Full State Feedback

NDSU ECE 463/663

Lecture #12

Inst: Jake Glower

Please visit Bison Academy for corresponding
lecture notes, homework sets, and solutions

Feedback

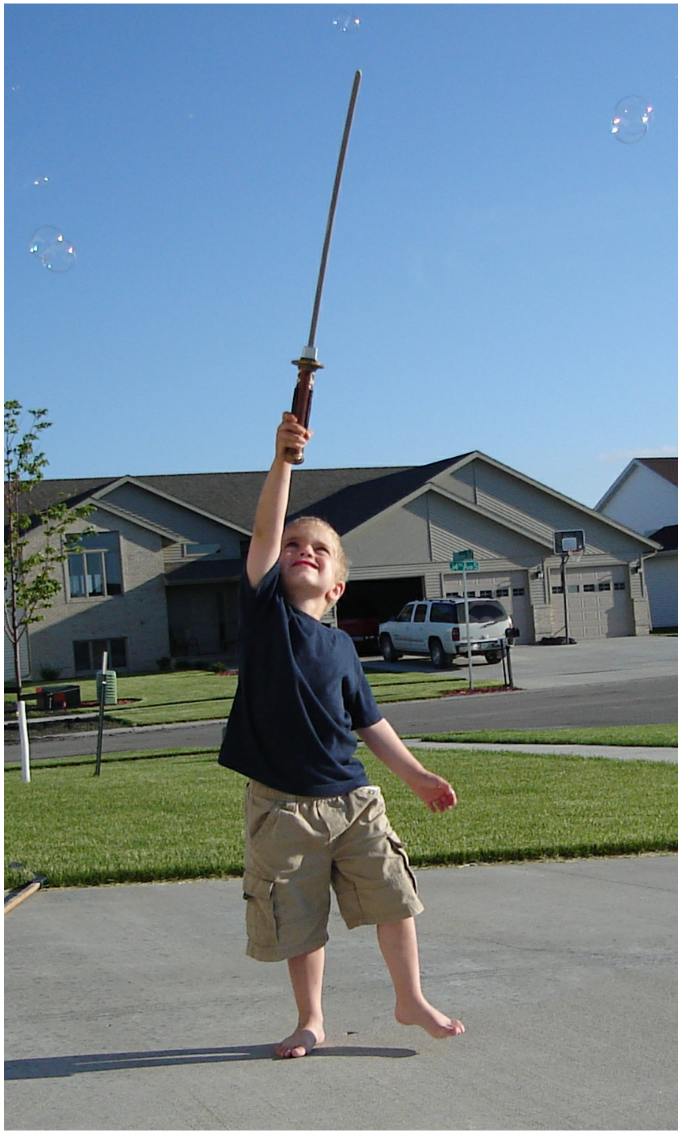
A system's dynamics determine how the system behaves.

Feedback is a tool which allows you to change the dynamics of a system.

For example, both walking and riding a bike are unstable without feedback.

- With practice, you learn to stand and walk
- With practice, you learn how to ride a bike.

As you practice, you figuring out how to adjust the input based upon the output



The Importance of Feedback

csinvesting.org/wp-content/uploads/2015/05/Boom-and-Bust1.png

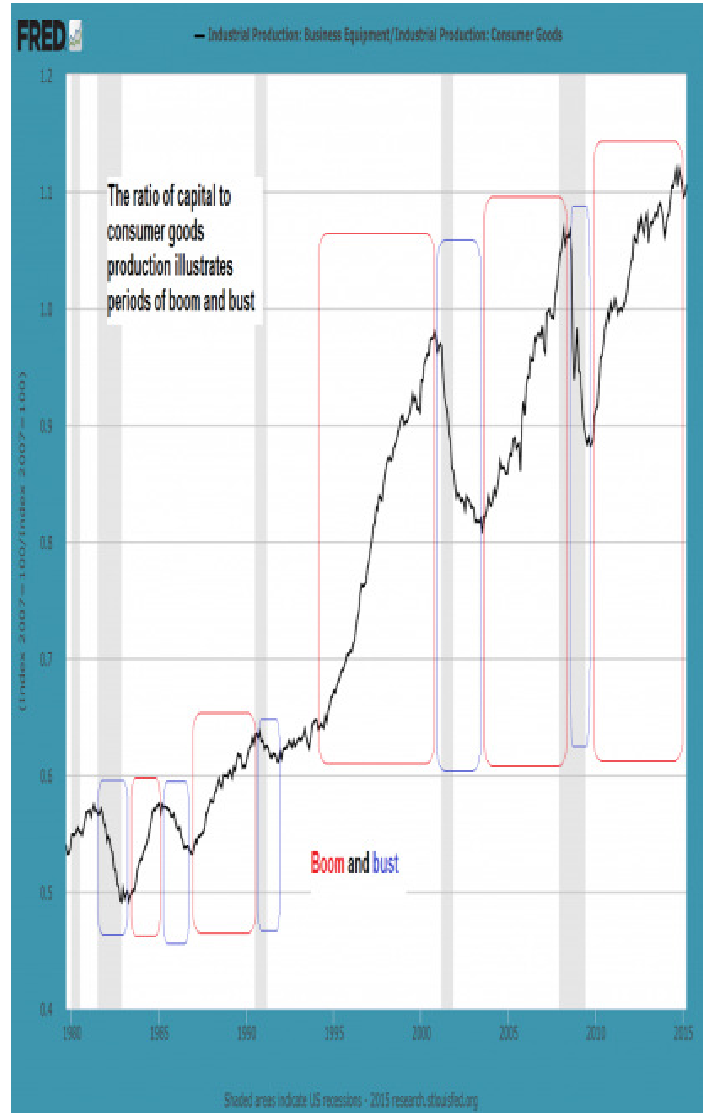
- Many systems are open-loop unstable
- Feedback is what makes them work

Economics

- Boom & bust cycles date back to the Roman Empire
- During good times, people buy more, companies sell more, companies hire more people, people buy more, etc.
- During bad times, people buy less, companies start laying off people, people buy even less

Federal Reserve

- Provides feedback to keep the growth rate at 3%
- Money supply, interest rates are the control inputs



Planetary Weather

www.digitalartsonline.co.uk

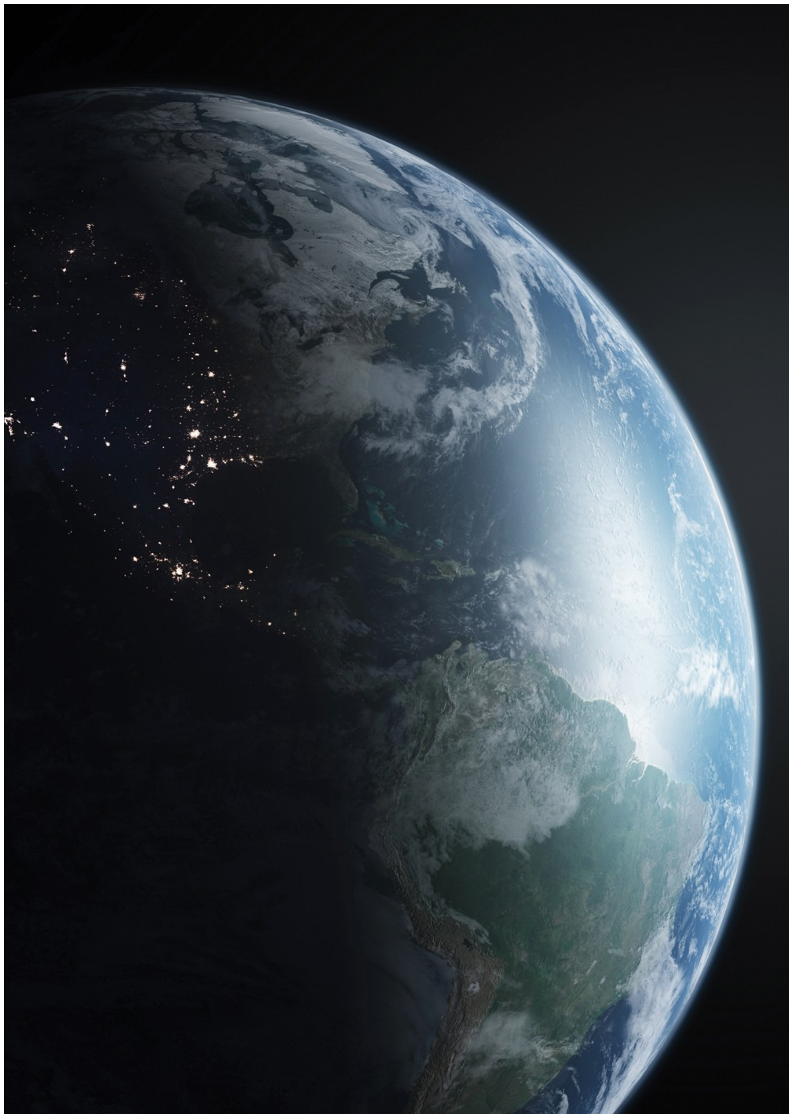
Cooling Cycle

- As the planet cools, more snow accumulates
- More snow reflects more sunlight, cooling the planet further
- Can (and did) produce a runaway ice age

Warming Cycle

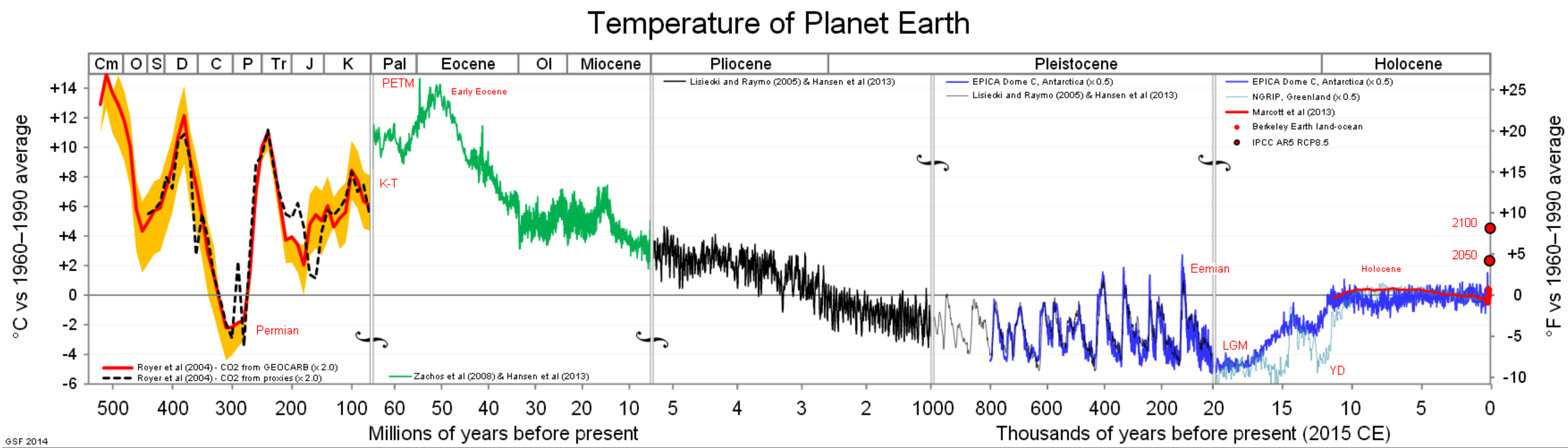
- Warmer weather melts ice
- Less ice means more sunlight is absorbed
- Which warms the planet further
- Can (and did) produce runaway heating

One thought is that life provides the feedback meachanism to stabilize the climate



Sidelight: Why did civilization take off 10,000 years ago?

- The last 10,000 years have been unusually consistent
- Dogs were domesticated 10,000 years ago
- Coincidence?



Glen Fergus https://commons.wikimedia.org/w/index.php?curid=31736468

Walking & Running

https://www.animalsandenglish.com/beetles-bugs--insects.html
funnypicture.org/wallpaper/2015/05/funny-cat-running-32-desktop-background.jpg

Crawling is open-loop stable

- 3 feet on the ground at all times
- gaits used by insects
- gait used by animals at low speed

Faster gaits are open-loop unstable

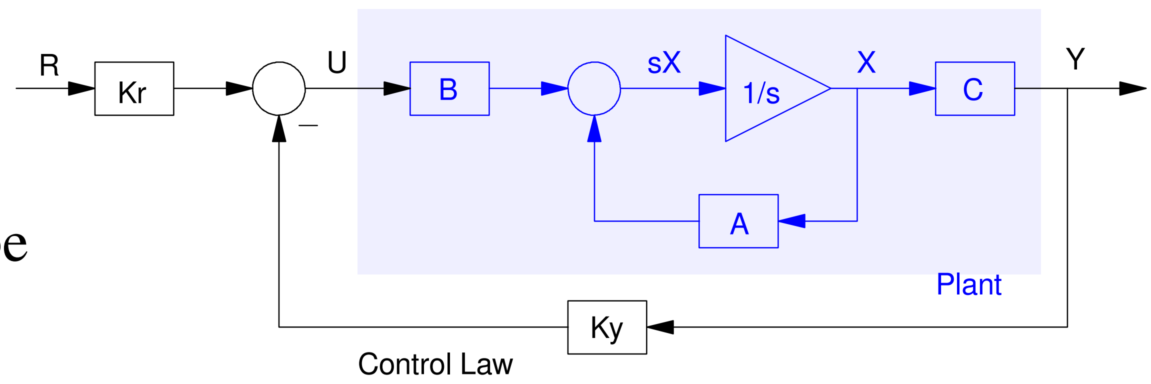
- Trot, Pace, Gallop, Bound
- Evolutionary advantage

Output Feedback

Assume you have a system

$$sX = AX + BU$$

$$Y = CX$$



If you define the input, U, to be

$$U = K_r R - K_y Y$$

then the dynamics become

$$sX = (A - BK_y C)X + BK_r R$$

$$Y = CX$$

The eigenvalues of (A - BKyC) define the closed-loop systems dynamics

- With 1 degree of freedom (Ky), the roots follow a 1-dimensional path
- Termed 'the root locus' in ECE 461: Controls Systems

Example: 4-stage RC filter

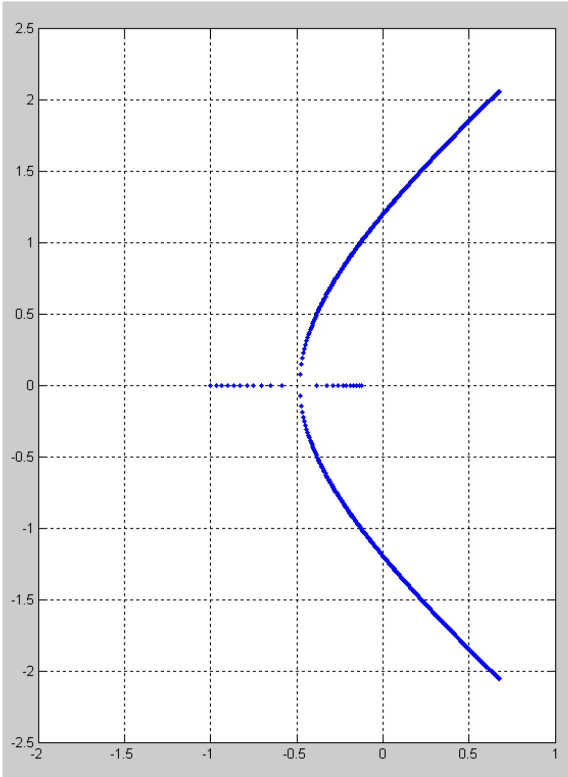
- Or heat flow in a 1-dimensional metal rod

$$sX = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} X + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

$$Y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} X$$

Plot the roots of (A - BKyY) for 0 < Ky < 100

```
A = [-2,1,0,0;1,-2,1,0;0,1,-2,1;0,0,1,-1]
B = [1;0;0;0]
C = [0,0,0,1]
Ky = [0:0.1:100]';
R = [];
for i=1:length(Ky)
   R = [R; eig(A - B*Ky(i)*C)'];
   end
plot(real(R),imag(R),'b.');
```



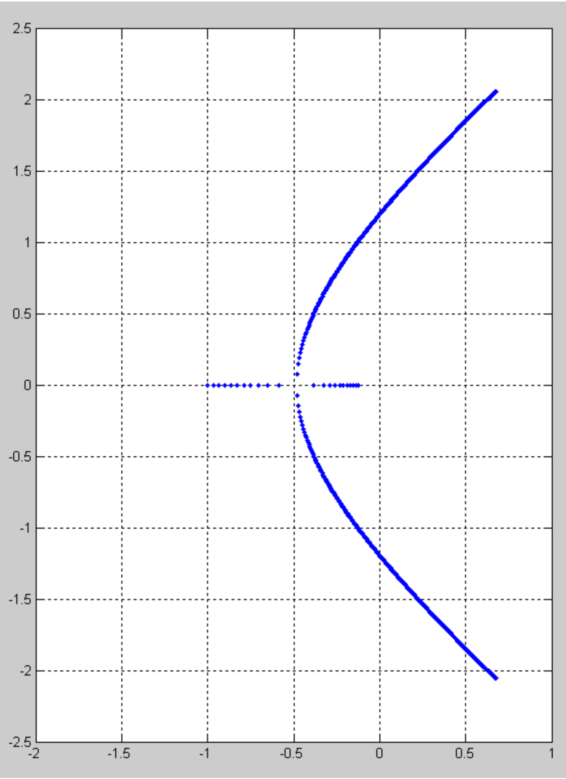
Note

- When Ky = 0, the roots are the eigenvalues of A

As Ky increases, the roots shift

- Initially, the system speeds up
- Then the poles become complex,
- Then they go unstable.

Ky	0	0.1	1	10
poles	- 3.532	- 3.522	- 3.414	- 3.338 + j0.882
	- 2.347	- 2.375	- 2.618	- 3.338 - j0.882
	- 1.	- 0.966	- 0.585	- 0.161 + j0.946
	- 0.120	- 0.136	- 0.381	- 0.161 - j0.946



What is the "best" feedback gain?

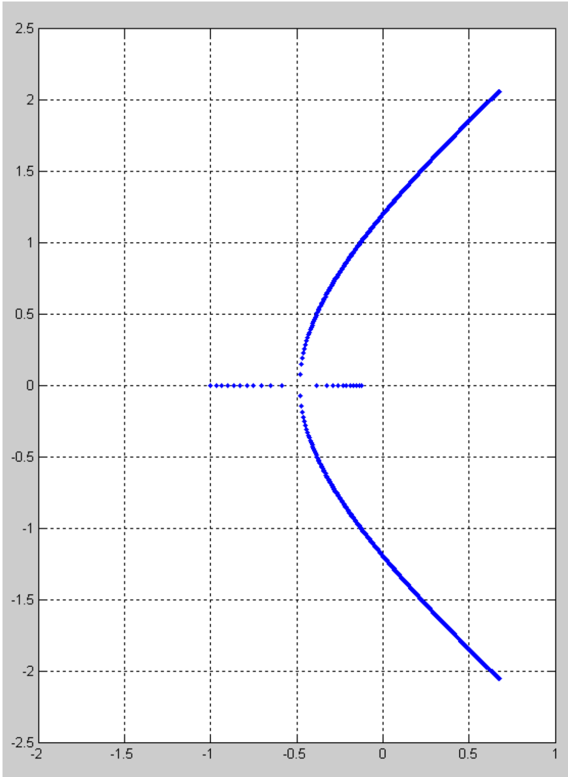
- Topic of ECE 461 Classical Controls

Depends upon what you mean by "best"

- High gains are good
  - *Faster response*
  - *Better tracking*
- Too much gain produces too much overshoot

The root locus plot gives you a shopping list

- Any pole on the root locus is achievable



Procedure:

- Pick your designed closed-loop pole
  - *Has to be on the root locus plot*
- Compute the gain at that point
  - *From ECE 461: GK(s) = -1*
  - *Not important for ECE 463 Modern Control*

Example: Pick Ky = 10 to place the closed-loop dominant pole at

$$s = -0.161 + j0.946$$

Find Kr to make the DC gain equal to 1.000

- output tracks the set point

The dynamics become:

$$sX = (A - BK_yC)X + BK_rR$$

$$Y = CX$$

At DC, s = 0

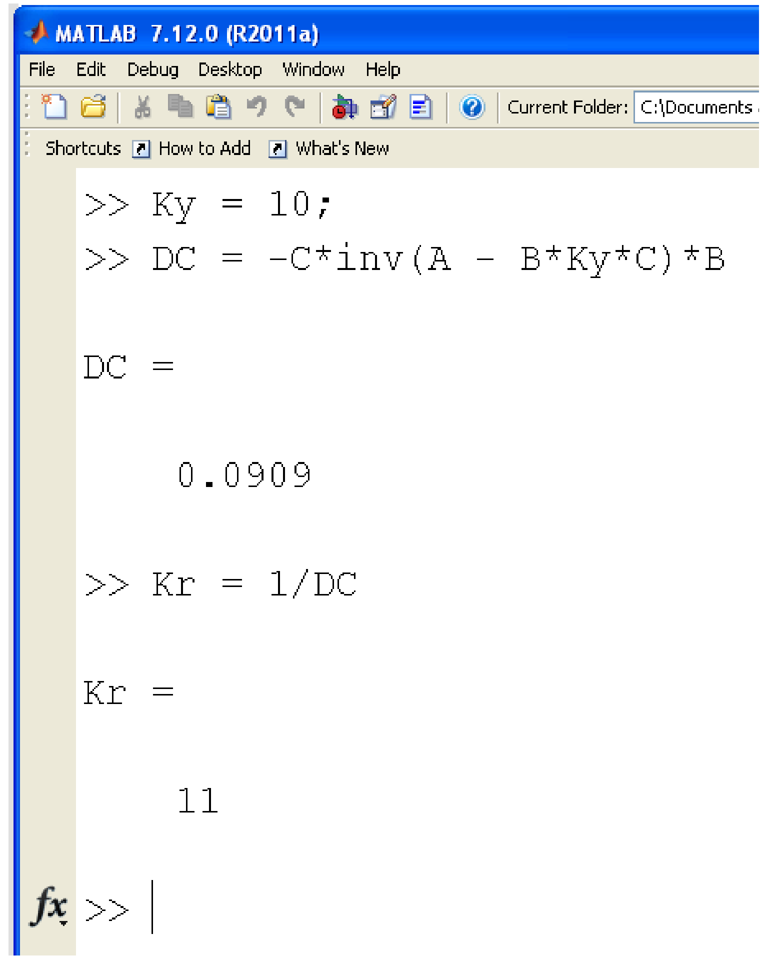
$$0 = (A - BK_yC)X + BK_rR$$

$$X = -(A - BK_yC)^{-1}BK_rR$$

$$Y = -C(A - BK_yC)^{-1}BK_rR$$

Pick Kr so that

$$-C(A - BK_yC)^{-1}BK_r = 1$$



A feedback control law would then be

$$U = K_r R - K_y Y$$

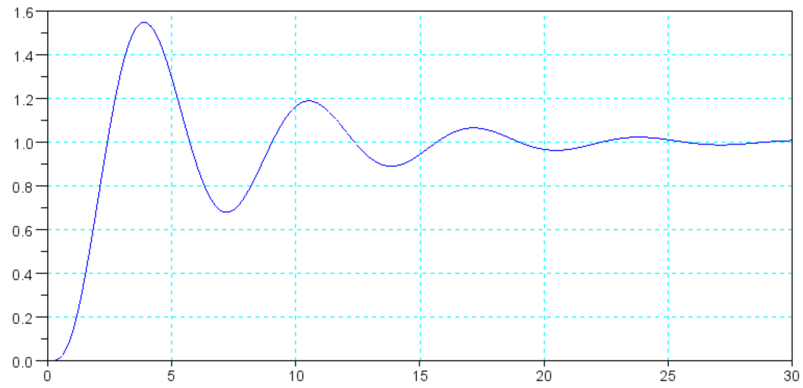
$$U = 11R - 10Y$$

The step response of the closed-loop system in Matlab is from:

```
G = ss(A-B*Ky*C, B*Kr, C, 0);
t = [0:0.01:30]';
y = step(G,t);
plot(t,y)
```

Note that

- The dominant pole is s = - 0.161 + j0.946, and
- The DC gain is one



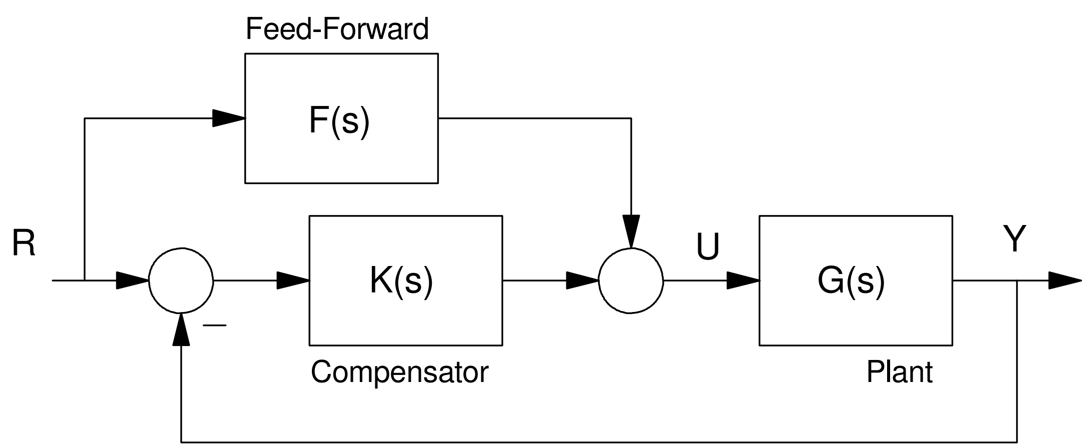
Comments on Output Feedback

With only one degree of freedom (Ky), the closed-loop poles follow a one-dimensional surface

- The root locus plot
- Defines what responses are possible by adjusting Ky

If you wand a different response add a pre-filter and a feedforward term

- Lead, Lab, PID compensators
- Covered in ECE 461 Controls Systems



Full-State Feedback:

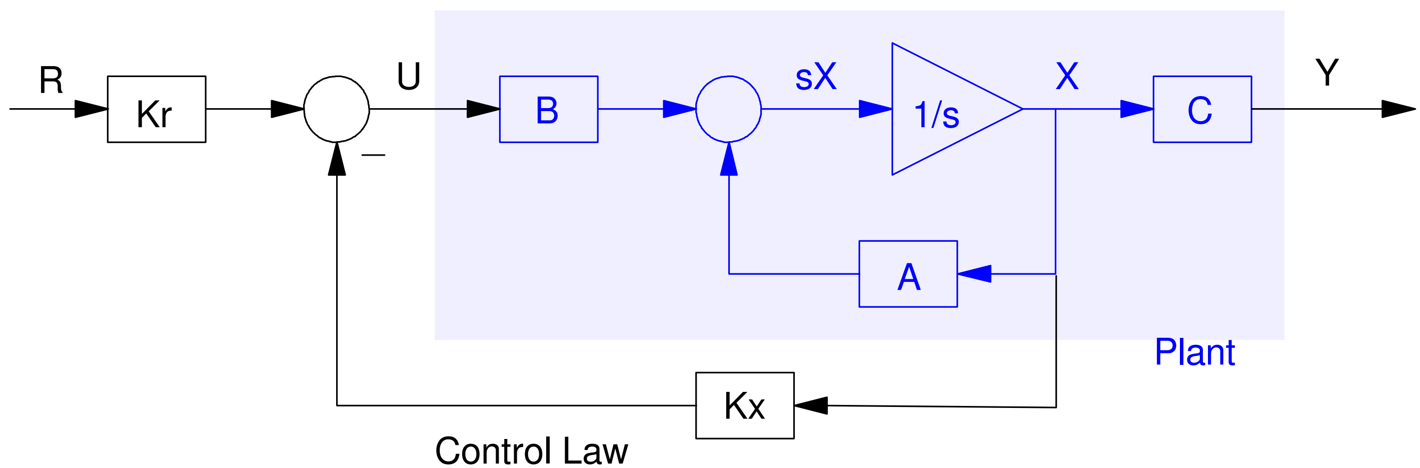
Instead of just feeding back the output (Y), feed back the states (X)

$$U = K_r R + K_x X$$

For an Nth-order system you now have N+1 degrees of freedom

- Kx has N terms
- Kr has 1 term

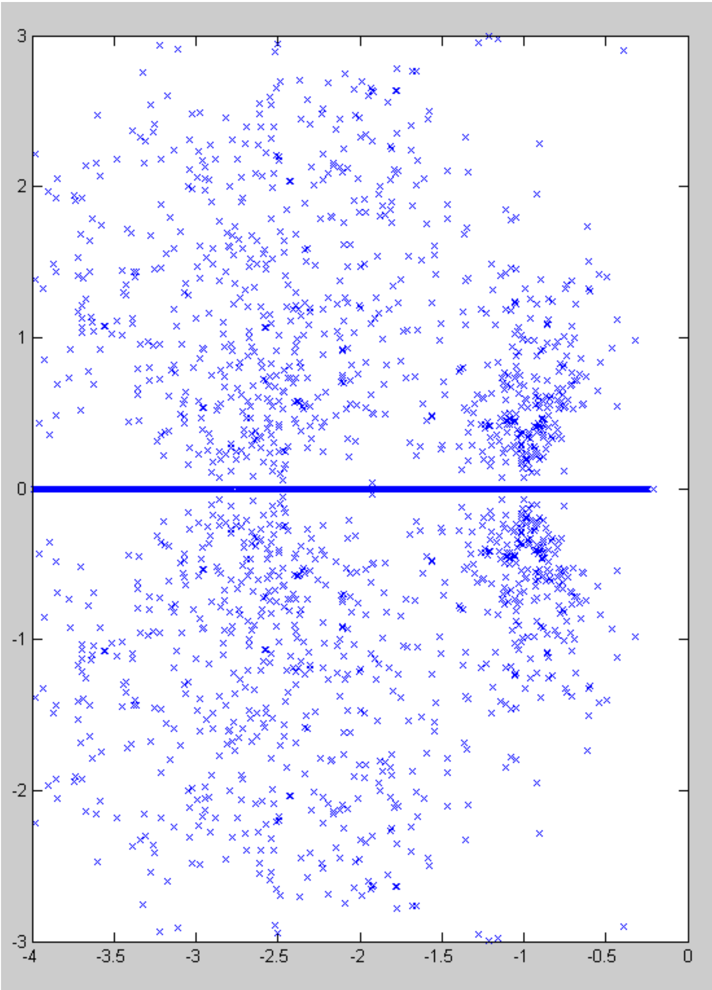
This means you can usually place the poles and DC gain anywhere



Problem: How do you find Kx and Kr?

Option 1: Trial and Error (Monte Carlo)

```
for i=1:1000
   Kx = rand(1,4)*100;
   R = eig(A - B*Kx);
   plot(real(R),imag(R),'bx');
   end
```



Doesn't really help

- Too many degrees of freedom

Finding Kx and Kr:

Option 2: Determine the closed-loop dynamics

- The eigenvalues of (A - B Kx)

$$A - BKx = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}$$

$$A - BKx = \begin{bmatrix} -2 - k_1 & 1 - k_2 & -k_3 & -k_4 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

The eigenvalues are a function of {k1, k2, k3, k4}

$$p(s) = \det(sI - A)$$

$$sI - (A - BK_x) = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & s \end{bmatrix} - \begin{bmatrix} -2-k_1 & 1-k_2 & -k_3 & -k_4 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$\det(sI - A) = \begin{vmatrix} s+2+k_1 & -1+k_2 & k_3 & k_4 \\ -1 & s+2 & -1 & 0 \\ 0 & -1 & s+2 & -1 \\ 0 & 0 & 1 & s+1 \end{vmatrix}$$

- This gives a 4th-order polynomial depending on {k1, k2, k3, k4}
- This method bogs down when you get past a 2nd-order system

There *has* to be a better way

- There is.... stay tuned...

Controllability:

With full state feedback, you have

- N equations (N eigenvalues to place) with
- N degrees of freedom (the gains in Kx)

Can all N eigenvalues be placed anywhere?

- Is there a solution for Kx given the desired closed-loop eigenvalues?

Answer

- Sometimes yes
- Sometimes no

No: Case 1

Assume B corresponds to an eigenvector.

$$B = \Lambda_1$$

Then, if you use a similarity transform

$$T = \Lambda$$

where $\Lambda$is the eigenvector matrix, then the system in diagonal form will be

$$sZ = T^{-1}ATZ + T^{-1}BU$$

$$sZ = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix} Z + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

With full-state feedback

$$U = -K_z Z + K_r R$$

$$U = -\begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} Z + K_r R$$

results in

$$sZ = \begin{bmatrix} \lambda_1 - k_1 & -k_2 & -k_3 & -k_4 \\ & \lambda_2 & & \\ & & \lambda_3 & \\ & & & \lambda_4 \end{bmatrix} Z + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} K_r R$$

Three eigenvalues are fixed, only one changes

- No - you cannot place all 4 poles anywhere if B is an eigenvector

Matlab Example:

- Let B be the first eigenvector:

```
>> A = [-2,1,0,0;1,-2,1,0;0,1,-2,1;0,0,1,-1]
>> [M,V] = eig(A)

   -0.4285   -0.6565    0.5774    0.2280
    0.6565    0.2280    0.5774    0.4285
   -0.5774    0.5774   -0.0000    0.5774
    0.2280   -0.4285   -0.5774    0.6565

>> B = M(:,1)

   -0.4285
    0.6565
   -0.5774
    0.2280

>> eig(A)

   -3.5321
   -2.3473
   -1.0000
   -0.1206
```

If you guess random feedback gains, only one pole moves:

```
>> Kx = 10*rand(1,4)
Kx =    8.1472    9.0579    1.2699    9.1338

>> eig(A-B*Kx)

   -7.3371
   -2.3473
   -1.0000
   -0.1206

>> Kx = 10*rand(1,4)
    6.3236    0.9754    2.7850    5.4688

>> eig(A-B*Kx)

   -2.3473
   -0.1206
   -1.0000
   -1.1017
```

No: Case 2:

Assume B contans all eigenvectors but one:

$$B = \Lambda_1 + \Lambda_2 + \Lambda_3$$

Converting the system to Jordan form results in

$$sZ = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix} Z + \begin{bmatrix} a \\ b \\ c \\ 0 \end{bmatrix} U$$

Even with full-state feedback, the pole at λ_4 will not move

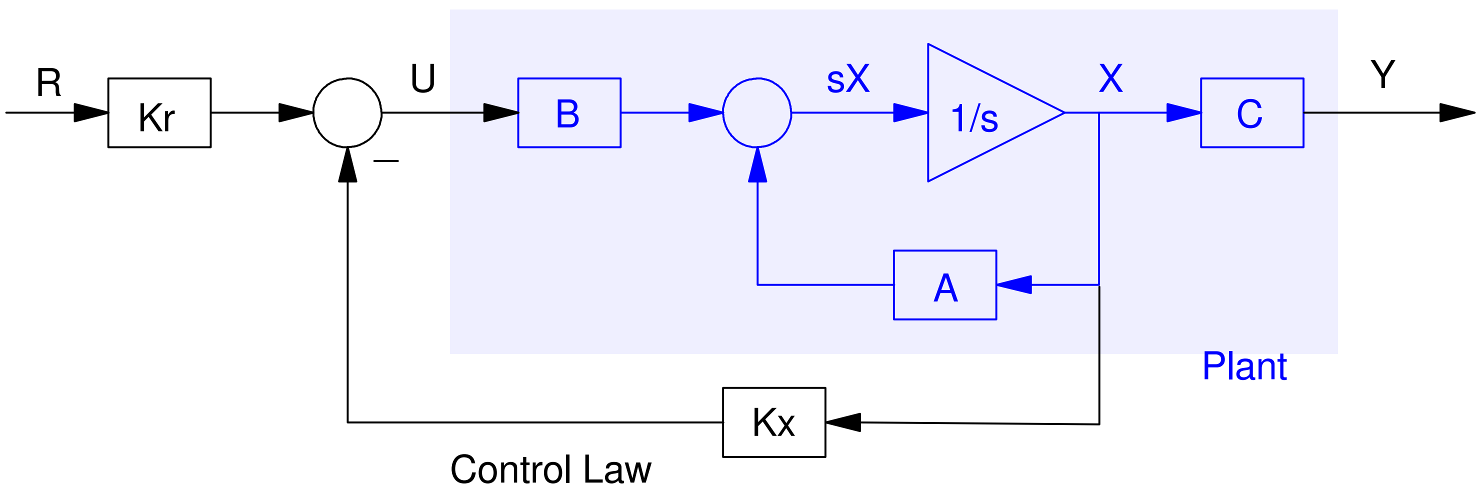
When can you place N poles anywhere?

The B matrix must contain all N eigenvectors

- This is another way of saying the sytem must be controllable
- (PBH rank test)

Problem: How to find Kx and Kr?

- Tomorrow's lecture...



Summary

Feedback is all important

- It allows you to use systems which are open-loop unstable
- It allows you to improve the response of a system
- It allows you to force a system to track a set point

Following Lectures

- How to determine the feedback gains to meet your design requirements