Modal Analysis

Suppose that $\lambda_k$ is complex

\[ \lambda_k = \sigma_k + j\omega_k \]

we have $(\lambda_k, x_k)$, $(\bar{\lambda}_k, \bar{x}_k)$
interested in complex sinusoid

Fact: If $x(t) = e^{\lambda_t} x_0$
$\quad x(0) = x_0$

Fact: If $x(0) = x_k$
$\quad \text{then}, \quad x(t) = e^{\lambda_k t} x_k$

Fact: If $x(0) = \bar{x}_k$
$\quad \text{then}, \quad x(t) = e^{\bar{\lambda}_k t} \bar{x}_k$

Main Punch Line

Suppose, $x(0) = x_k + \bar{x}_k$
$\quad = 2\Re\{ e^{\lambda_k t} x_k \}$ ← can apply this in practice.
$\quad$ cannot apply $x_k$
because complex

\[ x(t) = e^{\lambda_k t} x_k + e^{\bar{\lambda}_k t} \bar{x}_k \]
$\quad = 2\Re\{ e^{\lambda_k t} x_k \} \quad \leftarrow \text{real function of time} \]
\[ e^{\lambda t} = e^{\sigma t} e^{jw t} \quad \text{because} \quad \lambda = \sigma + jw \]

\[ x_K = \begin{bmatrix} 1 & e^{jx_K} \\ \vdots \\ 1 & e^{jx_{Kn}} \end{bmatrix} \]

\[ x_i(0) = \frac{x_K - \overline{x_K}}{j^2} = \text{Im} \{ x_K \} \]

\[ x_i(t) = e^{\sigma t} |x_K| \sin (\omega t + \angle x_K) \]

**Know above and be able to duplicate!**

**New Question,**

To find zero's examine

\[
\begin{bmatrix}
-2A & -B \\
C & 0
\end{bmatrix}
\]

To find zero direction, examine

\[
\begin{bmatrix}
-2A & -B \\
C & 0
\end{bmatrix} \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

Where does this come from?

Zero \( \left( z_0, x_0, u_0 \right) \) direction \( \Rightarrow x(t) = e^{z_0 t} x_0 \) and \( y(t) = 0 \) when \( u(t) = e^{z_0 t} \),

\[ x(0) = x_0 \]
Def of Transmission Zeros for any system.

\((z_0, x_0, u_0) \iff e^{z_0 t} x_0 \text{ and } y(t) = 0\)

is a zero when \(e^{z_0 t} u(t)\), \(x(0) = x_0\)

\[\begin{bmatrix}
    z_0 I - A & -B \\
    C & D
\end{bmatrix}
\begin{bmatrix}
    x_0 \\
    u_0
\end{bmatrix} = \begin{bmatrix}
    0 \\
    0
\end{bmatrix}\]

"Proof"

\(\dot{x} = Ax + Bu\)

\(sX - x_0 = AX + BU\)

\((sI - A)X = x_0 + BU\)

\[(sI-A) \frac{x_0}{s-z_0} = x_0 + B \frac{u_0}{s-z_0}\]

\[= \frac{(s-z_0)x_0 + BU_0}{s-z_0}\]

\[= \frac{(sI-z_0I-A+A)x_0 + BU_0}{s-z_0}\]

\[= \frac{(sI-A)x_0 - (z_0I-A)x_0 + BU_0}{s-z_0}\]

\((sI-A)x_0 = (sI-A)x_0 - (z_0I-A)x_0 + BU_0\)

\[\begin{bmatrix}
    (z_0 I - A) - B \\
    C & D
\end{bmatrix}
\begin{bmatrix}
    x_0 \\
    u_0
\end{bmatrix} = \begin{bmatrix}
    0 \\
    0
\end{bmatrix}\]

\(y(t) = CX + DU\)

\[y = CX + DU\]

\[0 = C \frac{x_0}{s-z_0} + D \frac{u_0}{s-z_0}\]

done
\[
\begin{align*}
\mathbf{z} &= \begin{bmatrix} m_L & m_W \end{bmatrix} \begin{bmatrix} x_p \\
\end{bmatrix} \\
G_{ol}(s) &= M(sI-A)^{-1}B
\end{align*}
\]

Matching at Low Freq:
\[
m_c = U_k [C_p (-A_p)^{-1} B_p]^{-1}
\]
\[
U_k^{-1} = U_k^H - U_k \text{ arbitrary matrix}
\]

Matching at High Freq:
\[
m_H = U_k [B_p^T B_p]^{-1} B_p^T
\]

\[
P = \begin{bmatrix} A_p, B_p, C_p \end{bmatrix}
\]

Design Plant:
\[
G_{ol} = \begin{bmatrix} A, B, M \end{bmatrix}
\]
\[
Q = m^TM
\]
\[
f \text{ adjusts BW}
\]

1. Take Plant
2. Augment Plant
   \[
   \text{resulting augmented plant is the Design Plant which is thrown into the \textit{Raciti} Machinery}
   \]

Singular Values of \( G_{ol} \)

- Matched high & low
- Don't know at middle frequency

Feed this to LQ Design Machinery
LQ loop: \( G_{wa} = [A, B, G] \)

Singular Values, might expect this

Remember,

1. Use \( m \) to control shape
   \( p \) to adjust BW (move plot up or down)
   
   - Small \( p \) \( \iff \) Large BW
   - Large \( p \) \( \iff \) Small BW

2. Matching may be physically RIDICULOUS!
   E.g., when units are different.
   
   Ex., gas pressure can usually be changed much more quickly than a liquid level.

\( \Rightarrow \) We must ask for reasonable things.

Pg 11 of set #4

**F8 - LQ servo Design**

\[ \begin{align*}
    \mathbf{u}(s) & \rightarrow \mathbf{B} \rightarrow \Phi \rightarrow \mathbf{x}(s) \rightarrow \mathbf{G} \rightarrow \mathbf{y}(s) \\
    \mathbf{B} & = [\mathbf{B}_{1}, \ldots, \mathbf{B}_{n}] \\
    \Phi & = [\phi_{1}, \ldots, \phi_{n}] \\
    \mathbf{G} & = [g_{1}, \ldots, g_{n}] \\
    \mathbf{y} & = \begin{bmatrix} \mathbf{y}_{1} \\ \vdots \\ \mathbf{y}_{n} \end{bmatrix} \\
    \mathbf{u}(s) & = \begin{bmatrix} u_{1}(s) \\ \vdots \\ u_{n}(s) \end{bmatrix} \\
    \mathbf{x}(s) & = \begin{bmatrix} x_{1}(s) \\ \vdots \\ x_{n}(s) \end{bmatrix} \\
    \end{align*} \]

We have \( \mathbf{G}, \mathbf{B}, \phi(s) = (s\mathbf{I} - \mathbf{A}) \)

\[ \begin{align*}
    \mathbf{y} & = \begin{bmatrix} \mathbf{y}_{1} \\ \vdots \\ \mathbf{y}_{n} \end{bmatrix} \\
    \mathbf{u}(s) & = \begin{bmatrix} u_{1}(s) \\ \vdots \\ u_{n}(s) \end{bmatrix} \\
    \mathbf{x}(s) & = \begin{bmatrix} x_{1}(s) \\ \vdots \\ x_{n}(s) \end{bmatrix} \\
    \end{align*} \]
Problems
- How do I put in reference commands?
  - have two controls, elevator - pitch
    flap rudder - flight path
- How do I command pitch, flight path angle?
- How good is my command following?

Based on classical ideas

\[ y = \text{output} \]

commanded output

must know what desired output is to evaluate system.

Remember, \( G_{2a} \) has nice properties.

not desired output,
same units as \( u_p \)
(maybe disturbance)

\[ G_{2a} = G \Phi B \] is nice

\[ S_{2a} = \left[ I + G_{2a} \right]^{-1} \] is nice

meaning,

\[ \sigma_{\text{max}} \left[ S_{2a}(j\omega) \right] \leq 1 \quad \forall \omega \]

\[ T_{2a} = I - S_{2a} = G_{2a} \left[ I + G_{2a} \right]^{-1} \]

\[ \sigma_{\text{max}} \left[ T_{2a}(j\omega) \right] \leq 2 \]
How do we make the above design method really practical?

Assume: 
\[ X_p = \begin{bmatrix} y \\ X_r \end{bmatrix} \text{ outputs} \quad P_8 = \begin{bmatrix} y \\ \theta \\ q \\ u \end{bmatrix} \]

Outputs can be made a subset of the state in many problems.

First: acknowledge state can be written in this way.

Second: Augment plant with integrators.

New state
\[ z_p = \begin{bmatrix} x_i \\ x_p \end{bmatrix} = \begin{bmatrix} x_i \\ y \\ x_r \end{bmatrix} \leftarrow \text{After augmenting state with integrators.} \]

\( x_i = \text{state of integrators} \)

\( x_p = \text{state of original plant including outputs + rest of state.} \)

\[ G = \begin{bmatrix} G_i & G_y & G_r \end{bmatrix} \]

\[ u = -Gx \]
\[ = -\begin{bmatrix} G_i & G_y & G_r \end{bmatrix} \begin{bmatrix} x_i \\ y \\ x_r \end{bmatrix} \]

\[ = -G_i x_i - G_y y - G_r x_r \]
How do we implement command following?

\[ u = -G_x x + G_y (r - y) - G_r x_r \]

\( r \) is the desired output which is compared with \( y \).

At this point, we have nice properties.

If we break the loop here and inject a signal, it will go all the way around and come back. The signal you get back will be

\( (G_u q) \times \) (Input signal)

Plot sensitivity \((1 + G_u q)^{-1}\) \(-\) will look nice.
Plot comp. sensitivity \(-\) will look nice.

\[ I - S_u q \]
\[ G_u q \times (I + G_u q)^{-1} \]
we always have unmodeled dynamics and uncertainties.

\[ \Delta \]

LQ method assures us we will have nice robustness properties to try to accommodate uncertainties at plant input.

Don't know how much uncertainty we can accommodate at plant output. Will address this next time.