04/16/25.

* Optimize the tracking in state feedback: K_r and Integral.

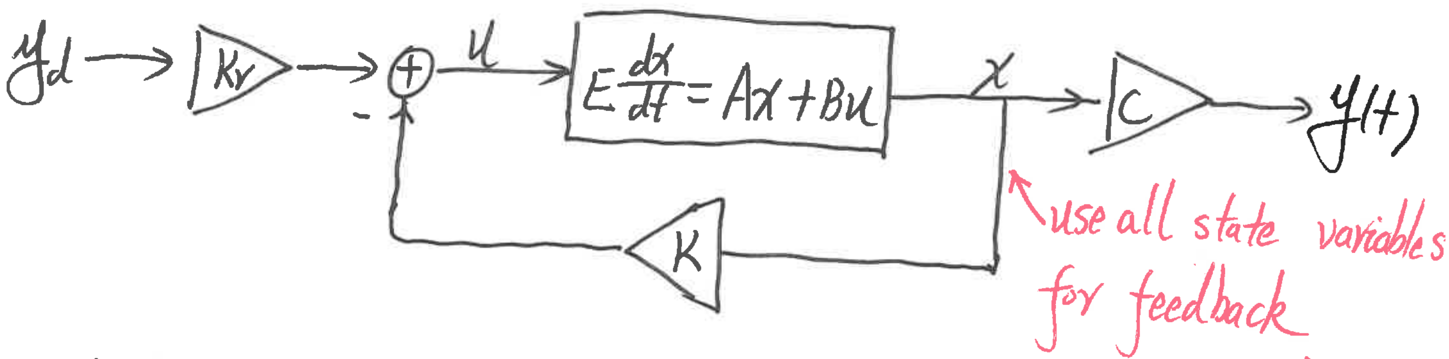
1. Review of State Feedback.

State space model: $E\frac{dx}{dt} = Ax + Bu$

output: $y = Cx$

control: $u = K_r y_d - Kx$.

Block Diagram:



Natural Response: stability.

close-loop state space model: substitute $u(t)$.

$$\frac{d}{dt}x = (A - BK)x + BK_r y_d.$$

(assume $E = I$, identity).

Natural frequency: $\lambda = \text{eig}(A - BK)$, i.e. solve $|sI - (A - BK)| = 0$.

Want all the λ's to be on the left half plane.

Model system: Given $K = [K_1, K_2 \cdots K_n] \Longrightarrow$ calculate $\lambda_{1,2,3\cdots}$

Design controller: find corresponding K. $\Longleftarrow$ choose $\lambda_{1,2,3\cdots}$

usually get a nth order characteristic equation from $|sI-(A-BK)|=0$.

$(s-\lambda_1)(s-\lambda_2)\cdots(s-\lambda_n)=0 \Longrightarrow$ expand to get $K_{1,2,\ldots n}$.

Using matlab / python, simply call function:

$$K = \text{place}\,(\underbrace{A,\ B}_{\text{define system}},\ \underbrace{[\lambda_1, \lambda_2 \cdots]}_{\text{specify pole location}})$$

See Propeller Arm Example in slides.

In the 1st example we see the system converges fast to a stable state, however, the final value of θ is not the same as $\theta_d(t=\infty)=1$.

2. Steady State of step response.

$$\frac{d}{dt}x = (A-BK)x + BK_r y_d.$$

Recall in D.T. for $n=\infty$, $x[n-1]=x[n]=x[\infty]$.

For C.T.: $x(\infty) = \text{const}$, so $\frac{d}{dt}x(\infty)=0$.

$$0 = (A-BK)x(\infty) + BK_r y_d.$$

$$x(\infty) = -(A-BK)^{-1}BK_r y_d.$$ (note order of matrix multiplication!)

Output: $y(\infty) = C x(\infty) = \underbrace{-C(A-BK)^{-1}BK_r}_{\text{want to be } 1} y_{d_0}$ (scalar).

For good tracking performance, want $y(\infty) = y_{d_0}$

$$\boxed{K_r = -\frac{1}{C(A-BK)^{-1}B}}$$

See example 2&3 from slides where K_r is fixed.

Last lecture, also discussed another way to determine K:

Linear Quadratic Regulator:

Ideally want all λ's to be very negative, however limited by actuator.

Define cost function: $Z = \int_0^\infty \underbrace{\sum_i q_i x_i^2(t)}_{\text{penalty on bad tracking}} + \underbrace{r u^2(t)}_{\text{penalty on actuation}} \, dt$.

Would like to $\min\limits_{K's} Z(K)$ Have the freedom to assign weight factor.

Matlab: $K = lqr(A, B, Q, R)$.

See example 4 from slides.

---

3. Adding Integral term as an additional state:

In D.T. (Lab 2B), we learned using PID control to improve tracking.

To model, we defined $sum[n] = sum[n-1] + err[n-1] \cdot \Delta T$.

Using Integral for feedback is advantageous as K_r relies on A, B, i.e. modelling accuracy.

Define a new state: $J(t) = \int_0^t (y(\tau) - y_d(\tau))\, d\tau$.

$$\text{or } \frac{d}{dt} J(t) = y(t) - y_d(t).$$

Add as the 1st state the state space model:

$$\frac{d}{dt} J(t) = \underbrace{C x(t)}_{y(t)} - y_d. \quad \updownarrow 1$$

$$\frac{d}{dt} x(t) = A x + B u \quad \updownarrow N$$

See slides for Example 5:

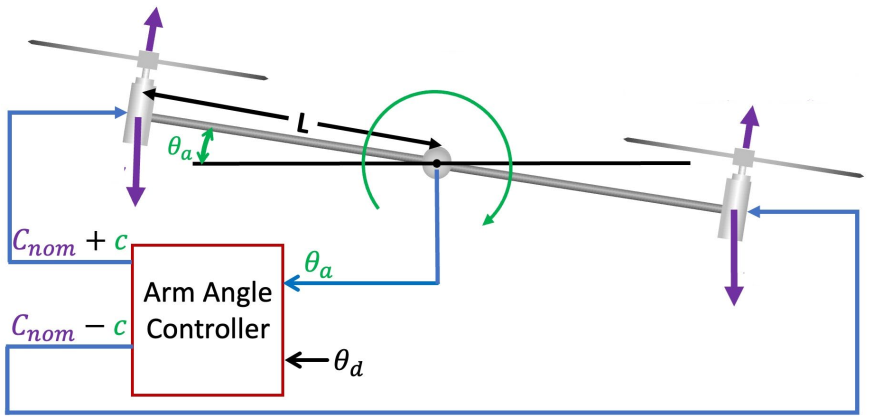
Define "Augmented" matrices:

$$A_+ = \begin{pmatrix} 0 & C \\ 0 & A \end{pmatrix} \quad B_+ = \begin{pmatrix} 0 \\ B \end{pmatrix}, \quad C_+ = \begin{pmatrix} 0 & C \end{pmatrix}.$$

Can use place( ) or lqr( ) to determine K's as before.

Note $Q_+ = (Q_{+1}, Q_{+2}\ Q_{+3}\ Q_{+4})$ has four weights.

Propeller Arm example

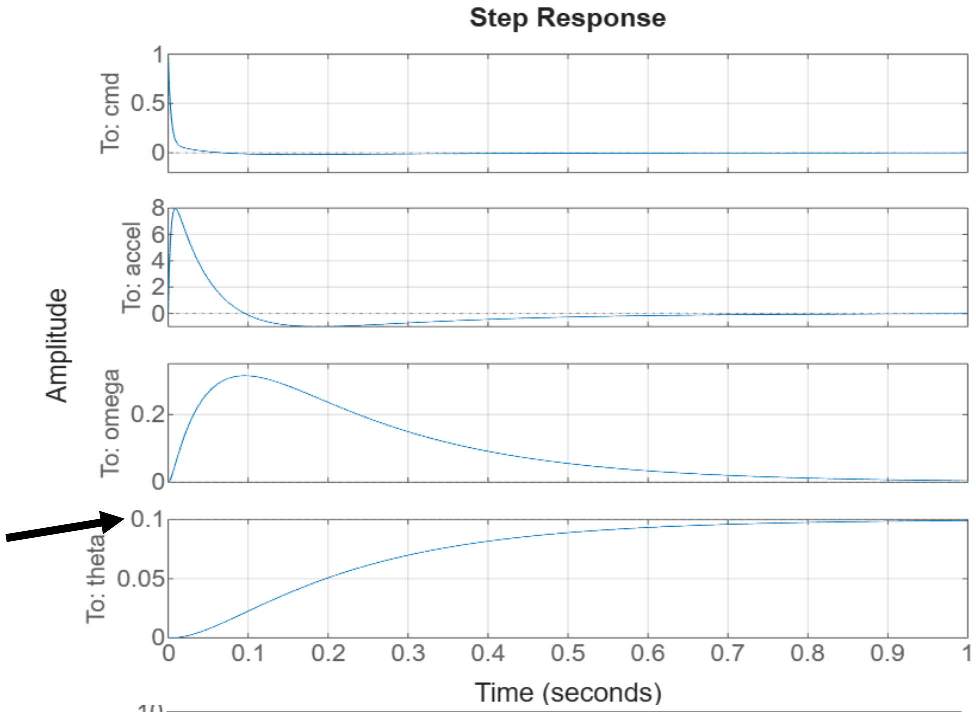


$$\frac{d}{dt}\begin{pmatrix}\alpha\\ \omega\\ \theta\end{pmatrix} = \begin{pmatrix}-\beta & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\end{pmatrix}\begin{pmatrix}\alpha\\ \omega\\ \theta\end{pmatrix} + \begin{pmatrix}\gamma\\ 0\\ 0\end{pmatrix}u(t) \qquad y(t) = \begin{pmatrix}0 & 0 & 1\end{pmatrix}\begin{pmatrix}\alpha\\ \omega\\ \theta\end{pmatrix} \qquad u(t) = K_r\theta_d - \begin{pmatrix}K_1 & K_2 & K_3\end{pmatrix}\begin{pmatrix}\alpha\\ \omega\\ \theta\end{pmatrix}$$

$$x = \begin{pmatrix}\alpha\\ \omega\\ \theta\end{pmatrix} \qquad A = \begin{pmatrix}-\beta & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\end{pmatrix} \qquad B = \begin{pmatrix}\gamma\\ 0\\ 0\end{pmatrix} \qquad C = \begin{pmatrix}0 & 0 & 1\end{pmatrix} \qquad K = \begin{pmatrix}K_1 & K_2 & K_3\end{pmatrix}$$

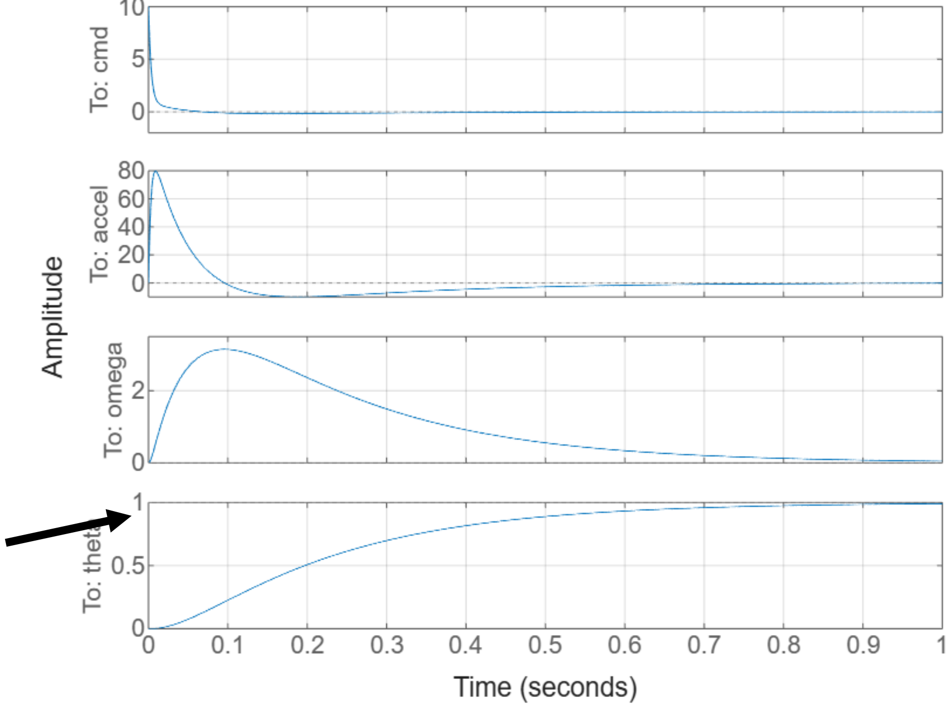
Example 1: Pole Placement

```
% Define all the matrices, using beta and gamma
A(1,1) = -50;
A(1,2) = 0.0;
A(1,3) = 0.0;
B(1) = 3000;
…………….

% Convert to a system data structure
sys = ss(A, B, C, D, 'StateName', {s1,s2,s3});

% Select closed loop poles for pole placement
design
poles = [-5, -20, -300];
Kplace = place(sys.A,sys.B,poles);


Kr = 1;
```



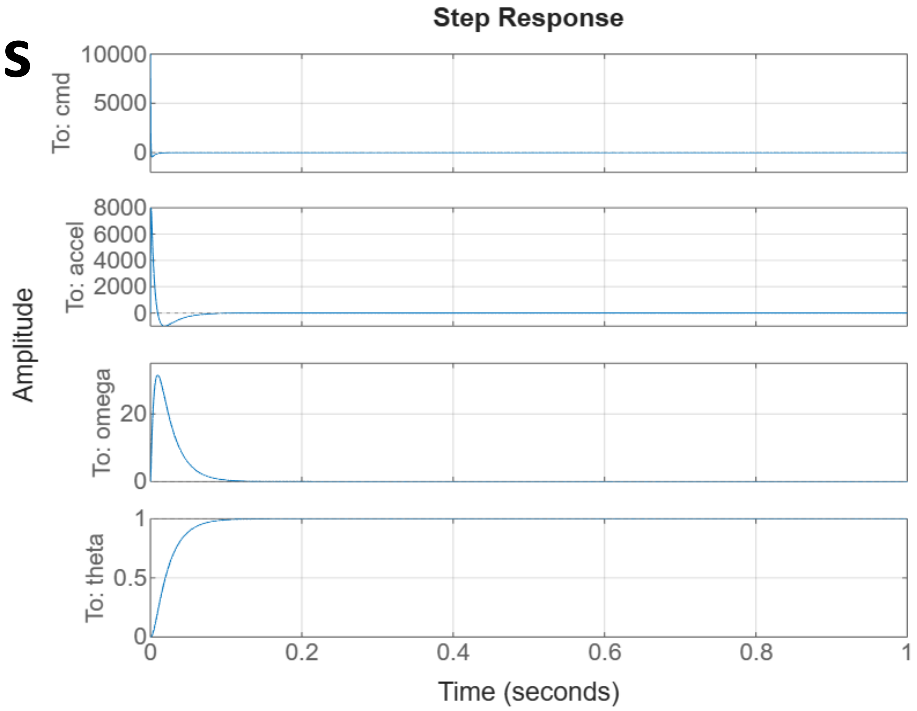
Example 2: Fix Kr

```
% Use the Correct Kr
Kr = -1/(C*(A-B*K)^(-1)*B);
```



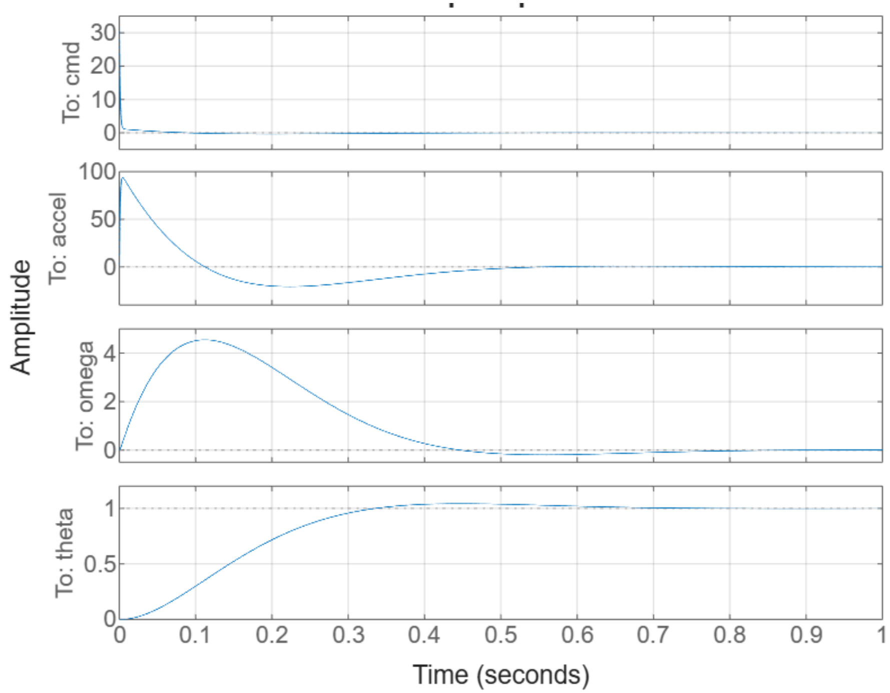
Example 3: Place at a different set of poles

```
% Try to place the poles at different
locations
poles = [-50, -200, -3000];
Kplace = place(sys.A,sys.B,poles);
```



Example 4: LQR

```
% Select state weights Q, and control weights R

Q = diag([0.1,0.1,1000]);

% Metric: Q(1,1)*x1^2+Q(2,2)*x2^2+Q(3,3)*x3^2 +
R*u^2
 R = 1;
 Klqr = lqr(sys,Q,R);
```



Example 5: Adding Integral Term

$$\frac{d}{dt}\begin{pmatrix} J \\ x \end{pmatrix} = \begin{pmatrix} 0 & C \\ 0 & A \end{pmatrix}\begin{pmatrix} J \\ x \end{pmatrix} + \begin{pmatrix} 0 \\ B \end{pmatrix} u(t) + \begin{pmatrix} -1 \\ 0 \end{pmatrix} y_d$$

$$y(t) = \begin{pmatrix} 0 & C \end{pmatrix}\begin{pmatrix} J \\ x \end{pmatrix} \qquad u(t) = -\begin{pmatrix} K_1 & K_2 & K_3 & K_4 \end{pmatrix}\begin{pmatrix} J \\ \alpha \\ \omega \\ \theta \end{pmatrix}$$

%no $K_r\theta_d$, why?? (Recall your Postlab2)

```
% to add the integrator part to the
controller, we augment the original
% equations by the "integrator state" J
% dJ/dr = y - y_d .

……
Ap = zeros(4);
Ap(2:4,:) = [zeros(3,1) A];
……….

% define the new system with Ap, Bp, Cp and Dp
sysp = ss(Ap, Bp, Cp, Dp, 'StateName',{'Int',
s1, s2, s3});

% State weights
Qp = diag([1000000,0.1,0.1,1000]);
Rp = diag(1);
Kp = lqr(sysp,Qp,Rp);
```