LQG Observers

NDSU ECE 463/663

Lecture #29

Inst: Jake Glower

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

Full-Order Observers

Given a dynamic system

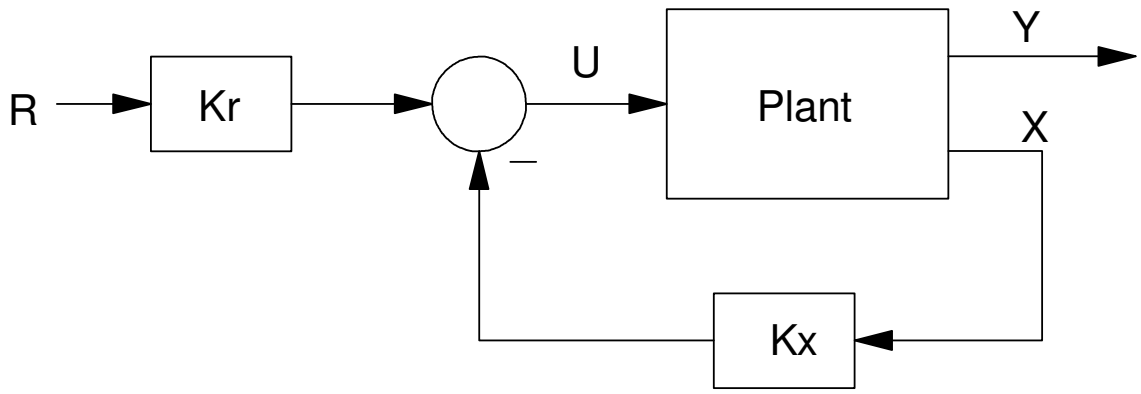
$$sX = AX + BU$$

$$Y = CX$$

you can stabilize the system using full-state feedback

$$U = -K_x X + K_r R$$

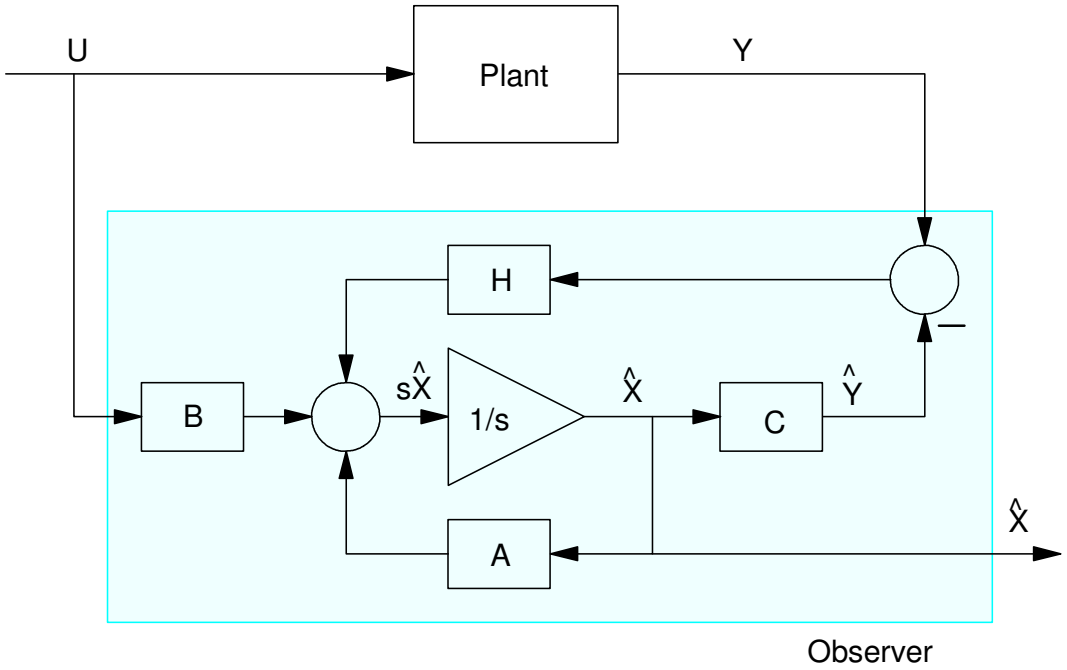
Kx, can be computed using Bass-Gura (pole placement) or LQR methods.



If the states are not measured, they can be estimated

$$sX_e = AX_e + BU + H(Y - Y_e)$$

$$Y_e = CX_e$$



Bass Gura can be used to find H

LQR can also be used to find H

- A is replaced with A^T
- B is replaced with C^T, and
- The gain you compute is H^T

$$s\begin{bmatrix} X \\ X_e \end{bmatrix} = \begin{bmatrix} A & 0 \\ HC & A - HC \end{bmatrix}\begin{bmatrix} X \\ X_e \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix}U$$

Example: Metal Bar (4th order RC filter).

First, pick Q and R. All states matter, so let Q = I

- Q = I
- R = 1

```
Q = diag([1,1,1,1]);
R = 1;
H = lqr(A', C', Q, R)'

    0.3713
    0.3116
    0.2826
    0.2705

eig(A - H*C)

   -3.5606
   -2.4418
   -1.1441
   -0.2248     Dominant pole for the observer
```

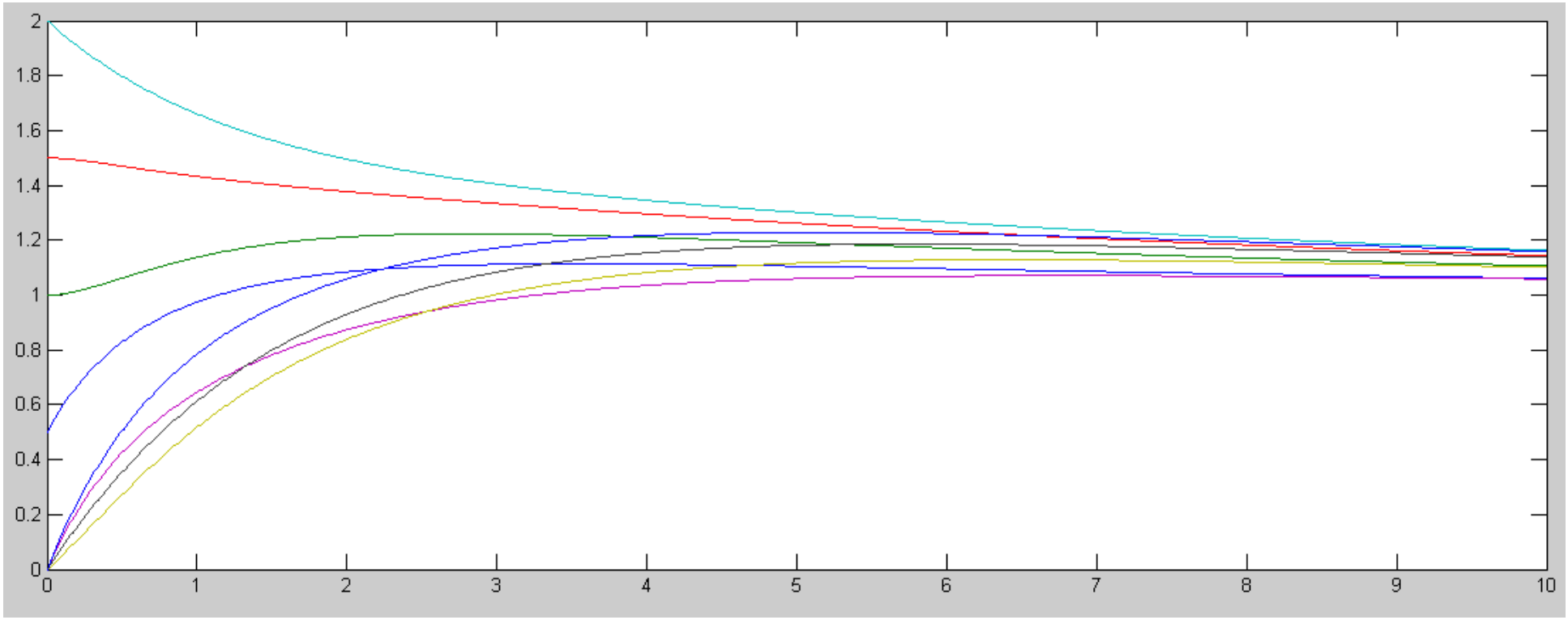
The augmented system (plant plus observer) is then

```
A8 = [A, zeros(4,4);H*C, A-H*C]
B8 = [B; B];
C8 = eye(8,8);
D8 = zeros(8,1);
```

Add initial conditions:

```
X0 = [0.5;1;1.5;2; 0;0;0;0]

    0.5000
    1.0000  initial conditions of the plant
    1.5000
    2.0000
  - - - - - - -
          0
          0  initial conditions of the observer
          0
          0
```

```
y8 = step3(A8, B8, C8, D8, t, X0, 0*t+1);
plot(t,y8)
```

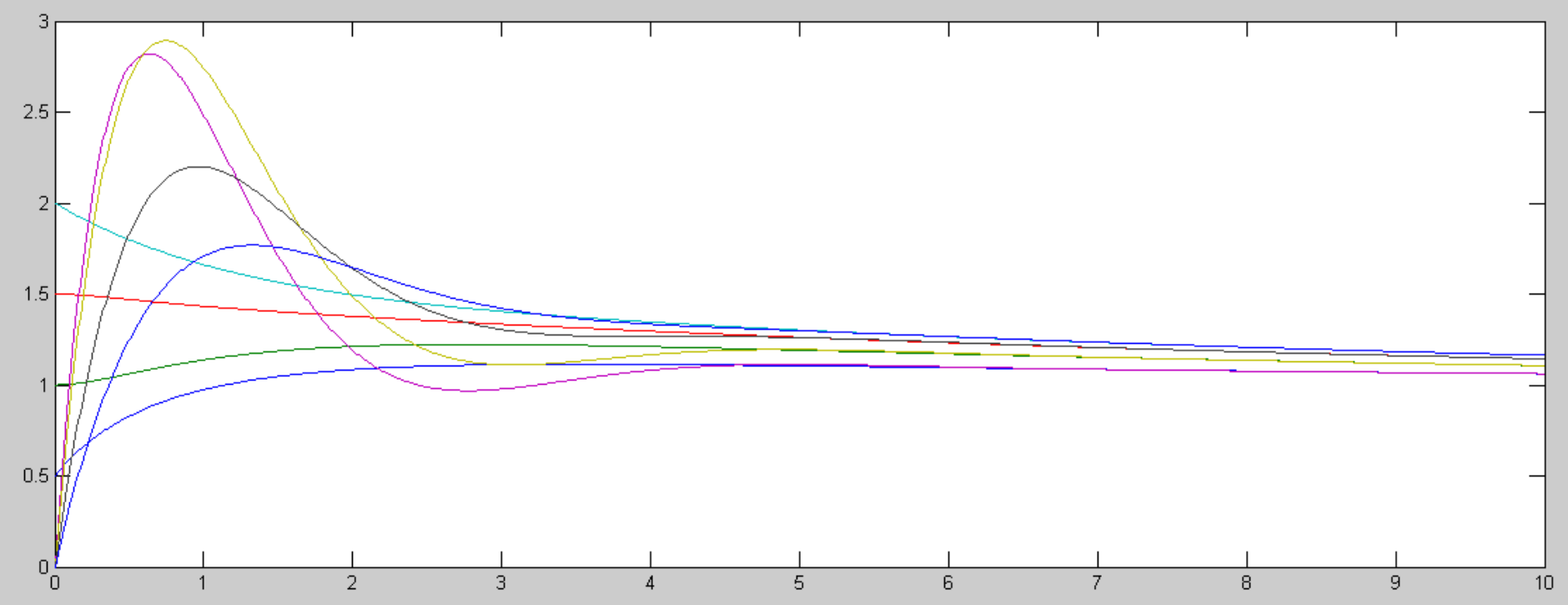


Plant and Observer States for Q = I, R = 1. Note that the states converge

To speed up the observer, increase the weightings.

Weighting X2 works best, so go with that. (Q is somewhat arbitrary)

```
Q = diag([1,1000,1,1]);
R = 1;
H = lqr(A', C', Q, R)'

    5.4325
    4.9030
    2.8963
    1.7915

A8 = [A, zeros(4,4);H*C, A-H*C];
B8 = [B; B];
C8 = eye(8,8);
D8 = zeros(8,1);
y8 = step3(A8, B8, C8, D8, X0, t);
plot(t,y8);
```



Step Response with Iniital Conditions: Q = diag(1 1000 1 1)   R = 1

Example 2: Gantry System: Output = Position

```
A = [0,0,1,0;0,0,0,1;0,-19.6,0,0;0,29.4,0,0]

         0         0    1.0000         0
         0         0         0    1.0000
         0  -19.6000         0         0
         0   29.4000         0         0

B = [0;0;1;-1]

     0
     0
     1
    -1

C = [1,0,0,0]

C =     1     0     0     0
```

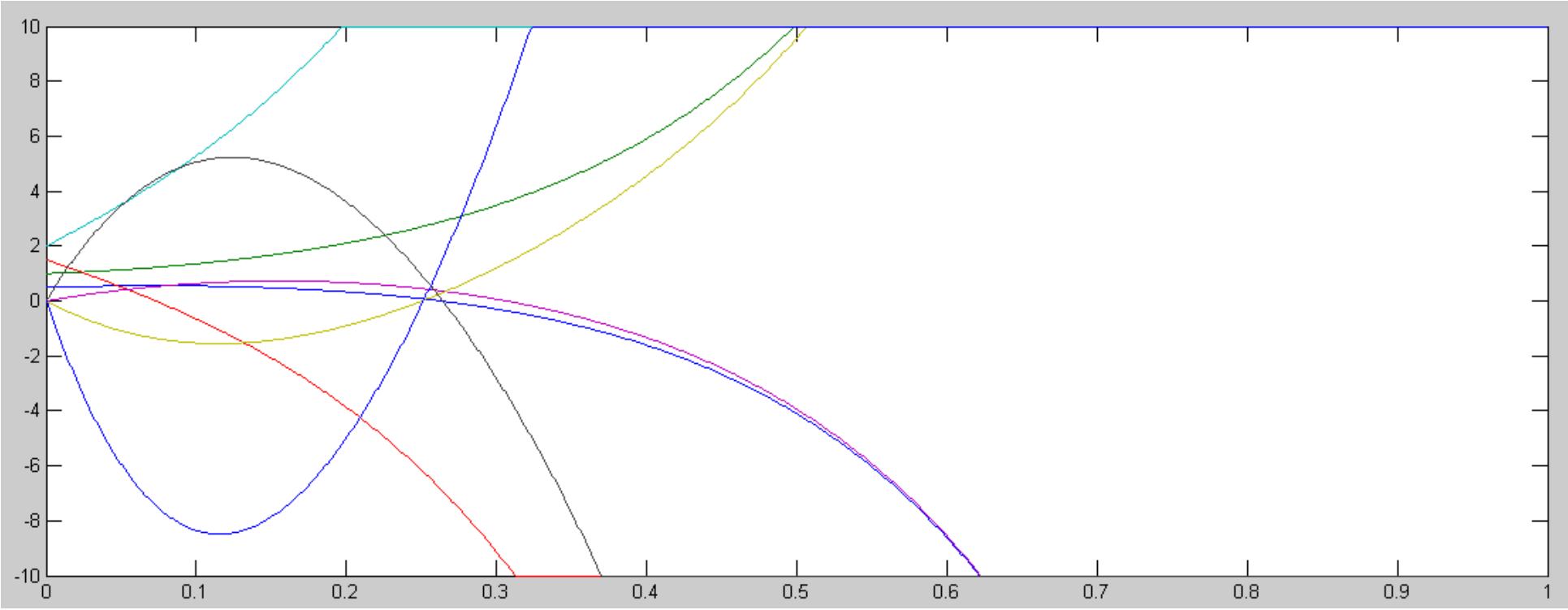
Q and R are tools you can adjust to get the response you want. Trying different weightings to see which ones speed up the system the most:

Weighting velocity seems to work best, so let Q = diag([1, 1, 1000, 1])

```
Q = diag([1,1,1e3,1]);
R = 1;
H = lqr(A', C', Q, R)'

   18.8640
  -57.8416
  177.4249
 -313.5820

eig(A - H*C)

  -4.0134 + 3.9521i
  -4.0134 - 3.9521i
  -5.6431
  -5.1940
```

The plant is unstable, so the open-loop response will take off to infinity. Regardless, the observer states do converge - even if the plant isn't.

```
A8 = [A, zeros(4,4);H*C, A-H*C];
B8 = [B; B];
C8 = eye(8,8);
D8 = zeros(8,1);
X0 = [0.5;1;1.5;2; 0;0;0;0];
t = [0:0.001:1]';
y8 = step2(A8, B8, C8, D8, X0, t);
plot(t,min(10,max(-10, y8)))
```



Plant and Observer States for an Open-Loop Step Response:

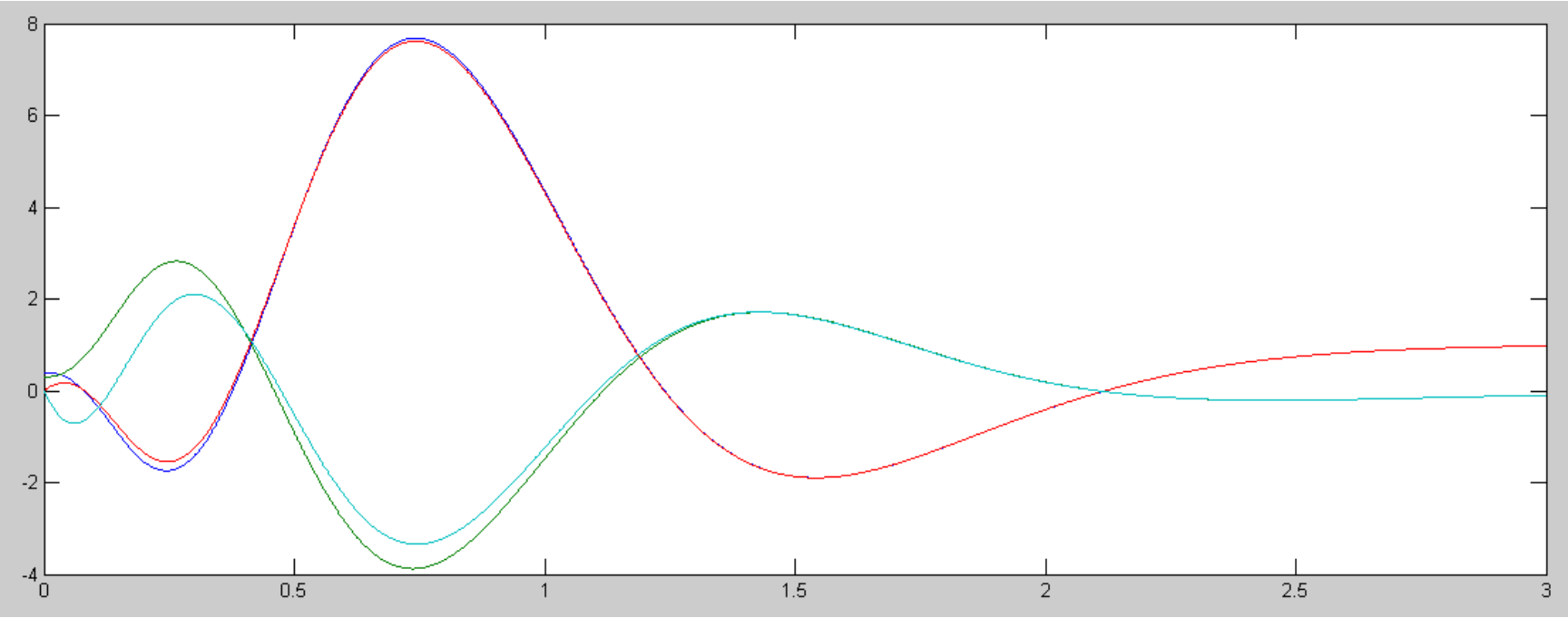
Add full-state feedback to stabilize the system:

```
Q = diag([1000,0,0,0]);
R = 1;
Kx = lqr(A, B, Q, R);
eig(A-B*Kx)

  -5.3273 + 2.4049i
  -5.3273 - 2.4049i
  -2.8173 + 1.0648i
  -2.8173 - 1.0648i

Kx =   -31.6228 -164.2922  -29.5050  -45.7942

A8 = [A, -B*Kx  ;  H*C, A-B*Kx-H*C];
DC = -C*inv(A - B*Kx)*B
   -0.0316

Kr = 1/DC
  -31.6228
```

The combined system is then

$$\begin{bmatrix} sX \\ sX_e \end{bmatrix} = \begin{bmatrix} A & -BK_x \\ HC\,A - BK_x - HC \end{bmatrix} \begin{bmatrix} X \\ X_e \end{bmatrix} + \begin{bmatrix} BK_r \\ BK_r \end{bmatrix} R$$

```
A8 = [A, -B*Kx;H*C, A-H*C-B*Kx];
B8 = [B*Kr; B*Kr];
C8 =
[1,0,0,0,0,0,0,0;0,1,0,0,0,0,0,0;0,0,0,0,1,0,0,0;0,0,0,0,0,1,0,0];

     1     0     0     0     0     0     0     0     position
     0     1     0     0     0     0     0     0     angle
     0     0     0     0     1     0     0     0     position est
     0     0     0     0     0     1     0     0     angle estimate

D8 = zeros(4,1);
X0 = [0.4;0.3;0.2;0.1; 0;0;0;0];
y8 = step2(A8, B8, C8, D8, X0, t);
plot(t,min(10,max(-10, y8)))>>
```

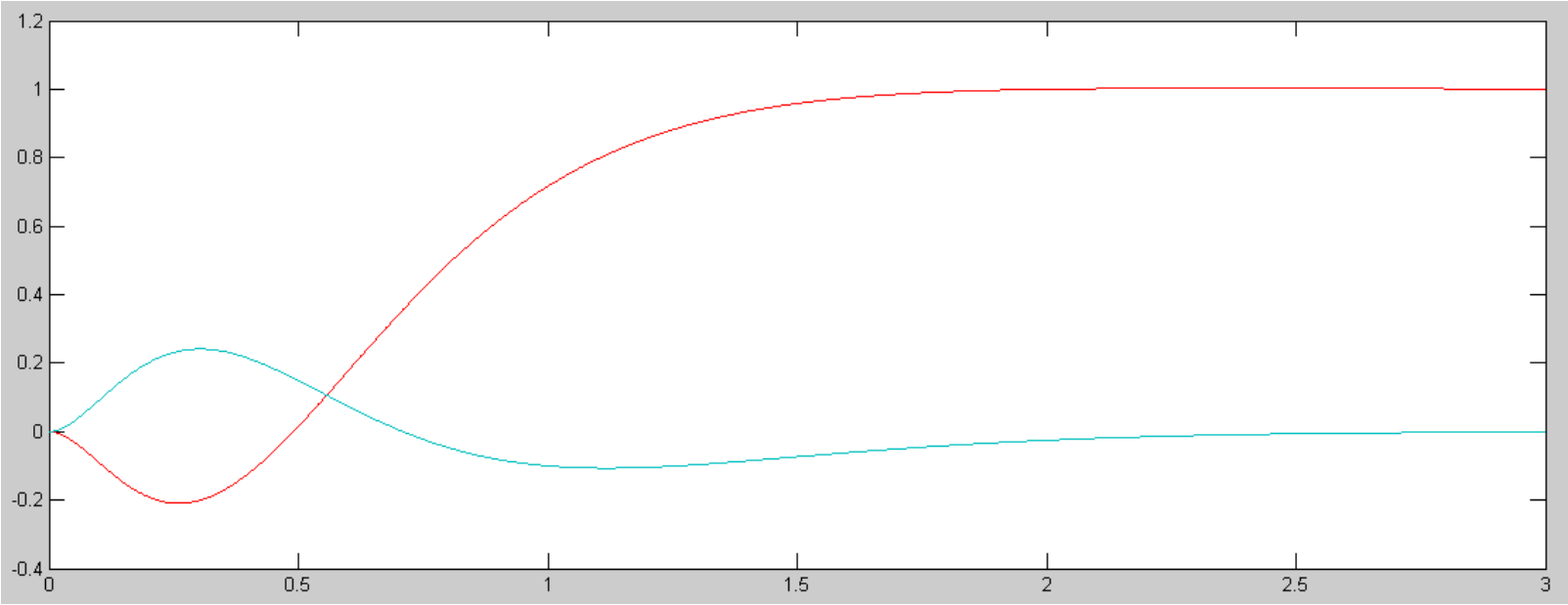


Step Response for the Plant, Observer, and Full-State Feedback with Errors in the Inital Estimates of the States

Note that the state estiamtes converge to the actual states.

Once the state estimates converge, (X0 is zero meaning no error in the state estimate) the system behaves much better:

```
y8 = step2(A8, B8, C8, D8, 0*X0, t);
plot(t,y8)
```



Step Response for the Plant, Observer, and Full-State Feedback with No Error in the State Estimates

Observers with Multiple Outputs:

With LQR, you can use multiple outputs. For example, assume you can measure both poistion and angle:

```
C = [1,0,0,0 ; 0 1 0 0]

     1     0     0     0    position (x)
     0     1     0     0    angle (q)

Q = diag([1,1,1e3,1e3]);
R = diag([1,1]);
H = lqr(A', C', Q, R)'

        x          q
    8.4497   -1.6848
   -1.6848   11.9071
   36.6179  -26.1793
   -8.1168   71.8085
```

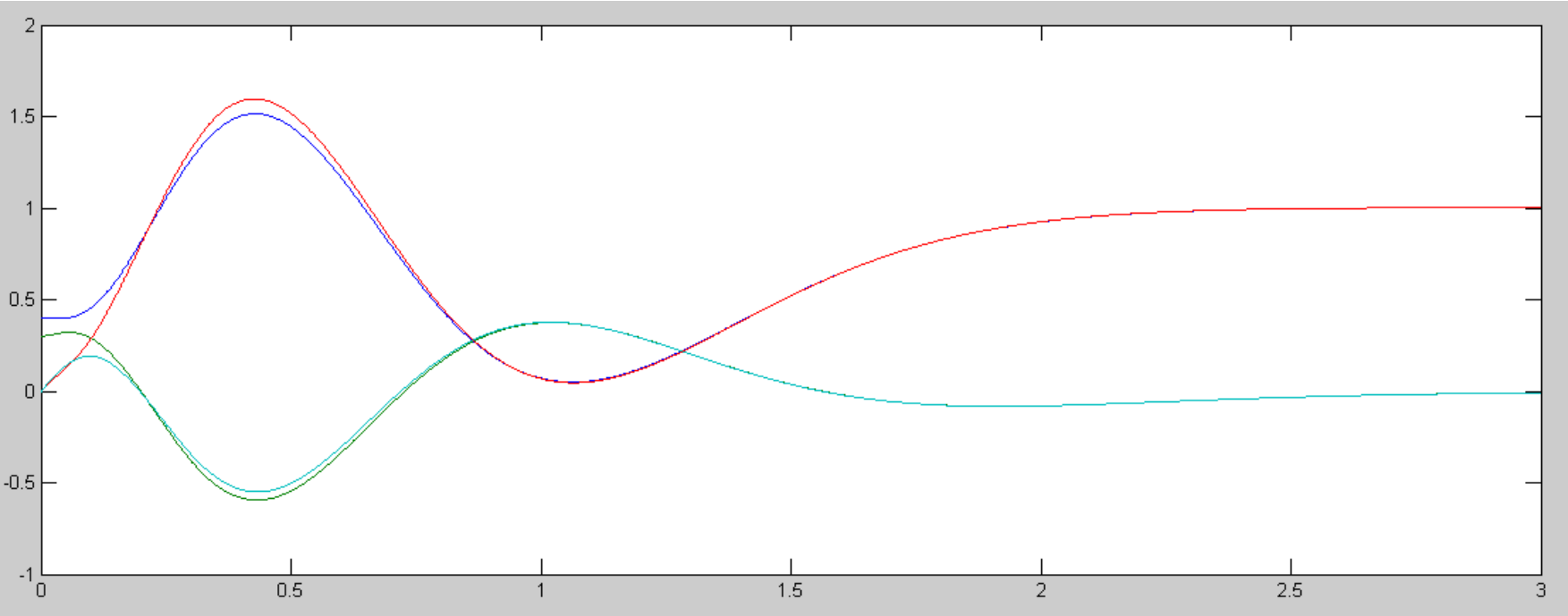
```
eig(A - H*C)

   -3.9395 + 4.1662i
   -3.9395 - 4.1662i
   -6.2388 + 2.5857i
   -6.2388 - 2.5857i
```

Note that with LQR methods, you can handle two outputs without any problems: the observer gain, H, has two columns - one for position (x) and one for angle (q)

The step response of the plant plus observer plus full-state feedback is then:

```
A8 = [A, -B*Kx;H*C, A-H*C-B*Kx];
B8 = [B*Kr; B*Kr];
C8 = [1,0,0,0,0,0,0,0;0,1,0,0,0,0,0,0;0,0,0,0,1,0,0,0;0,0,0,0,0,1,0,0]

     1     0     0     0     0     0     0     0   position, x
     0     1     0     0     0     0     0     0   angle, q
     0     0     0     0     1     0     0     0   position estiamte xm
     0     0     0     0     0     1     0     0   angle estimate qm

D8 = zeros(4,1);
```

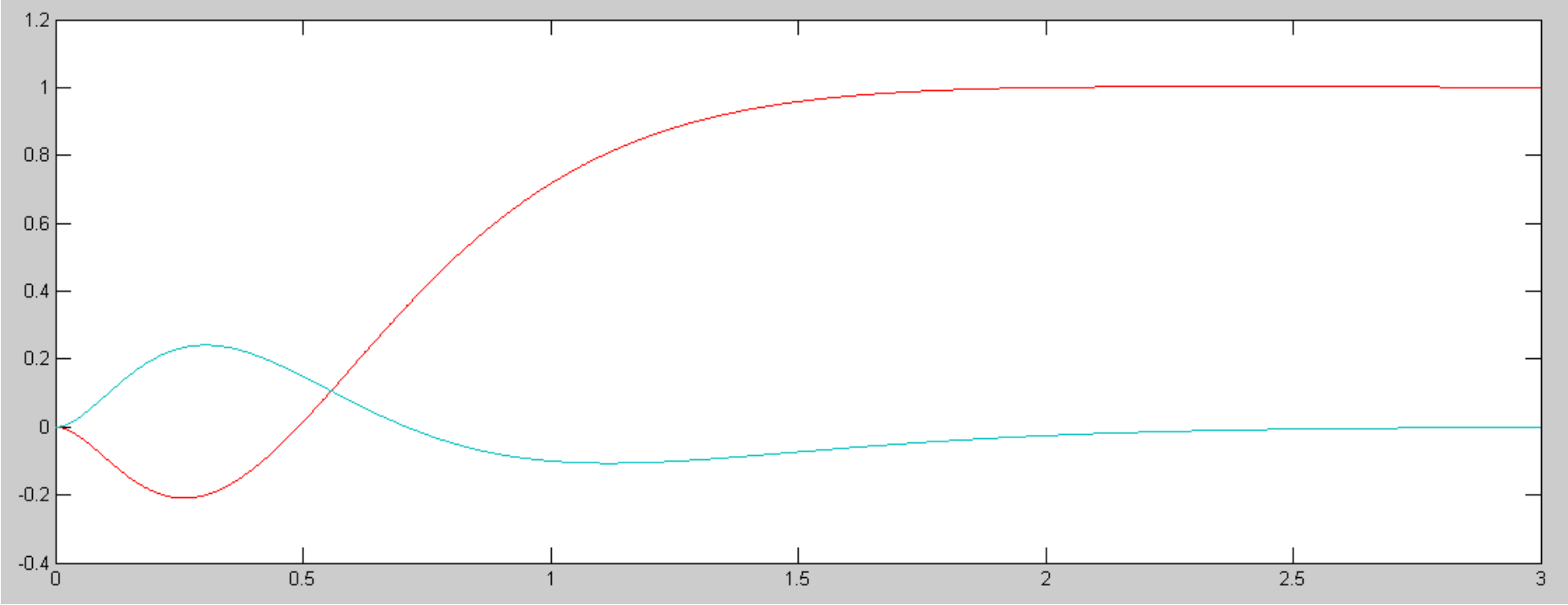
```
y8 = step2(A8, B8, C8, D8, X0, t);
plot(t,y8)
```



Position (blue and red) and angle (green teal) for the plant and observer with initial error in the state estimates:

If initial conditions match, the tracking is better:

```
>> y8 = step2(A8, B8, C8, D8, 0*X0, t);
>> plot(t,min(10,max(-10, y8)))
```



Step response with no error in the initial state estiamtes: Position (red) and angle (green)

Note that H is

x	q	
8.4497	-1.6848	x update
-1.6848	**11.9071**	q update
36.6179	-26.1793	sx update
-8.1168	**71.8085**	sq update

As you would expect

- The position sensor mostly updates position and velocity estimates
- The angle sensor mostly updates angle and angular velocity estimates