Available sensors/actuators

- Star tracker
  - keep image of star centred on focal plane of a telescope
  - provides 'noisy' but accurate (on average) measure of $\dot{\theta}_2$
- Rate gyro
  - provides 'clean' measure of $\dot{\theta}_2$
- Cold-gas jets
  - provide fast, accurate control torque

Model

- Model as two inertias connected by flexible structure

Reference input $r = \text{desired } \theta_2$

Controller output $u$ is torque $T_c$

States: $x = [\theta_2 \ \dot{\theta}_2 \ \theta_1 \ \dot{\theta}_1]^T$

State-space equations of motion:

$$x = \begin{bmatrix}
0 & 1 & 0 & 0 \\
-k/J_2 & -b/J_2 & k/J_2 & b/J_2 \\
0 & 0 & 0 & 1 \\
k/J_1 & b/J_1 & -k/J_1 & -b/J_1
\end{bmatrix} x + \begin{bmatrix}
0 \\
0 \\
1/J_1
\end{bmatrix} u$$

$$y = [1 \ 0 \ 0 \ 0] x + [0 \ 0] u$$

Case study

- Design of a satellite attitude control system
  - see Franklin, Powell & Emami-Naeini, Ch. 9.2
- Requirements:
  - accurate pointing of scientific sensor package
  - package isolated from vibrations, etc. of main body of satellite
  - must respond to commanded change in pointing direction with $t_s \leq 20$ s, $M_p \leq 15\%$
**Model**

- **Transfer function**
  \[ G(s) = \frac{Y(s)}{U(s)} = \frac{10bs + 10k}{s^2(s^2 + 11bs + 11k)} \]
  \[ = \frac{K(s + \sigma)}{s^2(s^2 + 2\zeta\omega_n s + \omega_n^2)} \]

- **Parameters**
  - Inertias:
    \[ J_1 = 1, \quad J_2 = 0.1 \]
  - Temperature fluctuations cause:
    \[ 0.09 \leq k \leq 0.4 \]
  - hence, oscillatory mode characterised by:
    \[ \leq \omega_n \leq 2 \text{ rad/s} \]
    \[ 0.02 \leq \zeta \leq 0.1 \]

**Strategy**

- **Design for 'worst case' nominal plant**
  - slowest natural response \( (\omega_n = 1) \)
  - lightest damping \( (\zeta = 0.02) \)

- **Hence nominal plant**: \[ J_1 = 1, \quad J_2 = 0.1, \quad k = 0.091, \quad b = 0.0036 \]

\[
\begin{bmatrix}
0 & 1 & 0 & 0 \\
-0.91 & -0.036 & 0.91 & 0.036 \\
0 & 0 & 0 & 1 \\
0.091 & 0.0036 & -0.091 & -0.0036 \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
1/J_1 \\
\end{bmatrix}
\]

\[ G(s) = \frac{0.036(s + 25)}{s^2(s^2 + 0.04s + 1)} \]

- Check 'robustness' by calculating performance with other parameter combinations

**Pole placement design of control law**

- **Nominal plant** \[ G(s) = \frac{Y(s)}{U(s)} = \frac{0.036(s + 25)}{s^2(s + 0.02 + j1)} \]

- **Symmetric root locus**

\[
\begin{align*}
\text{axis}([-2 2 -2 2]) \\
\text{sgrid}
\end{align*}
\]

\[ % n - m = \text{odd no.} \]
\[ % \text{Hence draw } 0 \text{ deg locus} \]
\[ G = \text{ss}(A, B, C, D); \]
\[ [z, p, k] = \text{zpkdata}(G, 'v'); \]
\[ Z = [z; -z]; \]
\[ P = [p; -p]; \]
\[ GG = \text{zpk}(Z, P, -k*k) \]
\[ \text{rlocus(GG)} \]

Get 0° locus
Symmetric root locus

axis([-2 2 -2 2])
sgrid

Select poles with
\( \omega_n = 0.5 \text{ rad/s} \)

[\rho,\text{clp}]=rlocfind(GG)

Specs: \( t_f \leq 20 \text{ s}, M_p \leq 15\% \)
\( \Rightarrow \) dominant poles
\( \omega_n = 0.5 \text{ rad/s}, \zeta = 0.5 \)

State-space design examined ...

- We now have a control law, assuming full state feedback: \( u = -Kx \)
- Before proceeding, examine this state-space design in classical control terms
  - open-loop transfer function
  - open-loop frequency response
  - what is the effective compensation?
  - stability margins
  - root locus: sensitivity to variations in open-loop gain
  - closed-loop transfer function

Open-loop transfer function

% Get all states as plant outputs:
Ga = G; Ga.c = eye(4);
% Form open-loop TF:
Gol = ss(K)*Ga;
zpk(Gol)

Zero/pole/gain:
\[ 1.161 (s+0.3601) \left( s^2 + 0.2604s + 0.8698 \right) \]
\[ s^2 \left( s^2 + 0.04s + 1 \right) \]

\( r \equiv 0 \) + \u 

**Plant poles**

**Compensator** zeros

\( \Rightarrow K = [-0.28 0.05 0.68 1.16] \)
Effect of feedback on OL freq. resp.

Stability margins

Closed-loop transfer function
Robustness test

% Try same controller with stiff spring model
k = 4/11;  % corresponds to wn = 2
b = k/50;  % corresponds to zeta = 0.02
As = [0 1 0 0
   -k/J2 -b/J2 k/J2 b/J2
   0 0 0 1
   k/J1 b/J1 -k/J1 -b/J1];
Gs = ss(As, B, C, D);
disp('Stiff-spring plant transfer function')
Gzpk = zpk(Gs)
Zero/pole/gain:
0.072727 (s+50)
---------------------
s^2 (s^2 + 0.08s + 4)

Stiff-spring plant with nominal controller

Gsa = Gs;  Gsa.c = eye(4);
Gscl = feedback(Gsa, tf(K));
Gscl = Gscl(1,1);
disp('Stiff-spring closed-loop transfer function')
zpk(Gscl)
Zero/pole/gain:
0.072727 (s+50)
---------------------
(s^2 + 1.04s + 0.3455) (s^2 + 0.2015s + 4.21)

figure(3), step(Gcl, Gscl)
legend('nominal plant', 'stiff plant')
grid on

Step responses of nominal and stiff-spring plants with nominal controller

% F&P select CLPs corresponding to rho = 3.05e7
eclp = rlocus(GG,3.05e7);
% We want the stable ones
declp = eclp(find(real(eclp)<0))
% Plot them
rlocus(GG)
hold on
plot(real(dclp), imag(dclp), 'gs', ...  
'MarkerFaceColor','g')
plot(real(dclp), imag(dclp), 'rs', ...  
'MarkerFaceColor', 'r')
hold off
axis([-8 8 -8 8]), axis equal
sgrid

Estimator

M_p = 8%
t_c (1%) = 21 s

M_p = 3%
t_c (1%) = 20 s
Estimator and controller poles

Estimator poles chosen very fast (8 x controller poles)
- avoid compromising robustness
- fast estimator almost equivalent to direct state feedback
  (provided model is accurate and sensor noise is low!)

Frequency response of effective FB compensator

bode(-H)  % -H because u = -K\hat{x} built into H

Estimator gains; regulator TF

% State-estimator gains
L = place(A', C', dec1p)'
\Rightarrow L = \begin{bmatrix} 22 \\ 242 \\ 1513 \\ 5498 \end{bmatrix}

% Form regulator
H = reg(G, K, L);
disp('Regulator transfer function')
zpk(H)
Zero/pole/gain:
-7413.5284 (s+0.321) (s^2 + 0.1985s + 0.8439)
-------------------------------------------------
(s^2 + 16.96s + 85.08) (s^2 + 6.244s + 79.29)

Open-loop transfer function

Grol = -H*G;
zpk(Grol)
Zero/pole/gain:
\begin{bmatrix} 269.5829 (s+25) (s+0.321) (s^2 + 0.1985s + 0.8439) \\ s^2 (s^2 + 0.04s + 1) (s^2 + 16.96s + 85.08) (s^2 + 6.244s + 79.29) \end{bmatrix}

'Compensator' dynamics
pzmap(Grol)
### Stability margins

% OLF
\[ \text{Grol} = -H \ast G; \]
\[ \text{margin(Grol)} \]

\[ \omega_c = 1.3 \text{ rad/s} \]
\[ \text{GM} = 14 \text{ dB} \]
\[ \text{PM} \approx 46^\circ \]

### Closed-loop transfer function

% Form closed-loop system
\[ \text{Grcl} = \text{feedback}(G, H, +1); \]
% Show poles and zeros of CL system
\[ \text{pzmap(Grcl)} \]

### Closed-loop response of regulated system

\[ \text{step}(\text{Grcl}, \text{Gsrcl}) \]

\[ M_p = 8\% \]
\[ t_s (1\%) = 21 \text{ s} \]
\[ M_p = 7\% \]
\[ t_s (1\%) = 11 \text{ s} \]
Examine stiff-spring system more closely ...

- Sensitivity to variations in loop gain
  \[ Gsrol = -H*Gs; \]
  \[ rlocus(Gsrol) \]

poorly-damped mode (\( \zeta = 0.05 \)) could be excited by disturbance inputs

Lightly damped mode now dominate the response

\[ \omega_n = 5.2, \zeta = 0.05 \]

Response to impulsive disturbance

\[ x0 = zeros(8,1); x0(2) = 1; \% \text{set } ydot = 1 \]
\[ initial(Grcl,Gsrcrl,x0) \]

Tracking system

\[ \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]

\[ u = -K\hat{x} + (KN_x + N_u)r \]

- Implement with:
  - Simulink model, or
  - with MATLAB commands

MATLAB implementation

\[ \begin{bmatrix} N_u \\ N_x \end{bmatrix} \]
\[ \begin{bmatrix} K \end{bmatrix} \]
\[ \begin{bmatrix} G \end{bmatrix} \]
\[ \begin{bmatrix} Hest \end{bmatrix} \]
\[ \begin{bmatrix} G1; \; ss(Nx) \end{bmatrix} \]
\[ \begin{bmatrix} 1; \; ss(K) \end{bmatrix} \]
\[ \begin{bmatrix} 1; \; G \end{bmatrix} \]
\[ \begin{bmatrix} G3*G2; \end{bmatrix} \]
Closed-loop response of tracking system

% Step response
step(Gtcl)

dc gain = 1

nominal plant

stiff-spring plant

<table>
<thead>
<tr>
<th>Time (sec.)</th>
<th>Amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.2</td>
</tr>
<tr>
<td>10</td>
<td>0.4</td>
</tr>
<tr>
<td>15</td>
<td>0.6</td>
</tr>
<tr>
<td>20</td>
<td>0.8</td>
</tr>
<tr>
<td>25</td>
<td>1.0</td>
</tr>
</tbody>
</table>

% Step response
step(Gtcl)

Simulink implementation

sensors = [1]; % observed output = y1
known = [1];   % control input = u1
Hest = estim(G, L, sensors, known);
Hest = Hest(2:5,:); % suppress y^ output

plot(tout,xout(:,1:4)-xout(:,5:8))

State estimation errors $e = x - \hat{x}$