Unit 6 Notes State Space Design

This last note is shortened due to the current circumstances. But we need to include some of the material to wrap up the course.

We have looked at state space in a way of having a systematic approach to solving/analyzing complex systems.

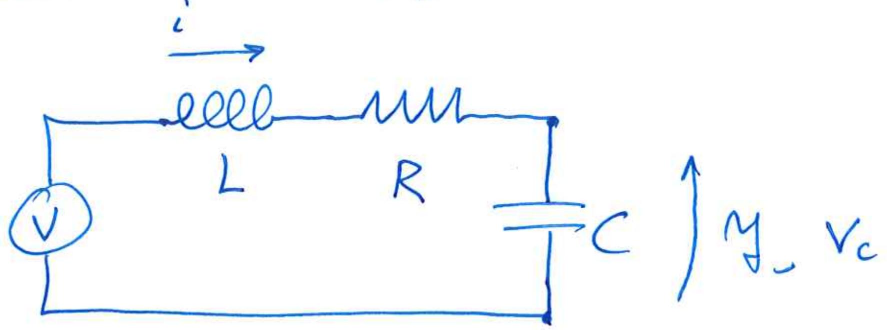
We wound up with the state space matrices as $x =$ vector of state variables

$$\dot{x} = \underbrace{A}_{\text{System matrix}}\ \underbrace{x}_{\text{state varibles}} + \underbrace{B}_{\text{input matrix}}\ \underbrace{u}_{\text{input}}$$

$$y = Cx + Du$$

State space system $\{A, B, C, D\}$

As a working example consider the second order system below:



State variables $x = \begin{bmatrix} i \\ V_c \end{bmatrix}$

$$L \frac{di}{dt} = V - Ri - V_c$$

$$C \frac{dv_c}{dt} = i$$

$$\begin{bmatrix} di/dt \\ dv_c/dt \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} i \\ V_c \end{bmatrix} + \begin{bmatrix} 1/L \\ 0 \end{bmatrix} V$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ V_c \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} V$$

Hence the state space system is

$\{A, B, C, D\}$

enter this into Matlab using H= ss(A, B, C, D)

We can get the step( ), impulse( ), bode( ) response from these as we can with a system generated from tf( ).

How is the transfer function generated from state space?

$$\dot{x} = Ax + Bv$$

$$y = Cx + Dv$$

$\begin{cases} x \text{ - state variable vector} \\ y \text{ - output} \\ v \text{ - input} \end{cases}$

Take Laplace transform

$$sIx = Ax + Bv$$

I is identity matrix

$$(sI - A)x = Bv$$

$$x = (sI-A)^{-1} Bv$$

$$y = \underbrace{\left(C(sI-A)^{-1}B + D\right)}_{H(s) \text{ transfer function}} v$$

Time Domain Solution

Solution of first order equation

$$\dot{x} = ax$$

x - single variable

a - scalar

homogeneous solution $x = g e^{at}$

g - constant.

$$\dot{x} = g a e^{at} = a\left(g e^{at}\right) = ax$$

hence $x = g e^{at}$ general homogeneous solution.

I.C. $x(0) = x_0 \Rightarrow x = x_0 e^{at}$

Consider exponential solution

$$x = g e^{at} = g\left(1 + at + \frac{a^2t^2}{2} + \frac{a^3t^3}{3!} + \cdots\right)$$

$$\frac{dx}{dt} = g\left(a + a^2 t + \frac{a^3 t^2}{2} + \cdots\right)$$

$$\frac{dx}{dt} = a\left(g\left(1 + at + \frac{a^2t^2}{2} + \dots\right)\right)$$

$$= a g e^{at} = a x$$

Now consider the solution to

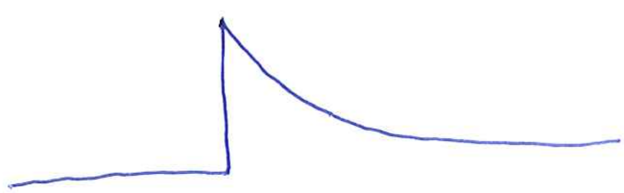
$$\frac{dx}{dt} = ax + b\,\delta(t) \qquad x(0^-) = 0$$

I.C. before input.

solution is $x(t) = be^{at}\,1(t)$

where $1(t)$ is unit step $\quad 1(t) = \begin{cases} 0 & t < 0 \\ 1 & t > 0 \end{cases}$

Show this



$$\frac{dx}{dt} = be^{at}\frac{d\,1(t)}{dt} + \frac{d\,be^{at}}{dt}\,1(t)$$

$$\frac{dx}{dt} = be^{at}\,\delta(t) + bae^{at}\,1(t)$$

$$\frac{dx}{dt} = a\,x(t) + b\,\delta(t)$$

Which is what we started with.

impulse response is the $h(t) = b\,e^{at}\,1(t)$

and for general input $u(t)$ ie $\frac{dx}{dt} = ax + bu(t)$

$$x(t) = h(t) * u(t)$$

$$x(t) = \int h(t-\tau)\,u(\tau)\,d\tau$$

$$= \int b e^{a(t-\tau)}\,u(\tau)\,d\tau$$

<u>Now consider the state space</u>

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

Consider homogeneous solution first

$$\dot{x} = Ax$$

Postulate solution

$$X = g\left(I + tA + \frac{t^2}{2}A^2 + \frac{t^3}{3!}A^3 + \cdots\right)$$

identity matrix

scalar

$A, A^2, A^3 \ldots$ square matrices.

$$\frac{dX}{dt} = g\left(0 + A + tA^2 + \frac{t^2}{2}A^3 + \cdots\right)$$

$$\frac{dx}{dt} = gA\left(I + tA + \frac{t^2}{2}A^2 + \cdots\right)$$

$$= \underbrace{Ag}\left(I + tA + \frac{t^2}{2}A^3 + \cdots\right)$$

only possible if

$Ag = gA$ $\quad$ g - scalar or $\begin{bmatrix} g & & \\ & \ddots & \\ & & g \end{bmatrix}$

$$\frac{dx}{dt} = AX$$

Hence $X = g\left(I + tA + \frac{t^2}{2}A^3 + \cdots\right)$

is homogeneous solution

define $e^{At} \equiv I + tA + \frac{t^2}{2} A^3 + \dots$

$X = g e^{At}$ homogeneous solution to state space problem.

Impulse response (single input $u(t) = \delta(t)$)

$$\frac{d}{dt} x = Ax + Bu$$

$$h(t) = B e^{At}$$

General solution in time domain

$$x(t) = h(t) * u(t)$$

$$= \int B e^{A(t-\tau)} u(\tau) \, d\tau$$

Frequency Domain solution (Poles)

$$\frac{dx}{dt} = Ax + Bu$$

$$y = Cx + Du$$

$x(t) \Leftrightarrow X(s)$

$y(t) \Leftrightarrow Y(s)$

$u(t) \Leftrightarrow U(s)$

$h(t) \Leftrightarrow H(s)$

$$H(s) = C(sI - A)^{-1} B$$

$$H(s) = C\,\mathcal{L}(h(t)) = CB\,\mathcal{L}(e^{At})$$

$$H(s) = C \frac{\text{Adjunct}(sI-A)}{\det(sI-A)} B$$

$C \text{ Adjunct}(sJ-A) B \Rightarrow$ some polynomial in s of order $N-1$

not really importance in present context.

$\det(sI-A) \rightarrow$ characteristic polynomial in s of order N

example $A = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$

$$sI - A = \begin{bmatrix} s-1 & -1 \\ 0 & s-2 \end{bmatrix}$$

$$|sI - A| = (s-1)(s-2)$$

1. roots of $\det(sI-A)$ are eigen values of A.
2. root of $\det(sI-A)$ are poles of $H(s)$

Proof (only if you are interested)

$A = V \Lambda V^{-1}$ Similarity transformation

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix}$$ diagonal matrix of eigenvalues

$V =$ matrix $N \times N$ of N eigenvectors

$$\det(sI - A) = \det(sVV^{-1} - V\Lambda V^{-1})$$

$$= \det\left(V\left(sI - \Lambda\right)V^{-1}\right)$$

$$= \det(V)\det(V^{-1})\det(sI - \Lambda)$$

$$= \prod_{n=1}^{N} (s - \lambda_n)$$

ie $\det(sI - A) = \prod_{n=1}^{N} (s - \lambda_n)$

$\therefore$ λ_n poles of transfer function

determine in matlab with eig(A)

if real part of all $\lambda_1, \lambda_2 \ldots \lambda_N$ are negative ie $Re(\lambda_n) < 0$ for $n = 1, \ldots N$

then system is stable otherwise unstable.

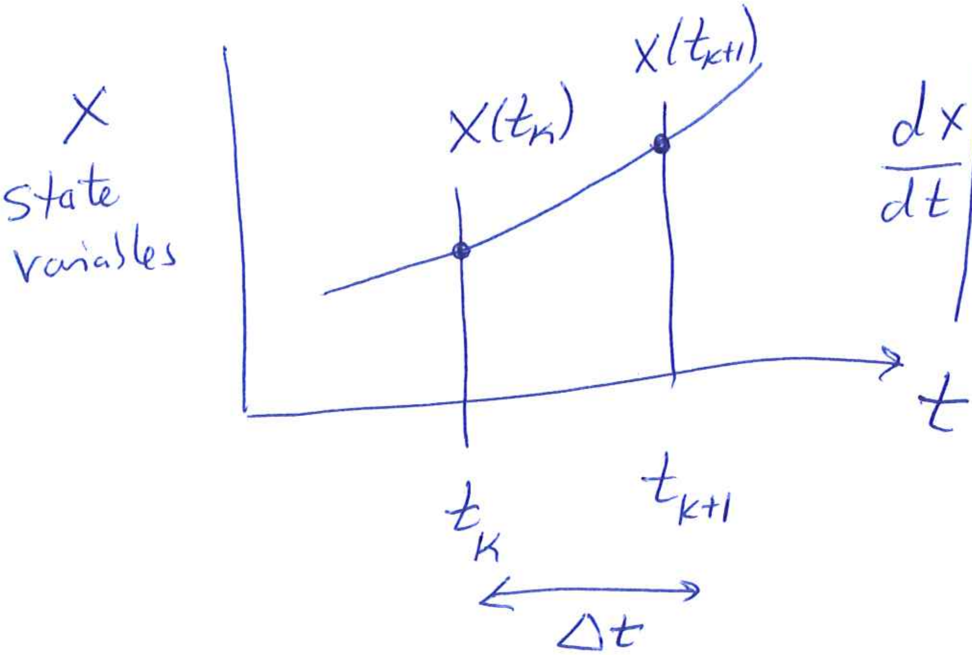
Transient Response of State space System

$$x(t) = \int B e^{A(t-\tau)} u(\tau)\, d\tau$$

can lead to closed form solution however usually this is not the case.

Solve numerically

$$\frac{dx(t)}{dt} = A\,x(t) + B\,u(t)$$



$$\left.\frac{dx}{dt}\right|_{t=t_k} \approx \frac{x(t_{k+1}) - x(t_k)}{\Delta t}$$

$$\frac{x(t_{k+1}) - x(t_k)}{\Delta t} \approx A x(t_k) + B u(t_k)$$

$$x(t_{k+1}) \approx x(t_k) + \Delta t \left(A x(t_k) + B u(t_k) \right)$$

$$x(t_{k+1}) \approx (I + \Delta t A)\, x(t_k) + \Delta t B\, u(t_k)$$

recursive equation for determining $x(t)$ approximately.

Use Matlab <u>ode45( )</u> for general DEQ solution.

<u>Better use lsim( )</u>

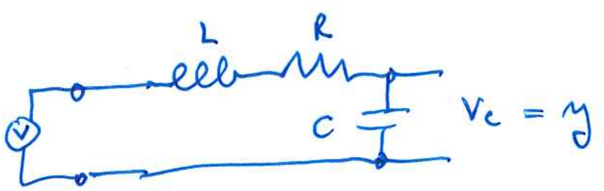
Multi-sensor feedback State feedback

A really useful application of state space is state feedback which is far more effective in placing poles of the closed loop system where desired.

Consider the LRC circuit of pg. 2-3 where

$$X = \begin{bmatrix} i \\ V_c \end{bmatrix}, \quad A = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1/L \\ 0 \end{bmatrix}$$

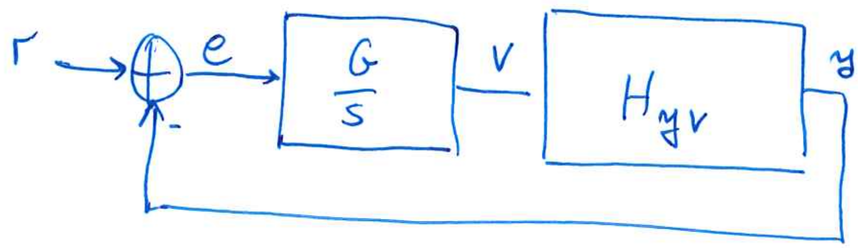
$$C = \begin{bmatrix} 0 & 1 \end{bmatrix} \qquad D = [0]$$



The transfer function from v to $V_c = y$ can be determined from

$$H_{yv} = C(SI - A)^{-1}B$$

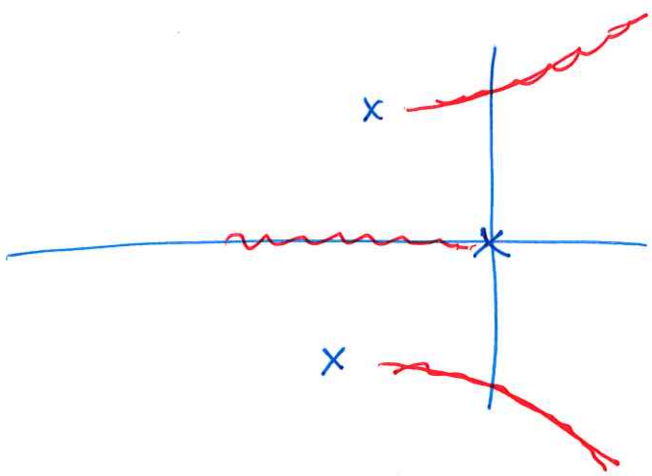
Suppose we wanted a type-1 loop where H_{yv} is the plant.



Example with $L = 0,1$, $C = 0,1$, $R = 1$ $\Bigg\}$ $A = \begin{bmatrix} -10 & -10 \\ 10 & 0 \end{bmatrix}$

$eig(A) \Rightarrow$ $\left.\begin{matrix} -5 + j\,8,66 \\ -5 - j\,8,66 \end{matrix}\right\}$ Two complex poles

Root Locus



Difficult to get an acceptable compensator even if we can add a number of lead circuits.

Instead put into state space and then feed all state variables back instead of just y.

We can show that closed loop poles can be placed anywhere!

We do this in three steps:

Step 1 Combine $\frac{1}{s}$ with H_{gv} into a state space system H_m

$$H_m = ss(A_m, B_m, C_m, D_m)$$

$$A_m = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{A} & \begin{matrix} 1/L \\ 0 \end{matrix} \\ \hline 0 & 0 & 0 \end{array}\right]$$ Augmentation adds in the integrator at the input

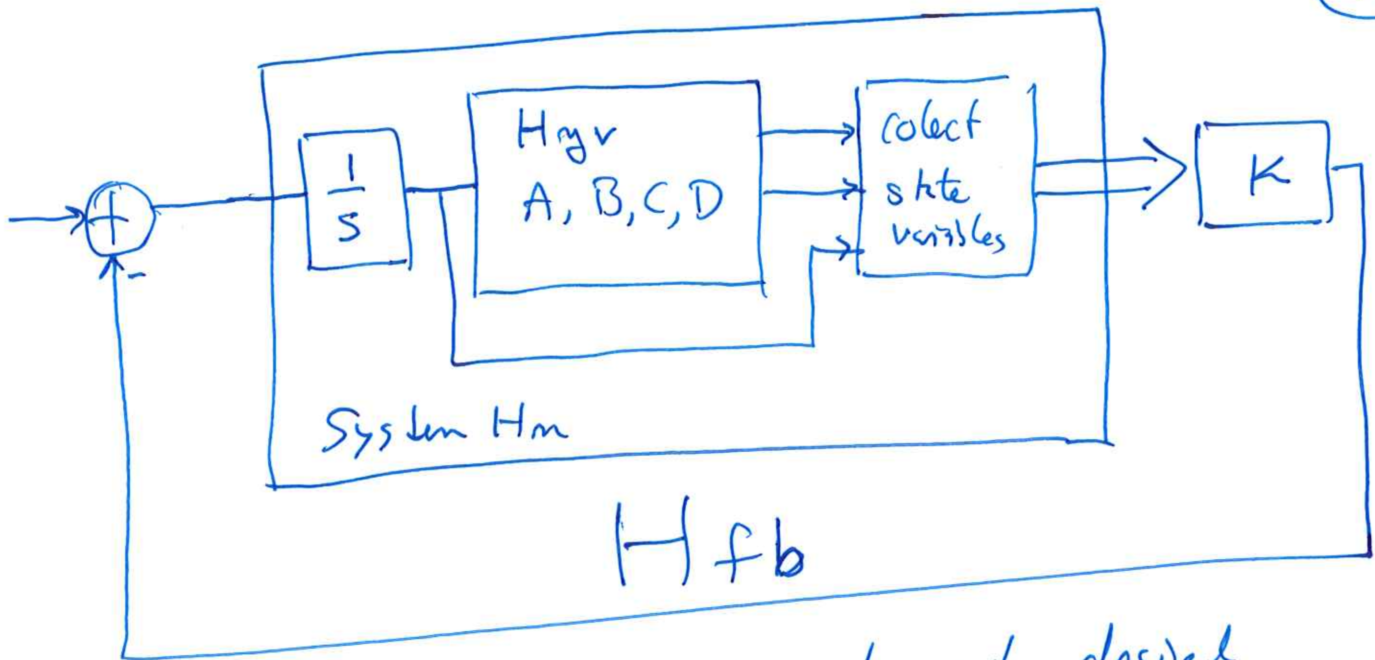
$$B_m = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$ Input goes into the integrator.

$$C_m = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

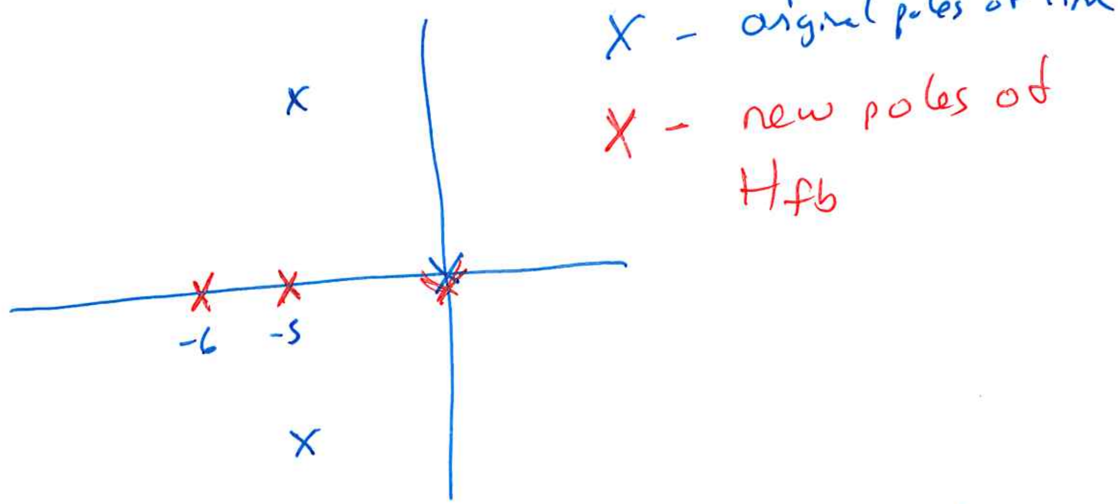
$$D_m = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

we want all three state variables as outputs

Step 2 Determine the feed back vector around H_m that will place the poles in the desired location



feedback of K places the poles at desired locations



Easier to form a feedback based on system Hfb than system Hm.

<u>Step 3</u> $H_{fb} = \text{feedback}(H_p, k)$

$$K = \text{place}(A_m, B_m \underbrace{[0, -5, -6]}_{\text{set of desired poles}})$$

complicated Matlab routine

Now we can use simple proportional feedback around system Hfb. You can do this with siso tool.

Steps

```
K = place(Am, Bm, [0, -5, -6])
Hp = ss(Am, Bm, eye(3), [0; 0; 0])
Hfb = feedback(Hp, K)
Hh = series(Hfb, [0, 1, 0])
```

Go to sisotool and see that a proportional feedback gets good results.