Full State 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 can eventually learn how to walk or ride a bike. What's happening as you practice and learn is you're figuring out how to adjust the input to the system based upon how it's behaving.

Mathematically, this can be seen as follows:

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 how the system with feedback (termed the closed-loop system) behaves.

Example: Consider the temperature along a metal bar (or a 4-stage RC filter) with dynamics

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

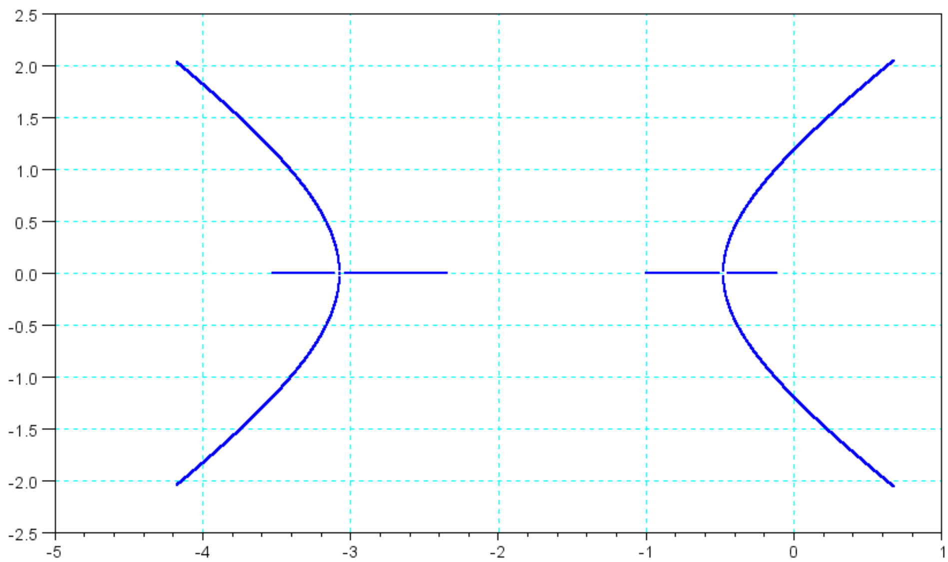
$$Y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} X$$

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.');
>> eig(A)

   -3.5321
   -2.3473
   -1.0000
   -0.1206

>> grid(c)
```



Note that when the gain, Ky, is zero, the poles are at the eigenvalues of A: { -3.5, -2.34, -1.00, -0.12 }. As the feedback gain increases, the poles shift - meaning the dynamics of the system with feedback (the closed-loop system) is changing. The above graph shows how the dynamics change. You can also see this by computing the closed-loop poles for different values of Ky

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

In this example, there is a single feedback gain. The roots likewise follow a one-dimensional curve as shown above. This is termed the systems root locus and is the topic of ECE 461/661 Controls Systems.

With a single feedback gain, the first step is to select the "best" gain - meaning the best spot on the root locus. If you want the dominant pole to be at

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

then the feedback gain, Ky, should be 10. To find Kr, you often set the DC gain to one (meaning the output tracks the set point). The DC gain is found from

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

$$Y = CX$$

At DC, s = 0

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

meaning

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

and

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

If the DC gain is to be one

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

or

$$K_r = \left(-C(A - BK_yC)^{-1}B\right)^{-1}$$

Example:

```
-->Ky = 10
 Ky  =

    10.

-->-C*inv(A - B*Ky*C)*B
 ans  =

    0.0909091

-->Kr = inv(ans)
 Kr  =

    11.
```

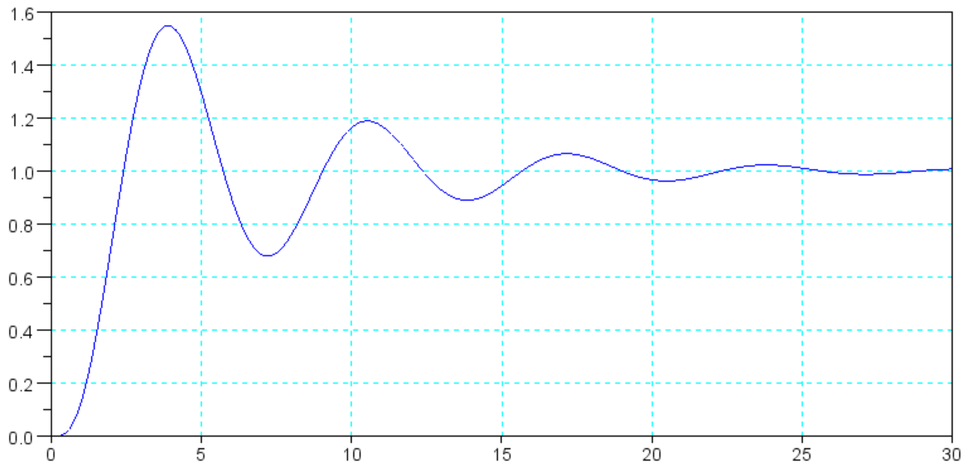
A feedback control law would then be

$$U = K_rR - K_yY$$

$$U = 11R - 10Y$$

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)
-->xgrid(4)
```



Note that

- The system behaves like a system with a dominant pole at s = - 0.161 + j0.946, and
- The DC gain is one

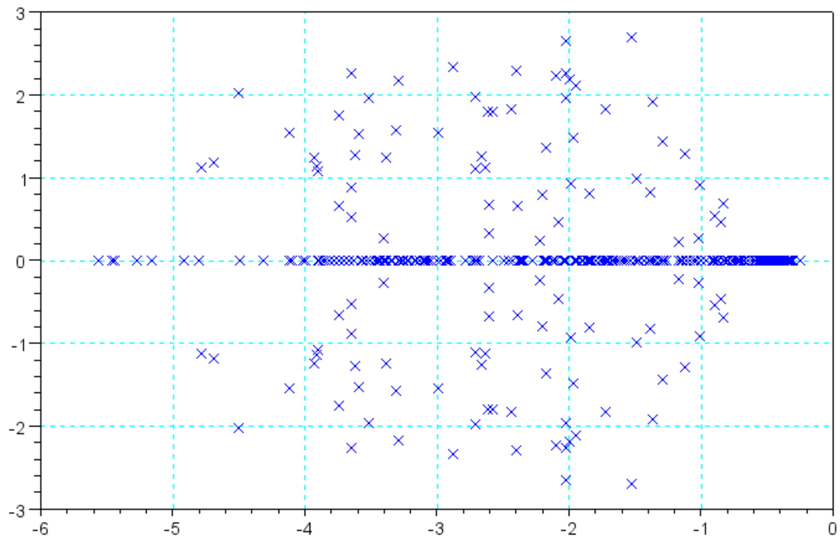
Full-State Feedback:

Suppose in stead that you could feed back all four states:

$$U = K_r R + K_x X$$

```
-->for i=1:100
-->   Kx = rand(1,4)*10;
-->   R = spec(A - B*Kx);
-->   plot(real(R),imag(R),'bx');
-->   end

-->xgrid(4)
```



Now, the poles are no longer following a nice curve like before. In fact, for this system you can place the poles anywhere you want. This shouldn't be surprising since the system has four degrees of freedom and four eigenvalues.

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

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

Pole Placement:

One way to find the poles of a system is to compute the characteristic polynomial, p(s)

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

Here,

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

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

= some 4th-order polynomial

If you match the coefficients of the polynomial with the desired polynomial, you have the gains, Kx.

There has to be a better way though. There is - stay tuned.

Controllability:

You would think that with four degrees of freedom, you could place four eigenvalues (poles) at will. Sometimes this is the case. For example, in the previous case, all four poles were shifting as you changed the gain. Sometimes this is not the case.

For example, consider the case where B corresponds to an eigenvector.

$$B=\Lambda_1$$

Then, if you use a similarity transform

$$T=\Lambda$$

where Λ 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

$$U=-K_zZ+K_rR$$

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

results in

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

Three eigenvalues are fixed, only one changes

In Matlab:

```
>> 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

>> 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
```

Note that three eigenvalues do not move. If this is the case, there is no point in trying to place all four poles.

Likewise, what we need is

- A way to detect if a system is controllable (if the poles can be placed at will), and
- If so, a way to compute the feedback gains, Kx, which place the poles of (A - BKx)