16.333 Lecture # 10

State Space Control

- Basic state space control approaches
State Space Basics

- State space models are of the form

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t) + Du(t)
\end{align*}
\]

with associated transfer function

\[G(s) = C(sI - A)^{-1}B + D\]

Note: must form symbolic inverse of matrix \((sI - A)\), which is hard.

- **Time response:** Homogeneous part \(\dot{x} = Ax, \ x(0)\) known
  - Take Laplace transform

\[
X(s) = (sI - A)^{-1}x(0) \quad \Rightarrow x(t) = \mathcal{L}^{-1}\left[(sI - A)^{-1}\right]x(0)
\]
  - But can show \((sI - A)^{-1} = \frac{I}{s} + \frac{A}{s^2} + \frac{A^2}{s^3} + \ldots\)
  - So \(\mathcal{L}^{-1}\left[(sI - A)^{-1}\right] = I + At + \frac{1}{2!}(At)^2 + \ldots = e^{At}\)
  - Gives \(x(t) = e^{At}x(0)\) where \(e^{At}\) is **Matrix Exponential**
  - \(\hat{\diamond}\) Calculate in MATLAB® using `expm.m` and not `exp.m` \footnote{MATLAB® is a trademark of the Mathworks Inc.}

- **Time response:** Forced Solution - Matrix case \(\dot{x} = Ax + Bu\)
  where \(x\) is an \(n\)-vector and \(u\) is a \(m\)-vector. Can show

\[
\begin{align*}
x(t) &= e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \\
y(t) &= Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t)
\end{align*}
\]
  - \(Ce^{At}x(0)\) is the initial response
  - \(Ce^{At}B\) is the impulse response of the system.
Dynamic Interpretation

- Since $A = T \Lambda T^{-1}$, then
  
  $$e^{At} = Te^{A}T^{-1} = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix} \begin{bmatrix} e^{\lambda_1 t} & \cdots & e^{\lambda_n t} \end{bmatrix} \begin{bmatrix} -w_1^T & \vdots & -w_n^T \end{bmatrix}$$

  where we have written
  
  $$T^{-1} = \begin{bmatrix} -w_1^T & \vdots & -w_n^T \end{bmatrix}$$

  which is a column of rows.

- Multiply this expression out and we get that
  
  $$e^{At} = \sum_{i=1}^{n} e^{\lambda_i t} v_i w_i^T$$

- Assume $A$ diagonalizable, then $\dot{x} = Ax$, $x(0)$ given, has solution
  
  $$x(t) = e^{At}x(0) = Te^{A}T^{-1}x(0)$$

  $$= \sum_{i=1}^{n} e^{\lambda_i t} v_i \{ w_i^T x(0) \}$$

  $$= \sum_{i=1}^{n} e^{\lambda_i t} v_i \beta_i$$

- State solution is a **linear combination** of the system modes $v_i e^{\lambda_i}$

  - $e^{\lambda_i t}$ – Determines the **nature** of the time response
  - $v_i$ – Determines extent to which each state **contributes** to that mode
  - $\beta_i$ – Determines extent to which the initial condition **excites** the mode
• Note that the $v_i$ give the relative sizing of the response of each part of the state vector to the response.

$$v_1(t) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} e^{-t} \text{ mode 1}$$

$$v_2(t) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} e^{-3t} \text{ mode 2}$$

• Clearly $e^{\lambda_i t}$ gives the time modulation
  - $\lambda_i$ real – growing/decaying exponential response
  - $\lambda_i$ complex – growing/decaying exponential damped sinusoidal

• **Bottom line:** The locations of the eigenvalues determine the pole locations for the system, thus:
  - They determine the stability and/or performance & transient behavior of the system.

  - It is their locations that we will want to modify with the controllers.
Full-state Feedback Controller

• Assume that the single-input system dynamics are given by

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

so that \( D = 0 \).

– The multi-actuator case is quite a bit more complicated as we would have many extra degrees of freedom.

• Recall that the system poles are given by the eigenvalues of \( A \).

– Want to use the input \( u(t) \) to modify the eigenvalues of \( A \) to change the system dynamics.

![Diagram of control system]

• Assume a full-state feedback of the form:

\[ u = r - Kx \]

where \( r \) is some reference input and the gain \( K \) is \( \mathcal{R}^{1 \times n} \).

– If \( r = 0 \), we call this controller a regulator
• Find the closed-loop dynamics:

\[
\dot{x} = Ax + B(r - Kx) = (A - BK)x + Br = A_{cl}x + Br
\]

\[
y = Cx
\]

• **Objective:** Pick \( K \) so that \( A_{cl} \) has the desired properties, e.g.,

- \( A \) unstable, want \( A_{cl} \) stable
- Put 2 poles at \(-2 \pm 2j\)

• Note that there are \( n \) parameters in \( K \) and \( n \) eigenvalues in \( A \), so it looks promising, but what can we achieve?

• **Example #1:** Consider:

\[
\dot{x} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u
\]

- Then

\[
\det(sI - A) = (s - 1)(s - 2) - 1 = s^2 - 3s + 1 = 0
\]

so the system is unstable.

- Define \( u = -\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} x = -Kx \), then

\[
A_{cl} = A - BK = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} 1 - k_1 & 1 - k_2 \\ 1 & 2 \end{bmatrix}
\]

- So then we have that

\[
\det(sI - A_{cl}) = s^2 + (k_1 - 3)s + (1 - 2k_1 + k_2) = 0
\]
Thus, by choosing $k_1$ and $k_2$, we can put $\lambda_i(A_{cl})$ anywhere in the complex plane (assuming complex conjugate pairs of poles).

- To put the poles at $s = -5, -6$, compare the desired characteristic equation
  \[(s + 5)(s + 6) = s^2 + 11s + 30 = 0\]
  with the closed-loop one
  \[s^2 + (k_1 - 3)x + (1 - 2k_1 + k_2) = 0\]
  to conclude that
  \[
  \begin{align*}
  k_1 - 3 &= 11 \\
  1 - 2k_1 + k_2 &= 30 \\
  \end{align*}
  \]
  so that $K = \begin{bmatrix} 14 & 57 \end{bmatrix}$, which is called Pole Placement.

- Of course, it is not always this easy, as the issue of controllability must be addressed.

- **Example #2**: Consider this system:
  \[
  \dot{x} = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u
  \]
  with the same control approach
  \[
  A_{cl} = A - BK = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} 1 - k_1 & 1 - k_2 \\ 0 & 2 \end{bmatrix}
  \]
  so that $\det(sI - A_{cl}) = (s - 1 + k_1)(s - 2) = 0$
  The feedback control can modify the pole at $s = 1$, but it cannot move the pole at $s = 2$.

- **This system cannot be stabilized with full-state feedback control**.

- What is the reason for this problem?
- It is associated with loss of controllability of the $e^{2t}$ mode.

- Basic test for controllability: $\text{rank } M_c = n$

$$M_c = [B | AB] = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$$

So that $\text{rank } M_c = 1 < 2$.

- **Must assume that the pair** $(A, B)$ **are controllable.**
Ackermann’s Formula

• The previous outlined a design procedure and showed how to do it by hand for second-order systems.
  – Extends to higher order (controllable) systems, but tedious.

• **Ackermann’s Formula** gives us a method of doing this entire design process is one easy step.

  \[ K = \begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix} \mathcal{M}_c^{-1} \Phi_d(A) \]

  \[ - \mathcal{M}_c = \begin{bmatrix} B & AB & \ldots & A^{n-1}B \end{bmatrix} \]

  \[ - \Phi_d(s) \text{ is the characteristic equation for the closed-loop poles, which we then evaluate for } s = A. \]

  \[ - \text{It is explicit that the system must be controllable because we are inverting the controllability matrix.} \]

• Revisit **Example #1**: \( \Phi_d(s) = s^2 + 11s + 30 \)

  \[ \mathcal{M}_c = \begin{bmatrix} B \\ AB \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \]

  So

  \[ K = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{-1} \left( \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^2 + 11 \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} + 30I \right) \]

  \[ = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 43 & 14 \\ 14 & 57 \end{bmatrix} = \begin{bmatrix} 14 & 57 \end{bmatrix} \]

• Automated in Matlab: `place.m` & `acker.m` (see `polyvalm.m` too)
• Origins? For simplicity, consider a third-order system (case #2), but this extends to any order.

\[
A = \begin{bmatrix}
-a_1 & -a_2 & -a_3 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}
\]

— This form is useful because the characteristic equation for the system is obvious \( \Rightarrow \det(sI - A) = s^3 + a_1 s^2 + a_2 s + a_3 = 0 \)

• Can show that

\[
A_{cl} = A - BK = \begin{bmatrix}
-a_1 & -a_2 & -a_3 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}
\]

\[
= \begin{bmatrix}
-a_1 - k_1 & -a_2 - k_2 & -a_3 - k_3 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\]

so that the characteristic equation for the system is still obvious:

\[
\Phi_{cl}(s) = \det(sI - A_{cl}) = s^3 + (a_1 + k_1)s^2 + (a_2 + k_2)s + (a_3 + k_3) = 0
\]

• We then compare this with the desired characteristic equation developed from the desired closed-loop pole locations:

\[
\Phi_d(s) = s^3 + (\alpha_1)s^2 + (\alpha_2)s + (\alpha_3) = 0
\]

to get that

\[
\begin{align*}
a_1 + k_1 &= \alpha_1 \\
\vdots \\
a_n + k_n &= \alpha_n \\
k_1 &= \alpha_1 - a_1 \\
\vdots \\
k_n &= \alpha_n - a_n
\end{align*}
\]

• Pole placement is a very powerful tool and we will be using it for most of our state space work.
Aircraft State Space Control

• Can now design a full state feedback controller for the dynamics:

\[
\dot{x}_{sp} = A_{sp}x_{sp} + B_{sp}\delta_e
\]

with desired poles being at \( \omega_n = 3 \) and \( \zeta = 0.6 \) \( \Rightarrow s = -1.8 \pm 2.4i \)

\[
\phi_d(s) = s^2 + 3.6s + 9
\]

\[K_{sp} = \text{place}(A_{sp}, B_{sp}, \text{roots}([1 2*0.6*3 3^-2])')\]

• Design controller \( u = \begin{bmatrix} -0.0264 & -2.3463 \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} \)

• With full model, could arrange it so phugoid poles remain in the same place, just move the ones associated with the short period mode

\[
s = -1.8 \pm 2.4i, \quad -0.0033 \pm 0.0672i
\]

\[ev = \text{eig}(A);\]
\% damp short period, but leave the phugoid where it is
\[P_{list} = \text{roots}([1 2*.6*3 3^-2])' \text{ ev([3 4],1)'};\]
\[K_1 = \text{place}(A,B(:,1),P_{list})\]

\[\Rightarrow u = \begin{bmatrix} 0.0026 & -0.0265 & -2.3428 & 0.0363 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix}\]
• Can also add the lag dynamics to short period model with $\theta$ included

$$\dot{x}_{sp} = \tilde{A}_{sp} x_{sp} + \tilde{B}_{sp} \delta^a_e, \quad \delta^a_e = \frac{4}{s + 4} \delta^c_e$$

$$\rightarrow \dot{x}_\delta = -4x_\delta + 4\delta^c_e, \quad \delta^a_e = x_\delta$$

$$\Rightarrow \begin{bmatrix} \dot{x}_{sp} \\ \dot{x}_\delta \end{bmatrix} = \begin{bmatrix} \tilde{A}_{sp} & \tilde{B}_{sp} \\ 0 & -4 \end{bmatrix} \begin{bmatrix} x_{sp} \\ x_\delta \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \end{bmatrix} \delta^c_e$$

• Add $s = -3$ to desired pole list

\[
\text{Plist} = [\text{roots}([1 2*.6*3 3^2])', -.25, -3];
\]

\[
\text{At2} = [\text{Asp2} \ Bsp2(:,1); \text{zeros}(1,3) -4]; \text{Bt2} = [\text{zeros}(3,1); 4]; \text{Kt} = \text{place}(\text{At2}, \text{Bt2}, \text{Plist});
\]

\[
\text{step}(\text{ss}(\text{At2}-\text{Bt2}*\text{Kt2}, \text{Bt2}, [0 0 1 0], 0), 35)
\]

\[
u = \begin{bmatrix} 0.0011 & -3.4617 & -4.9124 & 0.5273 \end{bmatrix}
\]

• No problem working with larger systems with state space tools

• Main control issue is finding “good” locations for closed-loop poles
Estimators/Observers

- **Problem:** So far we have assumed that we have full access to the state $x(t)$ when we designed our controllers.
  - Most often all of this information is not available.

- Usually can only feedback information that is developed from the sensors measurements.
  - Could try “output feedback”
    $$ u = Kx \Rightarrow u = \hat{K}y $$
  - Same as the proportional feedback we looked at at the beginning of the root locus work.
  - This type of control is very difficult to design in general.

- **Alternative approach:** Develop a replica of the dynamic system that provides an “estimate” of the system states based on the measured output of the system.

- **New plan:**
  1. Develop estimate of $x(t)$ that will be called $\hat{x}(t)$.
  2. Then switch from $u = -Kx(t)$ to $u = -K\hat{x}(t)$.

- Two key questions:
  - How do we find $\hat{x}(t)$?
  - Will this new plan work?
Estimation Schemes

- Assume that the system model is of the form:

\[
\dot{x} = Ax + Bu, \quad x(0) \text{ unknown}
\]

\[
y = Cx
\]

where

1. \(A, B,\) and \(C\) are known.
2. \(u(t)\) is known
3. Measurable outputs are \(y(t)\) from \(C \neq I\)

- **Goal:** Develop a dynamic system whose state

\[
\hat{x}(t) = x(t)
\]

for all time \(t \geq 0\). Two primary approaches:

- Open-loop.
- Closed-loop.
Open-loop Estimator

- Given that we know the plant matrices and the inputs, we can just perform a simulation that runs in parallel with the system

\[
\dot{x}(t) = A\dot{x} + Bu(t)
\]

- Then \( \dot{x}(t) \equiv x(t) \ \forall \ t \) provided that \( \dot{x}(0) = x(0) \)

- **Major Problem:** We do not know \( x(0) \)

- Analysis of this case. Start with:

\[
\begin{align*}
\dot{x}(t) &= Ax + Bu(t) \\
\dot{\hat{x}}(t) &= A\hat{x} + Bu(t)
\end{align*}
\]

- Define the **estimation error**: \( \tilde{x}(t) = x(t) - \hat{x}(t) \).
  
  - Now want \( \tilde{x}(t) = 0 \ \forall \ t \).
  
  - But is this realistic?
• Subtract to get:

\[
\frac{d}{dt}(x - \hat{x}) = A(x - \hat{x}) \Rightarrow \dot{x}(t) = A\hat{x}
\]

which has the solution

\[
x(t) = e^{At}\hat{x}(0)
\]

– Gives the estimation error in terms of the initial error.

• Does this guarantee that \(\hat{x} = 0\ \forall\ t\)?

Or even that \(\hat{x} \rightarrow 0\) as \(t \rightarrow \infty\)? (which is a more realistic goal).

– Response is fine if \(\hat{x}(0) = 0\). But what if \(\hat{x}(0) \neq 0\)?

• If \(A\) stable, then \(\hat{x} \rightarrow 0\) as \(t \rightarrow \infty\), but the dynamics of the estimation error are completely determined by the open-loop dynamics of the system (eigenvalues of \(A\)).

– Could be very slow.

– No obvious way to modify the estimation error dynamics.

• Open-loop estimation does not seem to be a very good idea.
Closed-loop Estimator

- Obvious way to fix the problem is to use the additional information available:
  - How well does the estimated output match the measured output?
  
    Compare: \( y = Cx \) with \( \hat{y} = C\hat{x} \)
  
  - Then form \( \tilde{y} = y - \hat{y} = Cx \)

\[ \begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) + L\dot{y}(t) \\
\hat{y}(t) &= C\hat{x}(t)
\end{align*} \]

where \( L \) is a user selectable gain matrix.

- **Analysis:**

  \[ \begin{align*}
  \hat{x} &= \dot{x} - \hat{x} = [Ax + Bu] - [A\hat{x} + Bu + L(y - \hat{y})] \\
  &= A(x - \hat{x}) - L(Cx - C\hat{x}) \\
  &= A\hat{x} - LC\hat{x} = (A - LC)\hat{x}
\end{align*} \]
• So the closed-loop estimation error dynamics are now
  \[ \dot{\hat{x}} = (A - LC)\hat{x} \] with solution \( \hat{x}(t) = e^{(A-LC)t} \hat{x}(0) \)

• **Bottom line:** Can select the gain \( L \) to attempt to improve the convergence of the estimation error (and/or speed it up).
  – But now must worry about observability of the system model.

• Note the similarity:
  – **Regulator Problem:** pick \( K \) for \( A - BK \)
    ◦ Choose \( K \in \mathcal{R}^{1 \times n} \) (SISO) such that the closed-loop poles
    \[ \det(sI - A + BK) = \Phi_c(s) \]
    are in the desired locations.
  – **Estimator Problem:** pick \( L \) for \( A - LC \)
    ◦ Choose \( L \in \mathcal{R}^{n \times 1} \) (SISO) such that the closed-loop poles
    \[ \det(sI - A + LC) = \Phi_o(s) \]
    are in the desired locations.

• These problems are obviously very similar – in fact they are called **dual problems**.
Estimation Gain Selection

- For regulation, we are concerned with controllability of \((A, B)\)

For a controllable system we can place the eigenvalues of \(A - BK\) arbitrarily.

- For estimation, we are concerned with observability of pair \((A, C)\).

For an observable system we can place the eigenvalues of \(A - LC\) arbitrarily.

- Test using the observability matrix:

\[
\text{rank } M_o \triangleq \text{rank} \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} = n
\]

- The procedure for selecting \(L\) is very similar to that used for the regulator design process.
• One approach:
  – Note that the poles of \((A - LC)\) and \((A - LC)^T\) are identical.
  – Also we have that \((A - LC)^T = A^T - C^T L^T\)
  – So designing \(L^T\) for this transposed system looks like a standard regulator problem \((A - BK)\) where
    \[
    \begin{align*}
    A & \Rightarrow A^T \\
    B & \Rightarrow C^T \\
    K & \Rightarrow L^T
    \end{align*}
    \]
    So we can use
    \[
    K_e = \text{acker}(A^T, C^T, P) , \quad L \equiv K_e^T
    \]

• Note that the estimator equivalent of Ackermann’s formula is that
  \[
  L = \Phi_e(s) M_o^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}
  \]
Simple Estimator Example

• Simple system

\[
A = \begin{bmatrix} -1 & 1.5 \\ 1 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad x(0) = \begin{bmatrix} -0.5 \\ -1 \end{bmatrix}
\]

\[
C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0
\]

– Assume that the initial conditions are not well known.

– System stable, but \( \lambda_{\text{max}}(A) = -0.18 \)

– Test observability:

\[
\text{rank} \begin{bmatrix} C \\ CA \end{bmatrix} = \text{rank} \begin{bmatrix} 1 & 0 \\ -1 & 1.5 \end{bmatrix}
\]

• Use open and closed-loop estimators. Since the initial conditions are not well known, use \( \hat{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \)

• Open-loop estimator:

\[
\begin{align*}
\dot{x} &= A\hat{x} + Bu \\
\hat{y} &= C\hat{x}
\end{align*}
\]

• Closed-loop estimator:

\[
\begin{align*}
\dot{x} &= A\hat{x} + Bu + L\hat{y} = A\hat{x} + Bu + L(y - \hat{y}) \\
&= (A - LC)\hat{x} + Bu + Ly \\
\hat{y} &= C\hat{x}
\end{align*}
\]

– Which is a dynamic system with poles given by \( \lambda_i(A - LC) \) and which takes the measured plant outputs as an input and generates an estimate of \( x \).
• Typically simulate both systems together for simplicity

• Open-loop case:

\[ \dot{x} = Ax + Bu \]
\[ y = Cx \]
\[ \hat{x} = A\hat{x} + Bu \]
\[ \hat{y} = C\hat{x} \]

\[ \Rightarrow \begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} u, \quad \begin{bmatrix} x(0) \\ \hat{x}(0) \end{bmatrix} = \begin{bmatrix} -0.5 \\ -1 \\ 0 \\ 0 \end{bmatrix} \]
\[ \begin{bmatrix} y \\ \hat{y} \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} \]

• Closed-loop case:

\[ \dot{x} = Ax + Bu \]
\[ \hat{x} = (A - LC)\hat{x} + Bu + LCx \]

\[ \Rightarrow \begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ LC & A - LC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} u \]

• Example uses a strong \( u(t) \) to shake things up
Figure 1: Open-loop estimator. Estimation error converges to zero, but very slowly.

Figure 2: Closed-loop estimator. Convergence looks much better.
Aircraft Estimation Example

- Take Short period model and assume that we can measure \( q \). Can we estimate the motion associated with the short period mode?

\[
\dot{x}_{sp} = A_{sp} x_{sp} + B_{sp} u \\
y = \begin{bmatrix} 0 & 1 \end{bmatrix} x_{sp}
\]

- Take \( x_{sp}(0) = [-0.5; -0.05]^T \)

- System stable, so could use an open loop estimator
- For closed-loop estimator, put desired poles at \(-3, -4\)
- For the various dynamics models as before

\[
C_{sp} = [0 \ 1]; \quad \% \ \text{sense} \ q \\
K_{e} = \text{place}(A_{sp}', C_{sp}', [-3 \ -4]); L_{e} = K_{e}';
\]

![Figure 3: Closed-loop estimator. Convergence looks much better.](image)

- As expected, the OL estimator does not do well, but the closed-loop one converges nicely
Where to put the Estimator Poles?

- Location heuristics for poles still apply – use Bessel, ITAE, ...
  - Main difference: probably want to make the estimator faster than you intend to make the regulator – should enhance the control, which is based on $\dot{x}(t)$.
  - ROT: Factor of 2–3 in the time constant $\zeta\omega_n$ associated with the regulator poles.

- **Note:** When designing a regulator, were concerned with “bandwidth” of the control getting too high ⇒ often results in control commands that saturate the actuators and/or change rapidly.

- Different concerns for the estimator:
  - Loop closed inside computer, so saturation not a problem.
  - However, the measurements $y$ are often “noisy”, and we need to be careful how we use them to develop our state estimates.

⇒ **High bandwidth estimators** tend to accentuate the effect of sensing noise in the estimate.
  - State estimates tend to “track” the measurements, which are fluctuating randomly due to the noise.

⇒ **Low bandwidth estimators** have lower gains and tend to rely more heavily on the plant model
  - Essentially an open-loop estimator – tends to ignore the measurements and just uses the plant model.
• Can also develop an **optimal estimator** for this type of system.
  – Which is apparently what Kalman did one evening in 1958 while taking the train from Princeton to Baltimore...
  
  – **Balances effect** of the various types of random noise in the system on the estimator:

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

where:

◇ \( w \): “process noise” – models uncertainty in the system model.

◇ \( v \): “sensor noise” – models uncertainty in the measurements.

---

**Final Thoughts**

• Note that the feedback gain \( L \) in the estimator only stabilizes the estimation error.
  – If the system is unstable, then the state estimates will also go to \( \infty \), with zero error from the actual states.

• Estimation is an important concept of its own.
  – Not always just “part of the control system”
  
  – Critical issue for guidance and navigation system

• More complete discussion requires that we study stochastic processes and optimization theory.

• **Estimation is all about which do you trust more: your measurements or your model.**
Combined Regulator and Estimator

- As advertised, we can change the previous control \( u = -Kx \) to the new control \( u = -\hat{K}\hat{x} \) (same \( K \)). We now have

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx
\end{align*}
\]

\[
\begin{align*}
\dot{x} &= A\hat{x} + Bu + L(y - \hat{y}) \\
\dot{\hat{y}} &= C\hat{x}
\end{align*}
\]

with closed-loop dynamics

\[
\begin{bmatrix}
\dot{x} \\
\dot{\hat{x}}
\end{bmatrix} = \begin{bmatrix}
A & -BK \\
LC & A - BK - LC
\end{bmatrix} \begin{bmatrix}
x \\
\hat{x}
\end{bmatrix} \Rightarrow \dot{x}_{cl} = A_{cl}x_{cl}
\]

- Not obvious that this system will even be stable: \( \lambda_i(A_{cl}) < 0? \)

- To analyze, introduce \( \tilde{x} = x - \hat{x} \), and the similarity transform

\[
T = \begin{bmatrix}
I & 0 \\
I & -I
\end{bmatrix} = T^{-1}
\]

- Rewrite the dynamics in terms of the state

\[
\begin{bmatrix}
x \\
\hat{x}
\end{bmatrix} = T \begin{bmatrix}
x \\
\hat{x}
\end{bmatrix}
\]

\[
A_{cl} \Rightarrow T^{-1}A_{cl}T \equiv \overline{A}_{cl}
\]

and when you work through the math, you get

\[
\overline{A}_{cl} = \begin{bmatrix}
A - BK & BK \\
0 & A - LC
\end{bmatrix}
\]
• Absolutely key points:

1. \( \lambda_i(A_{cl}) \equiv \lambda_i(\overline{A_{cl}}) \) [why?]
2. \( A_{cl} \) is block upper triangular, so can find poles by inspection:
   \[
   \det(sI - A_{cl}) = \det(sI - (A - BK)) \cdot \det(sI - (A - LC))
   \]

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

• So we can design the estimator and regulator separately with confidence that combination of the two will work VERY well.

• Compensator is a combination of the estimator and regulator.

\[
\dot{x} = A\hat{x} + Bu + L(y - \hat{y})
= (A - BK - LC)\hat{x} + Ly
u = -K\hat{x}
\]

\[
\Rightarrow \dot{x}_c = A_c x_c + B_c y \\
u = -C_c x_c
\]

— Keep track of this minus sign. We need one in the feedback path, but we can move it around to suit our needs.
• Let $G_c(s)$ be the compensator transfer function where
\[
\frac{u}{y} = -C_c(sI - A_c)^{-1}B_c = -G_c(s)
\]
\[
= -K(sI - (A - BK - LC'))^{-1}L
\]
so by my definition, $\Rightarrow u = -G_cy \equiv G_c(-y)$

• Reason for making the definition is that when we implement the controller, we often do not just feedback $-y(t)$, but instead have to include a reference command $r(t)$
  - Use **servo approach** and feed back $e(t) = r(t) - y(t)$ instead

\[ \begin{array}{cccc}
  r & e & G_c(s) & u & G(s) & y \\
  - & & & & & \\
\end{array} \]

  - So now $u = G_ce = G_c(r - y)$.
  - And if $r = 0$, then we still have $u = G_c(-y)$

• Important points:
  - Closed-loop system will be stable, but the compensator dynamics need not be.
  - Often very simple and useful to provide classical interpretations of the compensator dynamics $G_c(s)$. 
• Mechanics of closing the loop

\( G(s) : \dot{x} = Ax + Bu \)
\( y = Cx \)

\( G_c(s) : \dot{x}_c = A_c x_c + B_c e \)
\( u = C_c x_c \)

and \( e = r - y, u = G_c e, y = Gu. \)

• Loop dynamics \( L = GG_c \Rightarrow y = L(s)e \)

\[
\begin{align*}
\dot{x} &= Ax + Bu = Ax + BC_c x_c \\
\dot{x}_c &= A_c x_c + B_c e \\
\begin{bmatrix}
\dot{x} \\
\dot{x}_c \\
\end{bmatrix} &= 
\begin{bmatrix}
A & BC_c \\
0 & A_c \\
\end{bmatrix}
\begin{bmatrix}
x \\
x_c \\
\end{bmatrix} + 
\begin{bmatrix}
0 \\
B_c \\
\end{bmatrix} e \\
y &= \begin{bmatrix}
C & 0 \\
\end{bmatrix}
\begin{bmatrix}
x \\
x_c \\
\end{bmatrix}
\end{align*}
\]

• Now form the closed-loop dynamics by inserting \( e = r - y \)

\[
\begin{align*}
\begin{bmatrix}
\dot{x} \\
\dot{x}_c \\
\end{bmatrix} &= 
\begin{bmatrix}
A & BC_c \\
0 & A_c \\
\end{bmatrix}
\begin{bmatrix}
x \\
x_c \\
\end{bmatrix} + 
\begin{bmatrix}
0 \\
B_c \\
\end{bmatrix} \left( r - \begin{bmatrix}
C & 0 \\
\end{bmatrix}
\begin{bmatrix}
x \\
x_c \\
\end{bmatrix} \right) \\
&= \begin{bmatrix}
A & BC_c \\
-B_c C & A_c \\
\end{bmatrix}
\begin{bmatrix}
x \\
x_c \\
\end{bmatrix} + 
\begin{bmatrix}
0 \\
B_c \\
\end{bmatrix} r \\
y &= \begin{bmatrix}
C & 0 \\
\end{bmatrix}
\begin{bmatrix}
x \\
x_c \\
\end{bmatrix}
\end{align*}
\]
Performance Issue

- Often find with state space controllers that the DC gain of the closed loop system is not $1$. So $y \neq r$ in steady state.

- Relatively simple fix is to modify the original controller with scalar $N$

  $$u = r - Kx \Rightarrow u = Nr - Kx$$

- Closed-loop system on page 5 becomes

  $$\begin{align*}
  \dot{x} &= Ax + B(Nr - Kx) = A_{cl}x + BNr \\
  y &= Cx
  \end{align*}$$

  $$G_{cl}(s) = C(sI - A_{cl})^{-1}BN$$

  - Analyze steady state step response $\Rightarrow y_{ss} = G_{cl}(0)r_{step}$

  $$G_{cl}(0) = C(-A_{cl})^{-1}BN$$

  - And pick $N$ so that $G_{cl}(0) = 1 \Rightarrow N = \frac{1}{(C(-A_{cl})^{-1}B)}$

- A bit more complicated with a combined estimator and regulator

  - One simple way (not the best) of achieving a similar goal is to add $N$ to $r$ and force $G_{cl}(0) = 1$

  - Now the closed-loop dynamics on page 29 become:

    $$\begin{align*}
    \begin{bmatrix}
    \dot{x} \\
    \dot{x}_c
    \end{bmatrix} &= A_{cl} \begin{bmatrix} x \\
    x_c \end{bmatrix} + B_{cl}Nr \\
    y &= C_{cl} \begin{bmatrix} x \\
    x_c \end{bmatrix}
    \end{align*}$$

    $$\Rightarrow N = \frac{1}{(C_{cl}(-A_{cl})^{-1}B_{cl})}$$

- Note that this fixes the steady state tracking error problems, but in my experience can create strange transients (often NMP).
Example: Compensator Design

\[ G(s) = \frac{1}{s^2 + s + 1} \Rightarrow \dot{x} = Ax + Bu \]
\[ y = Cx \]

where

\[ A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \]

- Regulator: Want regulator poles to have a time constant of \( \tau_c = 1/(\zeta \omega_n) = 0.25 \text{ sec} \Rightarrow \lambda(A - BK_r) = -4 \pm 4j \) which can be found using \textbf{place} or \textbf{acker}

\[
K_r = \text{acker}(a, b, [-4+4*j; -4-4*j]);
\]

to give

\[
K_r = \begin{bmatrix} 31 \\ 7 \end{bmatrix}
\]

- Estimator: want the estimator poles to be faster, so use

\( \tau_e = 1/(\zeta \omega_n) = 0.1 \text{ sec.} \) Use real poles, \( \Rightarrow \lambda(A - L_e C) = -10 \)

\[
L_e = \text{acker}(a', c', [-10 \ -10]');
\]

which gives

\[
L_e = \begin{bmatrix} 19 \\ 80 \end{bmatrix}
\]

- Form compensator \( G_c(s) \)

\[
ac = a - b*K_r - L_e*c; bc = L_e; cc = K_r; dc = 0;
\]

\[
A_c = \begin{bmatrix} -19 & 1 \\ -112 & -8 \end{bmatrix} \quad B_c = \begin{bmatrix} 19 \\ 80 \end{bmatrix} \quad C_c = \begin{bmatrix} 31 & 7 \end{bmatrix}
\]

\[
G_c(s) = \frac{1149(s + 2.5553)}{s^2 + 27s + 264} = \frac{u}{e}
\]

Low frequency zero, with higher frequency poles (like a lead)
Figure 4: The compensator does indeed look like a high frequency lead (amplification from 2–16 rad/sec). Plant pretty simple looking.
Figure 5: The loop transfer function $L = G_c G$ shows a slope change around $\omega_c = 5$ rad/sec due to the effect of the compensator. Significant gain and phase margins.
Figure 6: Quite significant gain and phase margins.
Figure 7: Freeze the compensator poles and zeros and draw a root locus versus an additional plant gain $\alpha$, $G(s) \Rightarrow \tilde{G}(s) = \frac{\alpha}{(s^2 + s + 1)}$. Note location of the closed-loop poles!!
Figure 8: Closed-loop transfer – system bandwidth has increased substantially.
clear all
close all
figure(1);clf
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultLineMarkerSize',10)
figure(2);clf
set(gcf,'DefaultLineLineWidth',2)
set(gcf,'DefaultLineMarkerSize',10)

load b747
A = B = Asp = Bsp;
Csp = [0 1]; % sense q

Ke = place(Asp', Csp', [­3 ­4]); Le = Ke';
xo = [­.5; ­.05]; % start somewhere

t = [0: .01: 10]; N = floor(.15*length(t));

u = [0; [ones(N, 1);­ones(N, 1);ones(N, 1)/2;­ones(N, 1)/2;zeros(41, 1)]/5];
u = [0; [ones(N, 1);­ones(N, 1);ones(N, 1)/2;­ones(N, 1)/2]/20; u(length(t)) = 0;

[y, x] = lsim(Asp, Bsp, Csp, 0, u, t, xo);
plot(t, y)

% closed-loop estimator
% hook both up so that we can simulate them at the same time
% bigger state = state of the system then state of the estimator
A_cl = [Asp zeros(size(Asp)); Le*Csp­Asp­Le*Csp];
B_cl = [Bsp; Bsp];
C_cl = [Csp zeros(size(Csp)); zeros(size(Csp)) Csp];
D_cl = zeros(2,1);

% note that we start the estimators at zero, since that is
% our current best guess of what is going on (i.e. we have no clue :-)
% [y_cl, x_cl] = lsim(A_cl, B_cl, C_cl, D_cl, u, t, [xo; 0; 0]);

% open-loop estimator
% hook both up so that we can simulate them at the same time
% bigger state = state of the system then state of the estimator
A_ol = [Asp zeros(size(Asp)); zeros(size(Asp)) Asp];
B_ol = [Bsp; Bsp];
C_ol = [Csp zeros(size(Csp)); zeros(size(Csp)) Csp];
D_ol = zeros(2,1);

[y_ol, x_ol] = lsim(A_ol, B_ol, C_ol, D_ol, u, t, [xo; 0; 0]);
subplot(222)
plot(t,x_ol(:,[2]),t,x_ol(:,[4]),'--')
ylabel('x1');xlabel('time');grid
subplot(223)
plot(t,x_ol(:,[1])-x_ol(:,[3]))
ylabel('x1 error');xlabel('time');grid
subplot(224)
plot(t,x_ol(:,[2])-x_ol(:,[4]))
ylabel('x2 error');xlabel('time');grid
print -depsc spest_ol.eps
jpdf('spest_ol')
% Combined estimator/regulator design for a simple system
% G = 1/(s^2+s+1)
% Jonathan How
% Fall 2004
% close all; clear all
for ii=1:5
    figure(ii); clf; set(gcf,'DefaultLineWidth',2); set(gcf,'DefaultMarkerSize',10)
end

a=[0 1; -1 -1]; b=[0 1]; c=[1 0]; d=0;
k=acker(a,b,[-4+4*j;-4-4*j]);
l=acker(a',c',[-10 -10]');

% For state space for G_c(s)
ac=a-b*k-l*c; bc=l; cc=k; dc=0;
G=ss(a,b,c,d);
Gc=ss(ac,bc,cc,dc);

f=logspace(-1,2,400);
g=freqresp(G,f*j); g=squeeze(g);
gc=freqresp(Gc,f*j); gc=squeeze(gc);

figure(1); clf
subplot(211)
loglog(f,abs(g),f,abs(gc),'-'); axis([.1 1e2 .2 1e2])
xlabel('Freq (rad/sec)'); ylabel('Mag')
legend('Plant G','Compensator Gc'); grid
subplot(212)
semilogx(f,180/pi*angle(g),f,180/pi*angle(gc),'-');
axis([.1 1e2 -200 50])
xlabel('Freq (rad/sec)'); ylabel('Phase (deg)'); grid
legend('Plant G','Compensator Gc')

L=g.*gc;
figure(2); clf
subplot(211)
loglog(f,abs(L),f,abs(gc),'-'); axis([.1 1e2 .2 1e2])
xlabel('Freq (rad/sec)'); ylabel('Mag')
legend('Loop L'); grid
subplot(212)
semilogx(f,180/pi*angle(L),f,180/pi*angle(gc),'-');
axis([.1 1e2 -290 0])
xlabel('Freq (rad/sec)'); ylabel('Phase (deg)'); grid

% loop dynamics L = G Gc
al=[a b*cc zeros(2) ac];
bcl=[zeros(2,1);bc];
lc=[c zeros(1,2)];
dcl=0;
figure(3)
rlocus(al,bcl,lc,dcl)

% closed-loop dynamics
% unity gain wrapped around loop L
al=al-bcl*lc; bcl=bcl;lc=cl;dcl=d;
N=inv(ccl*inv(-acl)+bcl)
hold on; plot(eig(acl), 'd'); hold off
grid
%
% closed-loop freq response
%
Gcl = ss(acl, bcl+N, ccl, dcl);
gcl = freqresp(Gcl, f*1i); gcl = squeeze(gcl);
figure(4); clf
loglog(f, abs(g), f, abs(gcl), '--'); axis([.1 1e2 .01 1e2])
xlabel('Freq (rad/sec)'); ylabel('Mag')
legend('Plant G', 'closed-loop \(G_{cl}\)'); grid
title(['Factor of \(N=\)', num2str(N), ' applied to Closed loop'])

figure(5); clf
margin(al, bl, cl, dl)

figure(1); orient tall; print -depsc reg_est1.eps
jpdf('reg_est1')
figure(2); orient tall; print -depsc reg_est2.eps
jpdf('reg_est2')
figure(3); print -depsc reg_est3.eps
jpdf('reg_est3')
figure(4); print -depsc reg_est4.eps
jpdf('reg_est4')
figure(5); print -depsc reg_est5.eps
jpdf('reg_est5')