < 0.25 \approx \dfrac{1}{|w_1(j\omega)|}$ for $\omega \le 1$rad/sec

Also, want $\sigma_{\max}[T(j\omega)] = \sigma_{\max}[S(j\omega)G(j\omega)H(j\omega)] < \dfrac{1}{|e_m(j\omega)|} = \dfrac{10}{\omega} \approx \dfrac{1}{\sqrt{V}\,|w_2(j\omega)|}\ \forall \omega$

b) one loop shaping design: select $w_1(s) = \dfrac{0.1(s+100)}{(s+1.25)}$. This keeps $|w_1(j\omega)| > 15.8dB$ for $0 \le \omega \le 1$ rad/sec.

inverse of $w_1(s)$ is a lead network $\Rightarrow$ small at low frequencies $\Rightarrow$ rejects disturbances.

$\Rightarrow$ Pass unit intensity white noise through $w_1(s)$ to get d

select $\sqrt{V} w_2(s) = \dfrac{500(s+3.5)}{3.5(s+1000)} \approx 0.5s/3.5$ for $\omega > 5$. $\Rightarrow$ Pass unit intensity white noise through $\sqrt{V} w_2(s)$ to get $\underline{v}(t)$

Also, select control weight $R = 0.01$

Augmented system will have 8 states (5+1+2).

Application to F8 Aircraft

- Controller gains

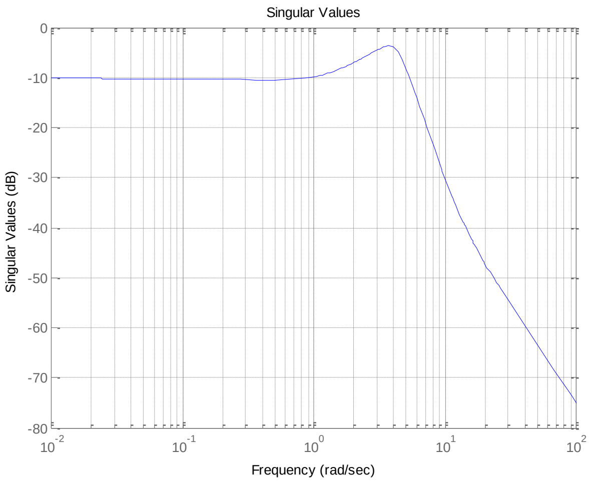
$$K = [-9.6256 \quad -2.0029 \quad -0.9889 \quad -0.0002 \quad 0.1241 \quad -0.0503 \quad 0.5181 \quad -4.3297$$
$$-0.1784 \quad 8.2263 \quad -0.0025 \quad 0.0056 \quad 1.3109 \quad 0.1561 \quad 0.0818 \quad -22.0555]$$

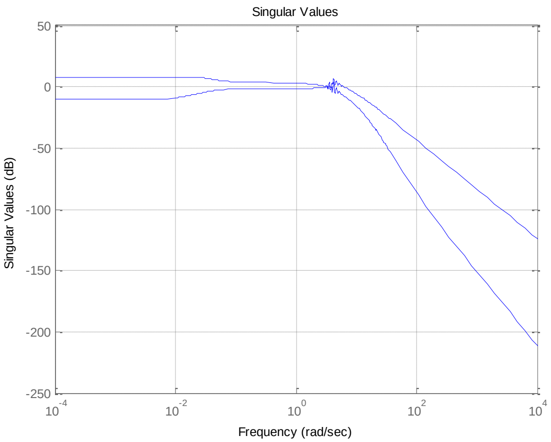
- Transpose of Observer gains

$$L^T = [\ 0.0224 \quad -0.0051 \quad 0.0279 \quad -0.0025 \quad -0.0242 \quad -0.0213 \quad 3.5696 \quad 0.0044$$
$$-0.0101 \quad 0.0041 \quad -0.0260 \quad 0.0008 \quad 0.0145 \quad 0.0175 \quad 0.0044 \quad 3.5802]$$

MATLAB routines:
- care
- lqg
- h2lqg
- h2syn

- Disturbance rejection for ω≤ 1 rad/sec and sigma plot of closed-loop transfer function





- Discrete gains via Tustin transformation of H(s) or via average gain method
- Best to simulate as a 2n-dimensional (16 in this case) system

Mixed Sensitivity Loop Shaping

- Controller transfer function H(s). Use Tustin to transform it to H(z)

Transfer function from output 1 to input...

$$\#1: \frac{1.595\,s^7 - 229.5\,s^6 - 3500\,s^5 - 3.072e004\,s^4 - 9.832e004\,s^3 - 2.611e005\,s^2 - 3.284e005s - 4389}{s^8 + 41.54\,s^7 + 889.3\,s^6 + 1.093e004\,s^5 + 7.88e004\,s^4 + 3.106e005\,s^3 + 4.363e005\,s^2 + 2.049e005\,s + 2733}$$

$$\#2: \frac{0.114\,s^7 - 177\,s^6 - 4350\,s^5 - 5.487e004\,s^4 - 3.251e005\,s^3 - 1.248e006\,s^2 - 1.706e006\,s - 2.229e004}{s^8 + 41.54\,s^7 + 889.3\,s^6 + 1.093e004\,s^5 + 7.88e004\,s^4 + 3.106e005\,s^3 + 4.363e005\,s^2 + 2.049e005\,s + 2733}$$

Transfer function from output 2 to input...

$$\#1: \frac{-15.38\,s^7 - 57.2\,s^6 + 2723\,s^5 + 2.061e004\,s^4 - 1105\,s^3 - 3.722e004\,s^2 - 813\,s + 5.344}{s^8 + 41.54\,s^7 + 889.3\,s^6 + 1.093e004\,s^5 + 7.88e004\,s^4 + 3.106e005\,s^3 + 4.363e005\,s^2 + 2.049e005\,s + 2733}$$

$$\#2: \frac{-78.9\,s^7 - 2307\,s^6 - 3.449e004\,s^5 - 2.453e005\,s^4 - 5.87e005\,s^3 - 4.829e005\,s^2 - 3040\,s + 166.5}{s^8 + 41.54\,s^7 + 889.3\,s^6 + 1.093e004\,s^5 + 7.88e004\,s^4 + 3.106e005\,s^3 + 4.363e005\,s^2 + 2.049e005\,s + 2733}$$

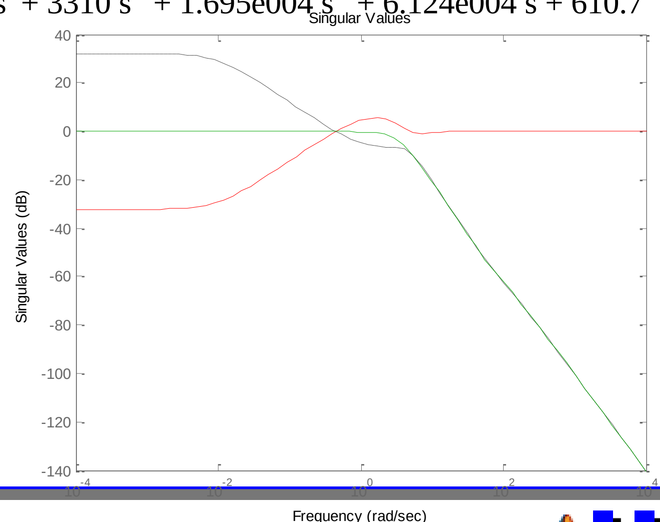
- Mixed sensitivity H_2 Loop-shaping

$$G(s)=\frac{10(s-1)}{(s+1)^2};\ w_1(s)=\frac{0.1(s+1000)}{100s+1};\ w_2(s)=0.1;\ w_3(s)=\frac{s^2}{(s+50)^2}$$

$$H(s)=\frac{0.9756\,s^4 + 99.51\,s^3 + 2635\,s^2 + 4975\,s + 2439}{s^5 + 147\,s^4 + 3310\,s^3 + 1.695e004\,s^2 + 6.124e004\,s + 610.7}$$

$$\min_{H(s)} \left\| \begin{matrix} w_1(s)S(s) \\ w_2(s)S(s)H(s) \\ w_3(s)T(s) \end{matrix} \right\|_2$$

```
s=zpk('s');
G=10*(s-1)/(s+1)^2;
W1=0.1*(s+100)/(100*s+1); W2=0.1; W3=s^2/(s+50)^2;
P=augtf(G,W1,W2,W3);
[K,CL,GAM]=h2syn(P);
L=G*K; S=inv(1+L); T=1-S;
sigma(L,'k-.',S,'r',T,'g')
```



H_∞ Controller Design - 1

- H_∞ **assumes disturbances to be bounded signals with finite energy** (unlike H_2 which assumes them to be white noise processes).

$$\underline{\dot{x}} = A\underline{x}(t) + B\underline{u}(t) + E\underline{w}(t)$$
$$\underline{z}(t) = C_1\underline{x}(t) + D_1\underline{u}(t)$$
$$\underline{y}(t) = C\underline{x}(t) + D\underline{u}(t) + D_2\underline{w}(t)$$

$\underline{x}$ ~	state
$\underline{z}$ ~	defines cost function
$\underline{y}$ ~	measurements (outputs)

Assumptions:

(i) $[A\ B]$ controllable (or stabilizable)

(ii) $[A\ C]$ observable (or detectable)

(iii) $V = \begin{bmatrix} E \\ D_2 \end{bmatrix}\begin{bmatrix} E^T & D_2^T \end{bmatrix} = \begin{bmatrix} EE^T & ED_2^T \\ D_2E^T & D_2D_2^T \end{bmatrix} \geq 0; D_2D_2^T > 0$

(iv) $Q = C_1^TC_1; M = C_1^TD_1; R = D_1^TD_1 > 0 \;\&\; Q - MR^{-1}M^T \geq 0$

H_∞ controller ensures the ∞-norm of $T_{zw}(j\omega) < \gamma$?

Lublin, Grocott and Athans,
Chapter 40 of Control Handbook

- What is H(s) ?

$$\underline{\hat{w}}(t) = -W_\infty \underline{\hat{x}}(t) \text{ and } \underline{u}(t) = -K_\infty \underline{\hat{x}}(t)$$
$$Estimate: \underline{\dot{\hat{x}}} = [A - BK_\infty - (E - Z_\infty L_\infty D_2)W_\infty - Z_\infty L_\infty (C - DK_\infty)]\underline{\hat{x}} + Z_\infty L_\infty \underline{y}$$
$$\underline{u}(s) = -H(s)\underline{y}(s) = -K_\infty (sI - A_\infty)^{-1} Z_\infty L_\infty \underline{y}(s)$$
$$where\ A_\infty = A - (E - Z_\infty L_\infty D_2)W_\infty - BK_\infty - Z_\infty L_\infty (C - DK_\infty)$$

H_∞ Controller Design - 2

- How to get the gains K_∞, L_∞ and W_∞: Via γ-coupled Riccati equations

$$K_\infty = R^{-1}B^T\tilde{P}_\infty + R^{-1}M^T;\ L_\infty = (\tilde{\Sigma}_\infty C^T + ED_2^T)(D_2D_2^T)^{-1};W_\infty = -\frac{1}{\gamma^2}E^T\tilde{P}_\infty;Z_\infty = (I_n - \frac{1}{\gamma^2}\tilde{\Sigma}_\infty\tilde{P}_\infty)^{-1}$$

$$\tilde{A} = A - BR^{-1}M^T;\tilde{Q} = Q - MR^{-1}M^T;\overline{A} = A - ED_2^T(D_2D_2^T)^{-1}C;\overline{W} = I - D_2^T(D_2D_2^T)^{-1}D_2$$

Control CARE: $\tilde{P}_\infty\tilde{A} + \tilde{A}^T\tilde{P}_\infty + \tilde{Q} - \tilde{P}_\infty(BR^{-1}B^T - \frac{1}{\gamma^2}EE^T)\tilde{P}_\infty = 0$

Estimation CARE: $\overline{A}\Sigma_\infty + \Sigma_\infty\overline{A}^T + E\overline{W}E^T - \Sigma_\infty\left(C^T(D_2D_2^T)^{-1}C - \frac{1}{\gamma^2}C_1^TC_1\right)\Sigma_\infty = 0$

- γ is such that it satisfies the following conditions:

(i) $\tilde{P}_\infty \geq 0$

(ii) The closed-loop control matrix $A - EW_\infty - BK_\infty$ is stable

(iii) $\tilde{\Sigma}_\infty \geq 0$

(iv) The closed-loop estimation matrix $A - Z_\infty L_\infty C + \frac{1}{\gamma^2}\tilde{\Sigma}_\infty C_1^TC_1$ is stable

(v) $|\lambda_{\max}(\tilde{\Sigma}_\infty\tilde{P}_\infty)| < \gamma^2$

H_∞ Control Design Examples

- Example (Loop shaping)

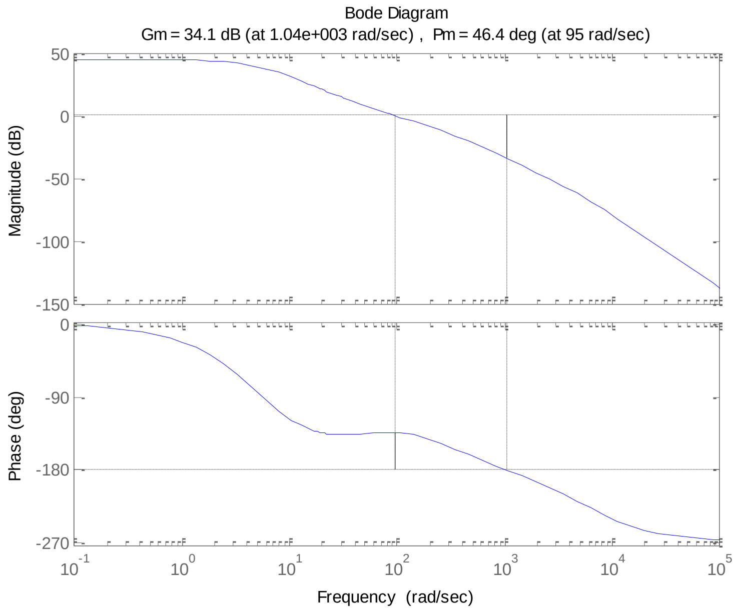
$$G(s) = \frac{400}{s^2 + 2s + 400};$$

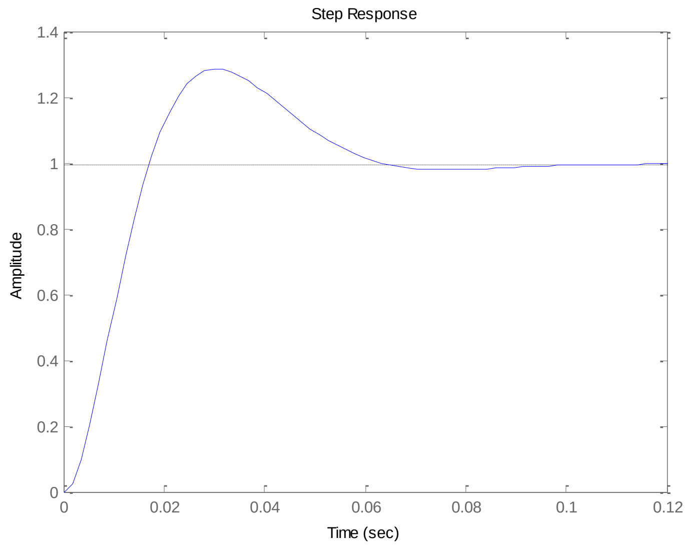
$$W_1(s) = \frac{100(0.005s+1)^2}{(0.2s+1)^2}; W_2 = 0.01; W_3(s) = \frac{s^2}{(s+200)^2}$$

$$H(s) = \frac{302897.6238\ (s+200)^2\ (s+47.14)\ (s^2 + 2s + 400)}{(s+5432)\ (s+134.5)\ (s+5)^2\ (s^2 + 502.1s + 7.049e004)}$$

```
gs=tf([400],[1 2 400])
s=tf('s')
W1=tf(100*conv([0.005 1],[0.005 1]),conv([0.2 1],[0.2 1]))
% W1=100*(0.005s+1)^2/(0.2*s+1)^2
W2=0.01
W3= tf([1 0 0],conv([1,200],[1 200]))
P=augtf(gs,W1,W2,W3)
[K,CL,GAM,INFO] = hinfsyn(P)
```

- The system is robust to wide range of damping term in *G(s)*. More robust by reducing W_2





H_∞ Control Design Examples

- Example 2 (X-29 Aircraft)

$$G(s) = \frac{20(s+3)(s-35)}{(s+10p/6)(s-p)(s+20)(s+35)}; \; p=3$$

$$W_1(s) = \frac{(s+10)}{(s+0.01)(1+0.0001s)}; W_2 = 0.01; W_3(s) = []$$

$$H(s) = \frac{-1773398.1663\,(s+8472)\,(s+35)\,(s+20)\,(s+5)\,(s+1.499)}{(s+1e004)\,(s+9421)\,(s+3)\,(s+0.01)\,(s^2 + 116.8s + 1.048e004)}$$

- Check that the system is robust to changes in *p*. It is stable even if *p=6*