LQG/LTR with Servo Compensators

NDSU ECE 463/663

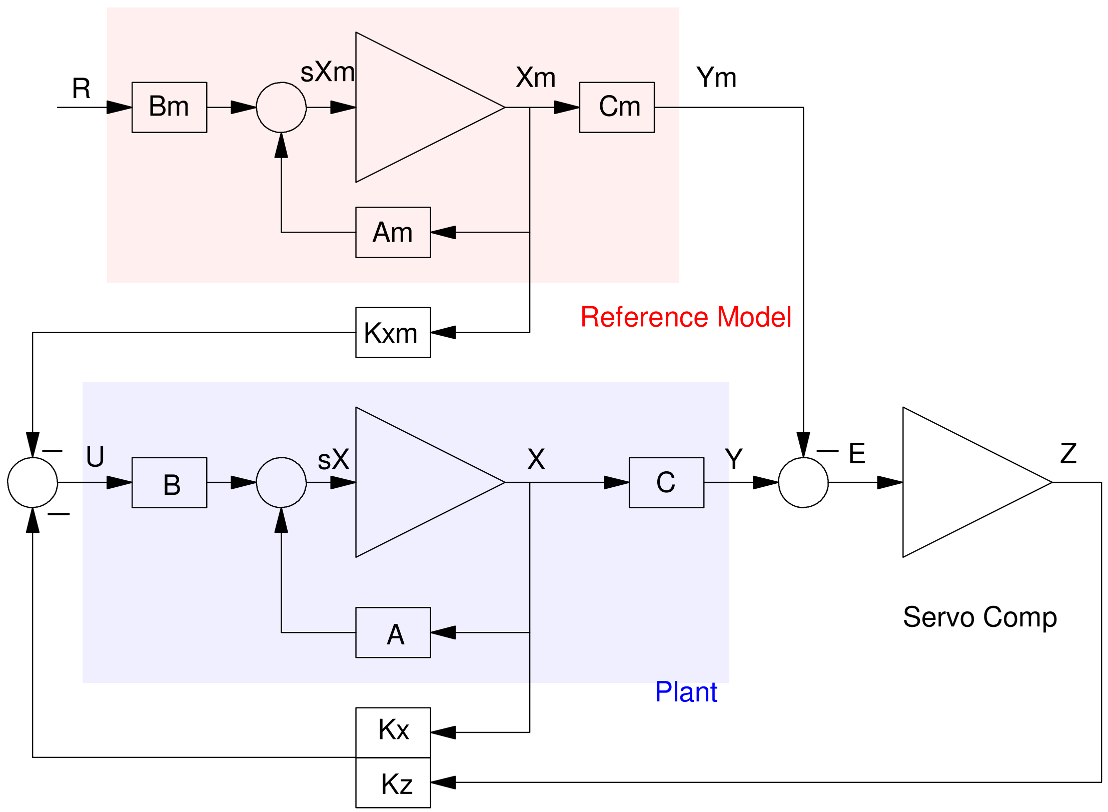
Lecture #32

Inst: Jake Glower

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

LQG/LTR with a Servo Compensator

- Previous designs did not results in a DC gain of 1.00
- Add a servo compensator to force the DC gain to 1.00



Formulation

Given a system

$$sX = AX + BU$$

$$Y = CX$$

along with a reference model which defines how the system *should* behave

$$sX_m = AX_m + B_m R$$

$$Y_m = C_m X_m$$

Add a servo-compensator to force the plant to track the model at DC:

$$sZ = Y - Y_m$$

The augmented system is then

$$s\begin{bmatrix} X \\ Z \\ X_m \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ C & 0 & -C_m \\ 0 & 0 & A_m \end{bmatrix}\begin{bmatrix} X \\ Z \\ X_m \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}U + \begin{bmatrix} 0 \\ 0 \\ B_m \end{bmatrix}R$$

$$U = -\begin{bmatrix} K_x & K_z & K_{xm} \end{bmatrix}\begin{bmatrix} X \\ Z \\ X_m \end{bmatrix}$$

The system output (defining Q) is

$$Y = Z = \begin{bmatrix} 0 & I & 0 \end{bmatrix}\begin{bmatrix} X \\ Z \\ X_m \end{bmatrix}$$

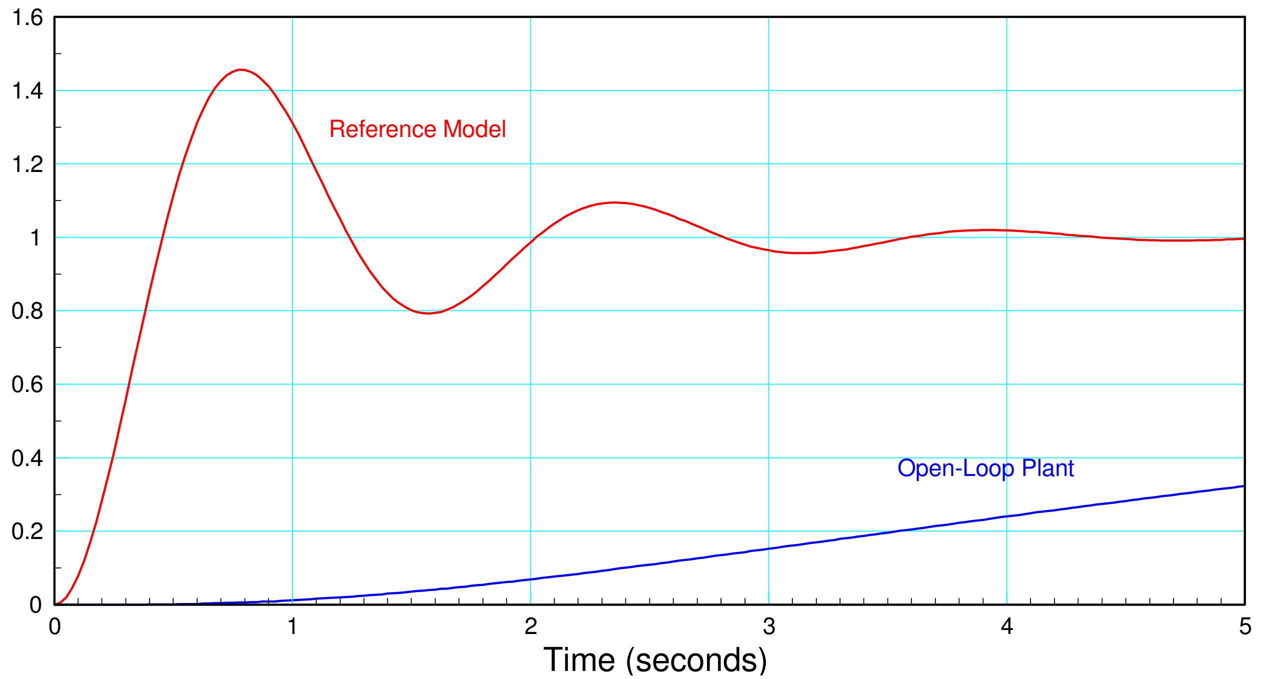
Forcing Z to zero forces the plant to track the reference model.

Example: 4th-Order Heat Equation

Assume you want the 4th-order heat equation to behave as

$$Y_m = \left(\frac{17}{s^2+2s+17}\right) R$$

Here, an underdamped system is used just to challenge the design method.



First generate the augmented system

Plant:

```
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]
```

Reference Model:

```
Am = [0,1;-17,-2];
Bm = [0;17];
Cm = [1,0];
```

7th-Order Augmented System:

```
A7 = [A, zeros(4,3);C,0,-Cm;zeros(2,5),Am]
```

$$
\begin{array}{cccc:c:cc}
\multicolumn{4}{c}{\text{plant}} & \text{servo} & \multicolumn{2}{c}{\text{ref model}} \\
-2 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & -2 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 1 & 0 & -1 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & -17 & -2
\end{array}
$$

```
B7 = [B;0;0;0]
Br = [zeros(5,1);Bm]
C7 = [  C, 0, 0*Cm ; 0*C, 0, Cm]
```

Q weights the servo compensator state for a quick system:

```
Cz = [ 0 0 0 0 1 0 0 ];

Q7 = Cz' * Cz

     x1    x2    x3    x4    z    ref model
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0
     0     0     0     0     1     0     0
     0     0     0     0     0     0     0
     0     0     0     0     0     0     0

R7 = 1;
```

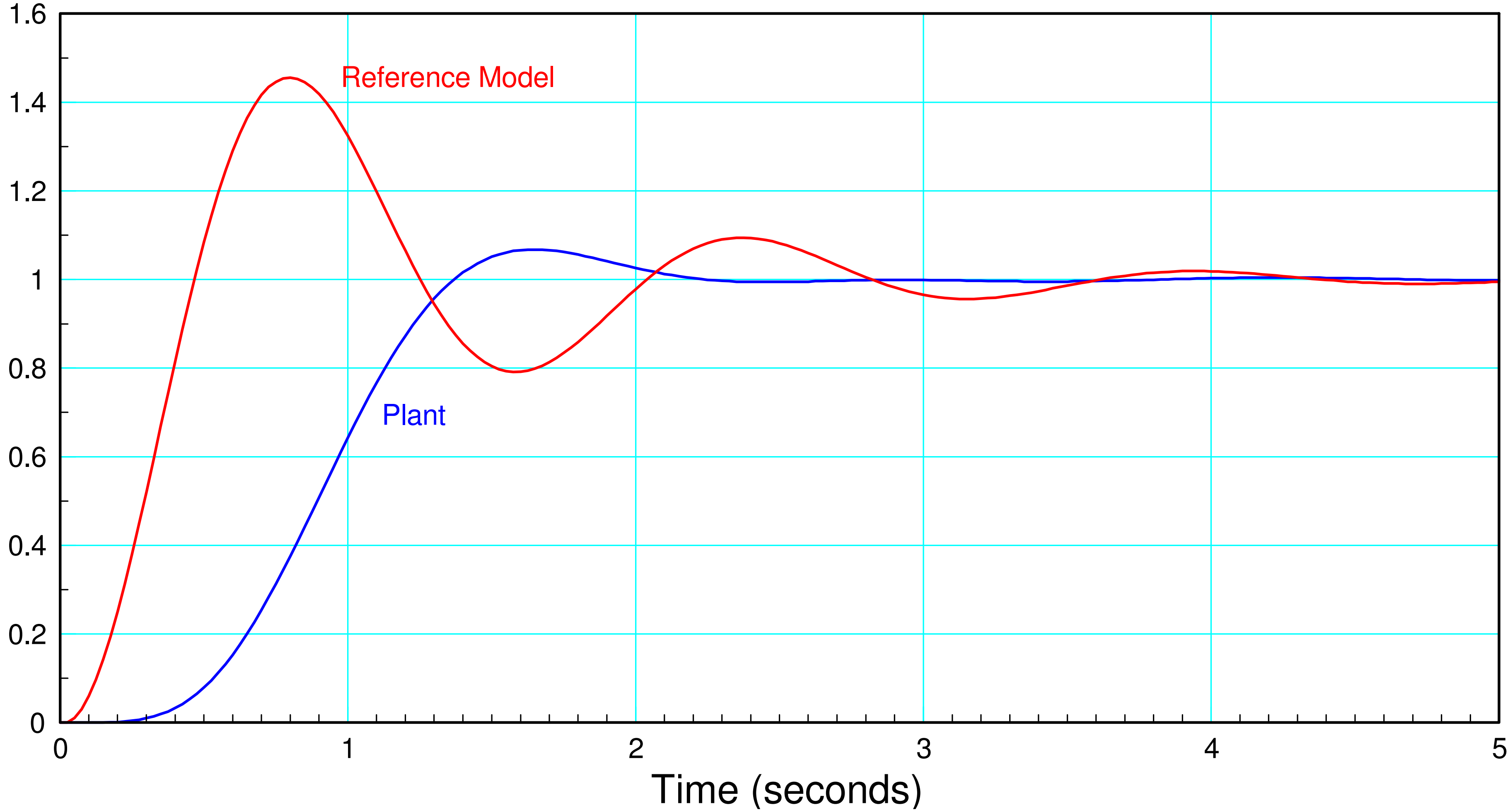
Determine the feedback gains with a weighting of 10^6 Q

```
K7 = lqr(A7, B7, Q7*1e6, R7)

7.3587    41.7931   181.6160  618.1126  1000.  -125.240 -71.358
```

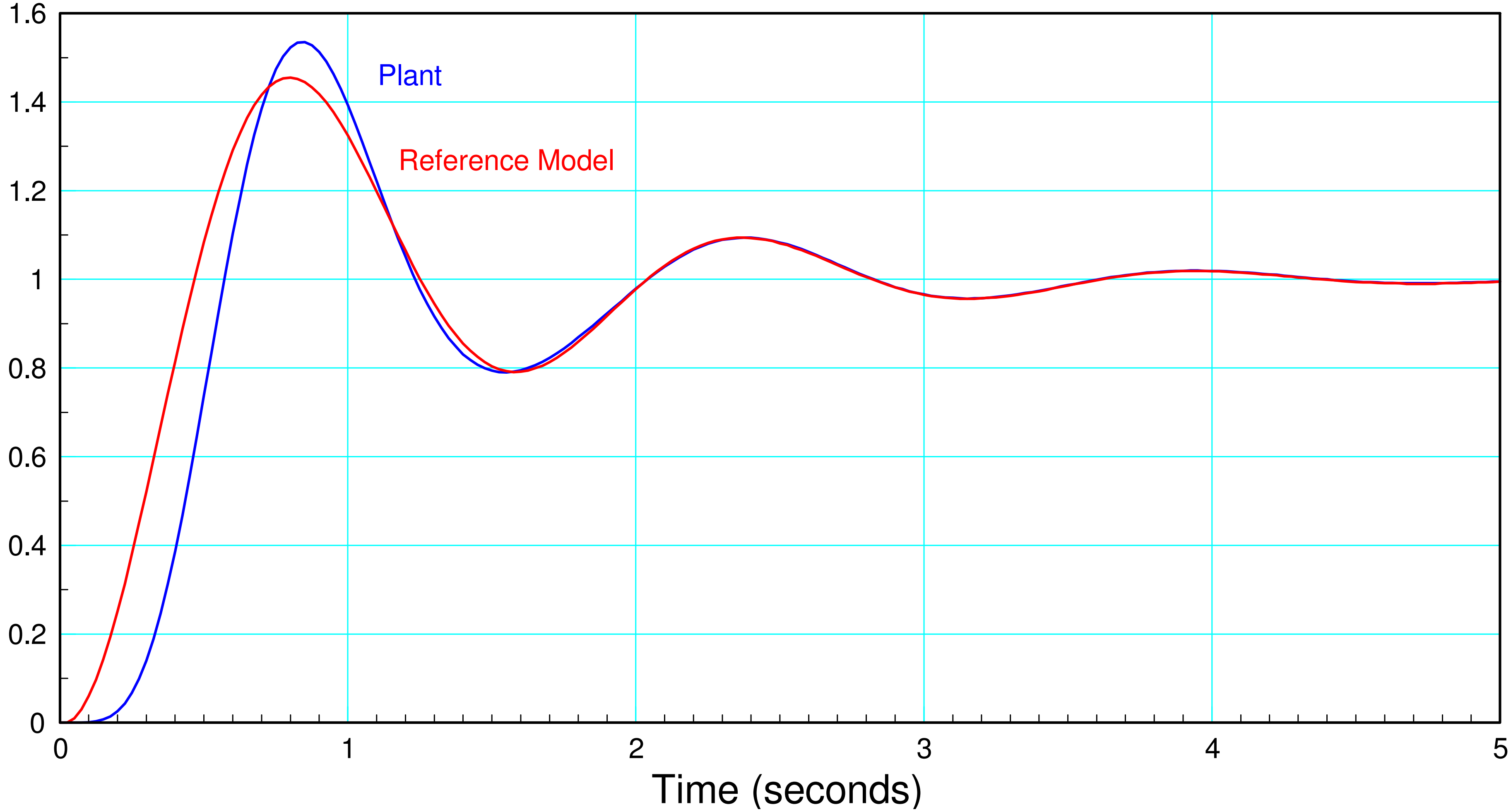
The closed-loop poles are:

```
eig(A7-B7*K7)

  - 1.3359721 + 3.3681608i   Dominant Poles
  - 1.3359721 - 3.3681608i
  - 4.5325933
  - 3.5771084 + 2.002054i
  - 3.5771084 - 2.002054i
  - 1. + 4.i                 Reference Model States
  - 1. - 4.i
```

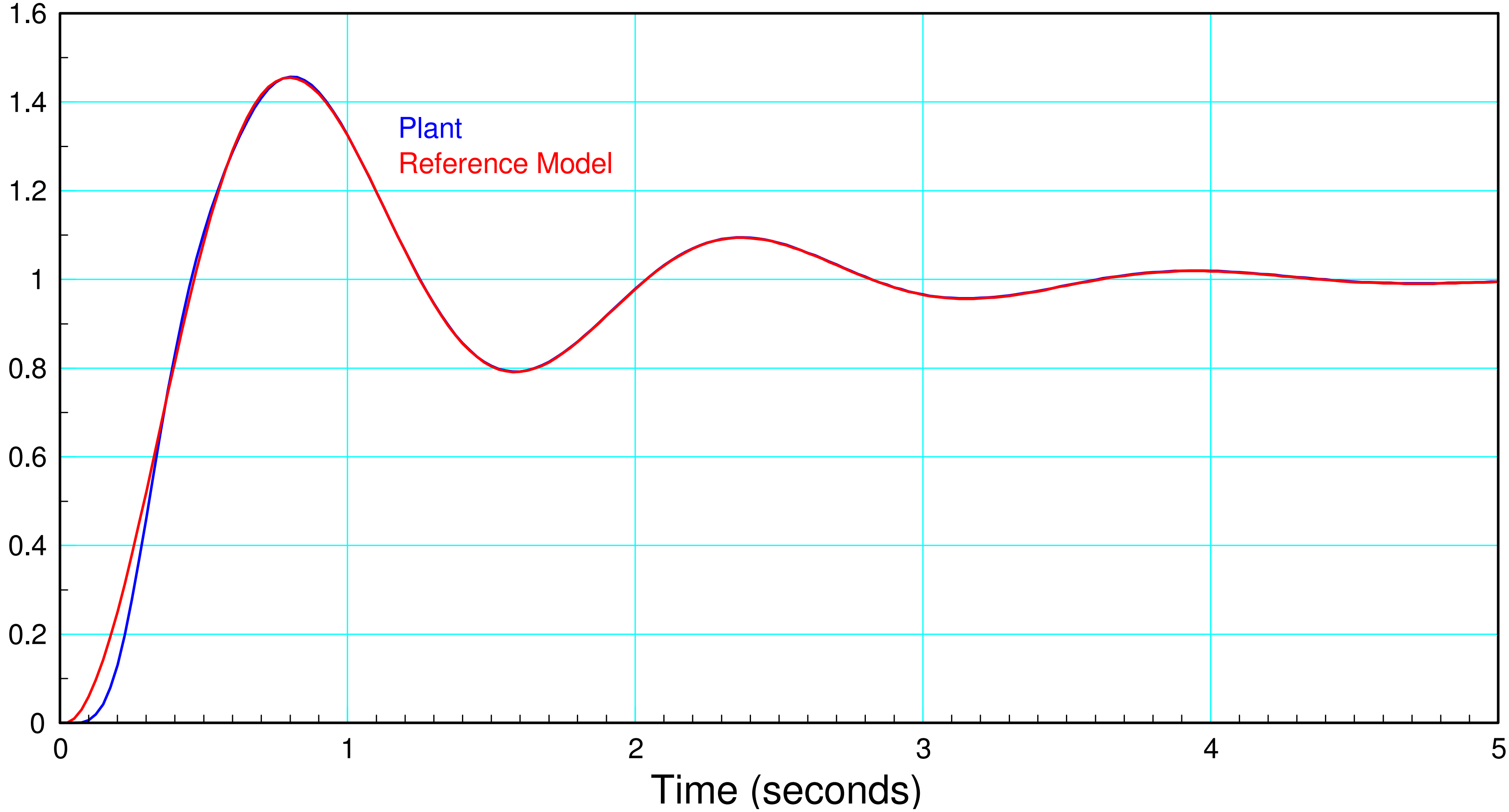
Step response with $Q = 1e6\, C_z^T C_z$



Step Response with $Q = 1e9\ C_z^T C_z$



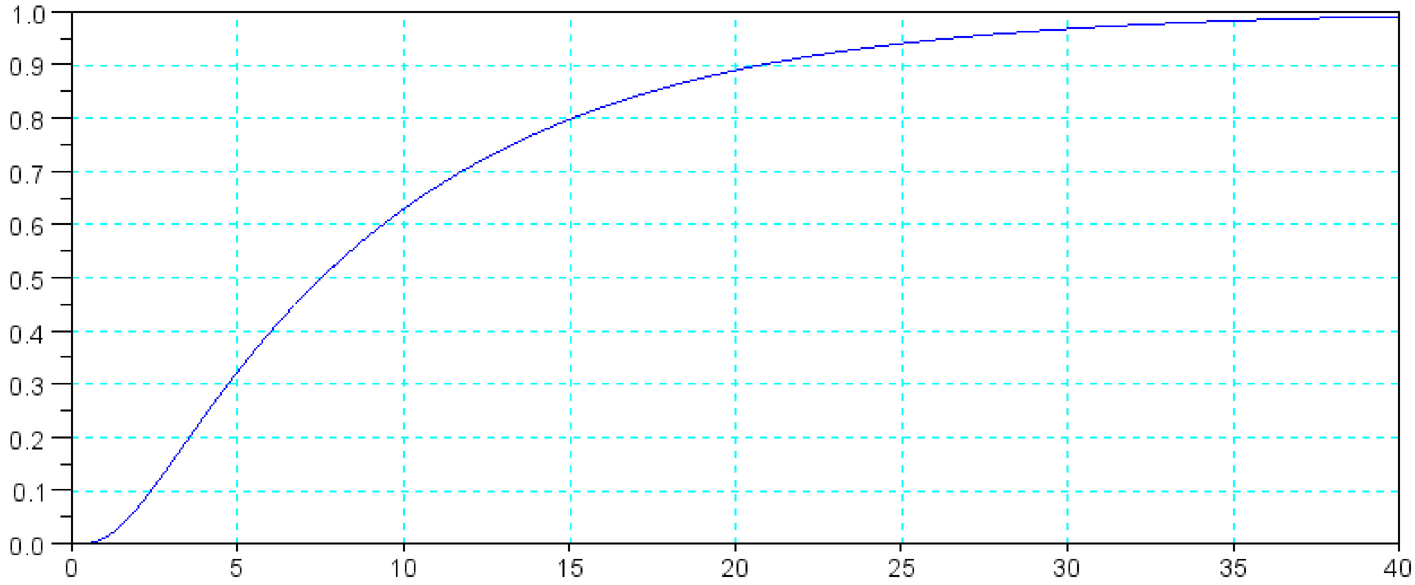
Step Response with Q = 1e12 $C_z^T C_z$



Sidelight

You're trying to make the system in a way it doesn't want (heat equation oscillating and responding quickly). This results in large control gains and large inputs.

It works better if you try to make the system behave the way it wants. If you look at the open-loop step response, you see how it *wants* to behave:

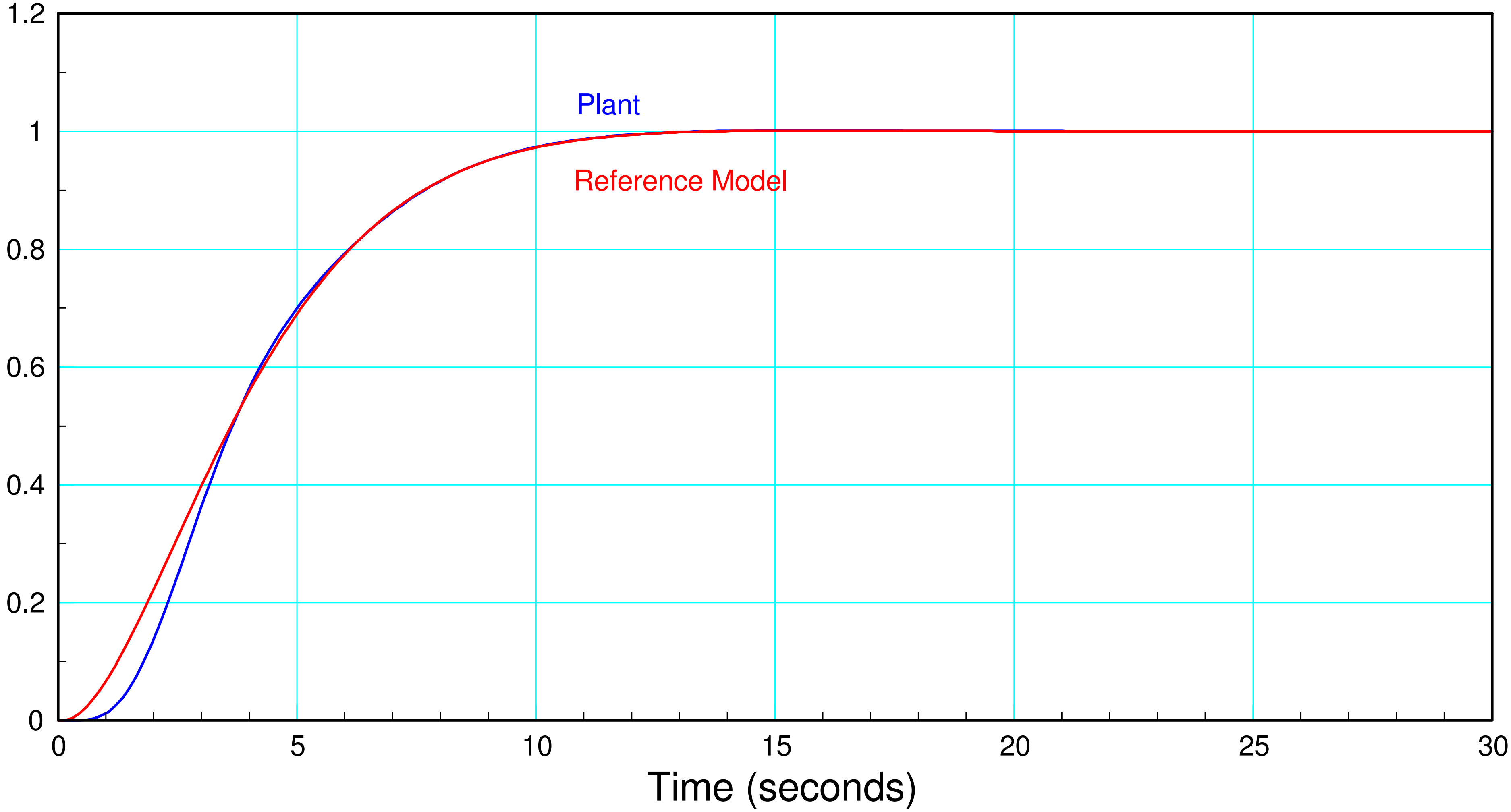


Make the system a bit faster (10 second settling time) and damping ratio 0.8

$$y_m = \left(\frac{0.2}{s^2+0.8s+0.2}\right)Ref$$

```
K7 = lqr(A7,B7,Q7*1e3,R7);
```

```
2.110   6.448   16.072   34.984 31.622 -55.521  -36.011
```

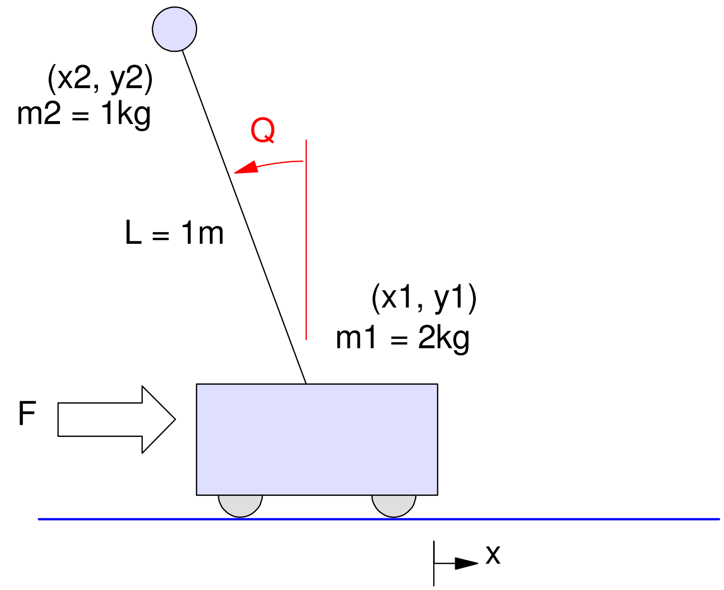


Plant
Reference Model
1.2
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20
25
30
Time (seconds)

Example 2: Cart & Pendulum System

Use LQG/LTR Techniques for the cart and pendulum system

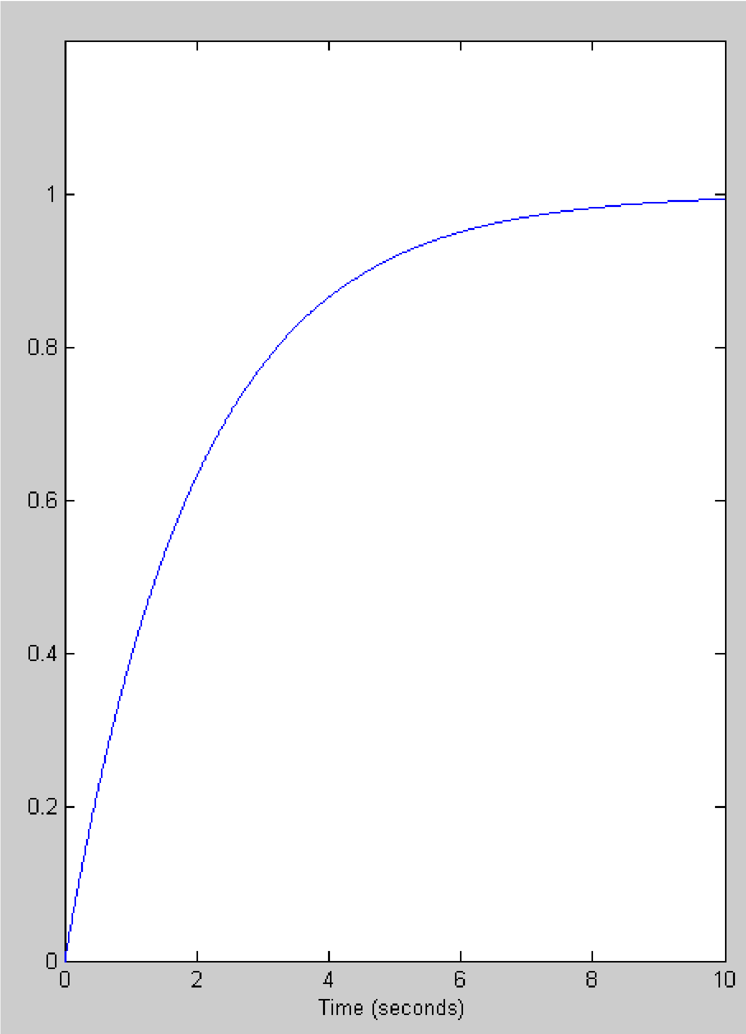
$$s\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -4.9 & 0 & 0 \\ 0 & 14.7 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ -0.5 \end{bmatrix}F$$



Step 1: Define the reference model

- 8 second settling time
- No overshoot

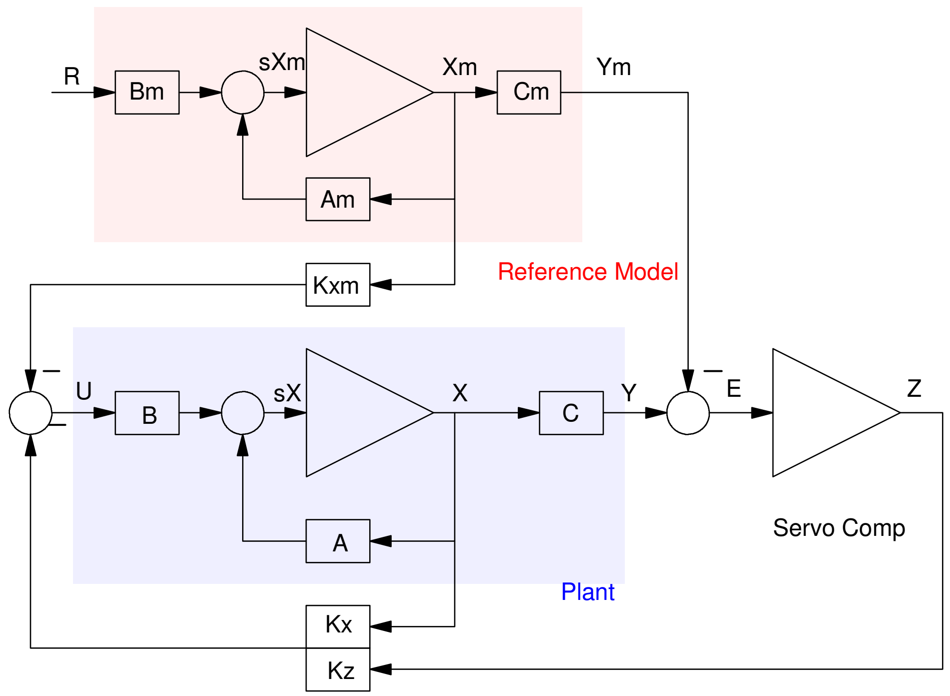
```
%Reference Model

Gm = zpk([],[-0.5],1);

DC = evalfr(Gm,0);
Gm = Gm / DC;
X = ss(Gm);
Am = X.a;
Bm = X.b;
Cm = X.c;

[n,m] = size(Am);

X = zeros(4,1);
Xm = zeros(n,1);
```



Step 2: Find the feedback gains

Create the augmented system

```
Aa = [ A, zeros(4,1), zeros(4,n) ;
       C, 0, -Cm;
       zeros(n,4), zeros(n,1), Am];
Ba = [B; 0; zeros(n,1)];
Bar = [zeros(4,1); 0; Bm];

Ca = [0*C, 1, 0*Cm];
```



Find the feedback gains

```
Q = Ca' * Ca;
R = 1;

Ka = lqr(Aa, Ba, Q*1e2, 1);

Kx = K7(1:4);
Kz = K7(5);
Km = K7(6:5+n);
```

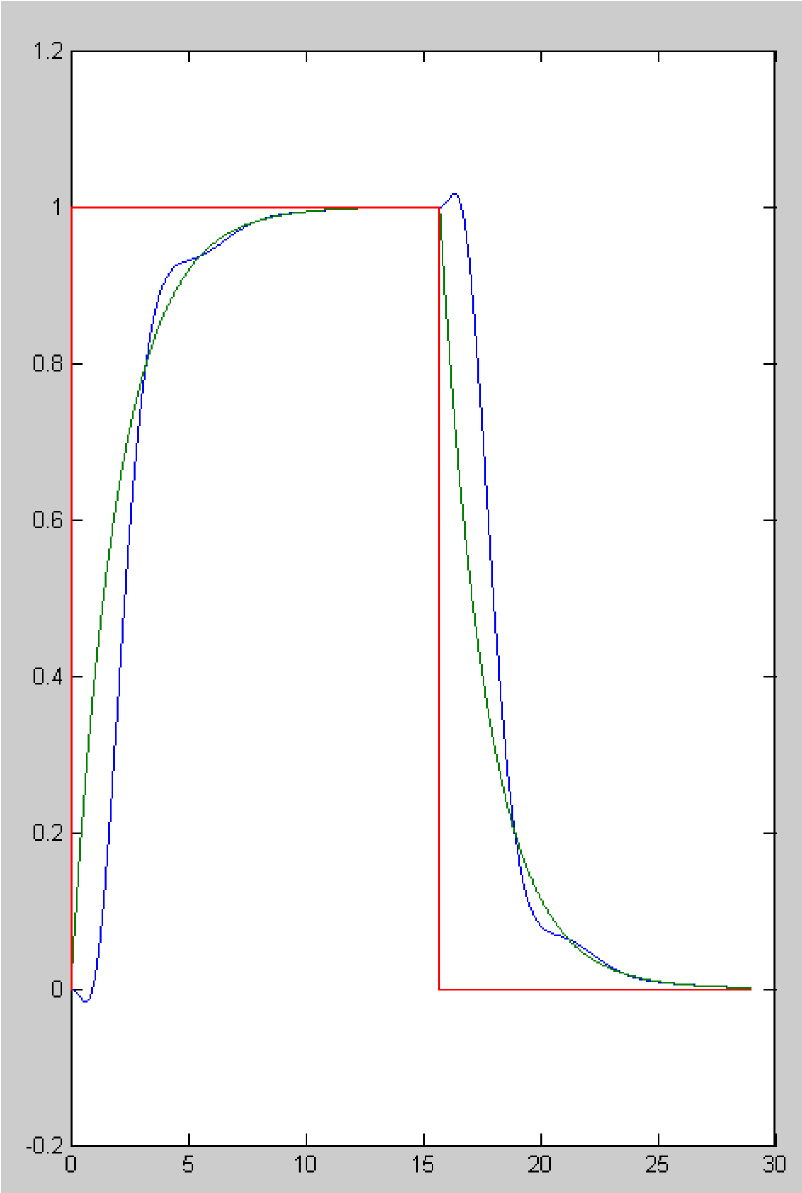
No need to adjust the DC gain (The servo compensator forces Y = Ym at DC)

Main Loop

```
while(t < 29)
 Ref = sign(sin(0.2*t));
 U = -Km*Xm - Kx*X - Kz*Z;

 dX = CartDynamics(X, U);
 dXm = Am*Xm + Bm*Ref;
 dZ = X(1) - Cm*Xm;

 X = X + dX * dt;
 Xm = Xm + dXm * dt;
 Z = Z + dZ*dt;

 t = t + dt;
 n = mod(n+1, 5);
 if(n == 0)
    CartDisplay(X, [Cm*Xm;0;0;0], Ref);
 end
 y = [y ; X(1), Cm*Xm, Ref];
end
```

Response #1

$$G_m = \left(\frac{0.5}{s+0.5}\right)$$



The plant tracks the reference model at DC

- Servo compensator assures that

Poor tracking

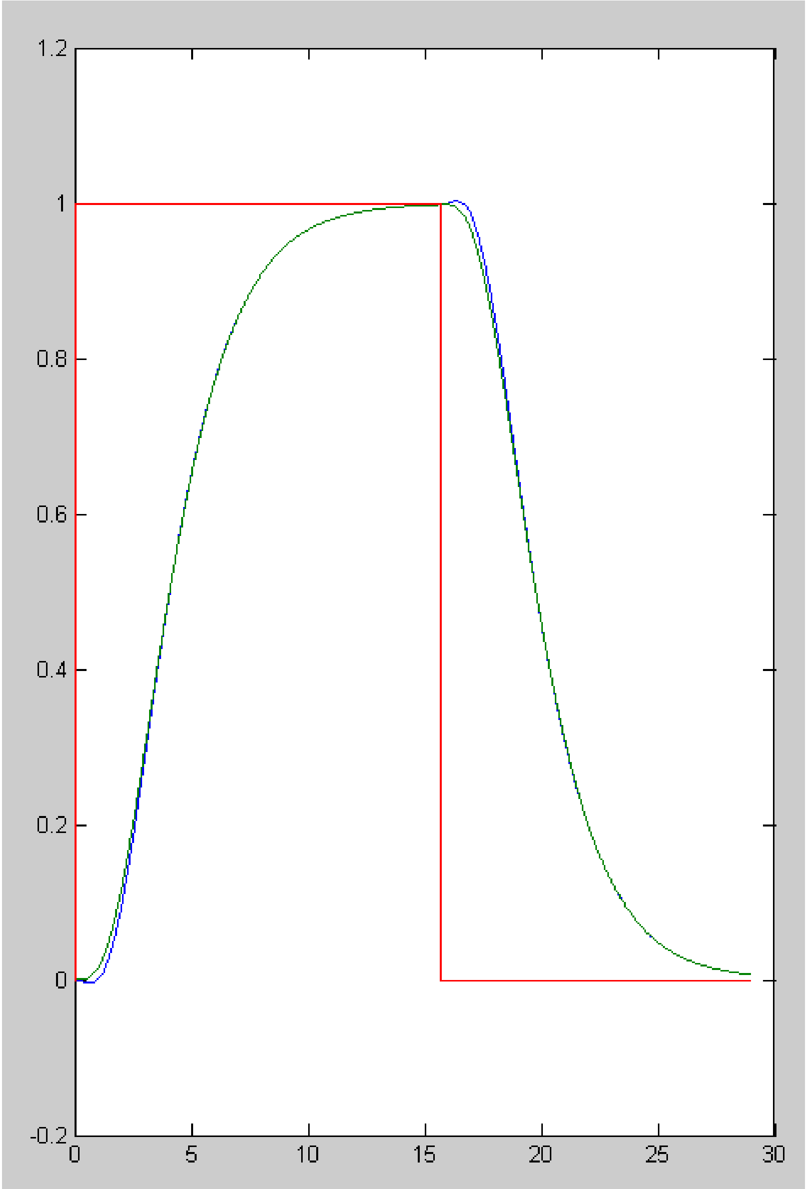
- You're trying to make a 4th-order system behave like a 1st-order system

Response #2:

$$G(s) = \left(\frac{-}{(s+0.5)(s+1)(s+1.2)(s+1.4)} \right)$$

Much better tracking

- Make a 4th-order system behave like a 4th-order system

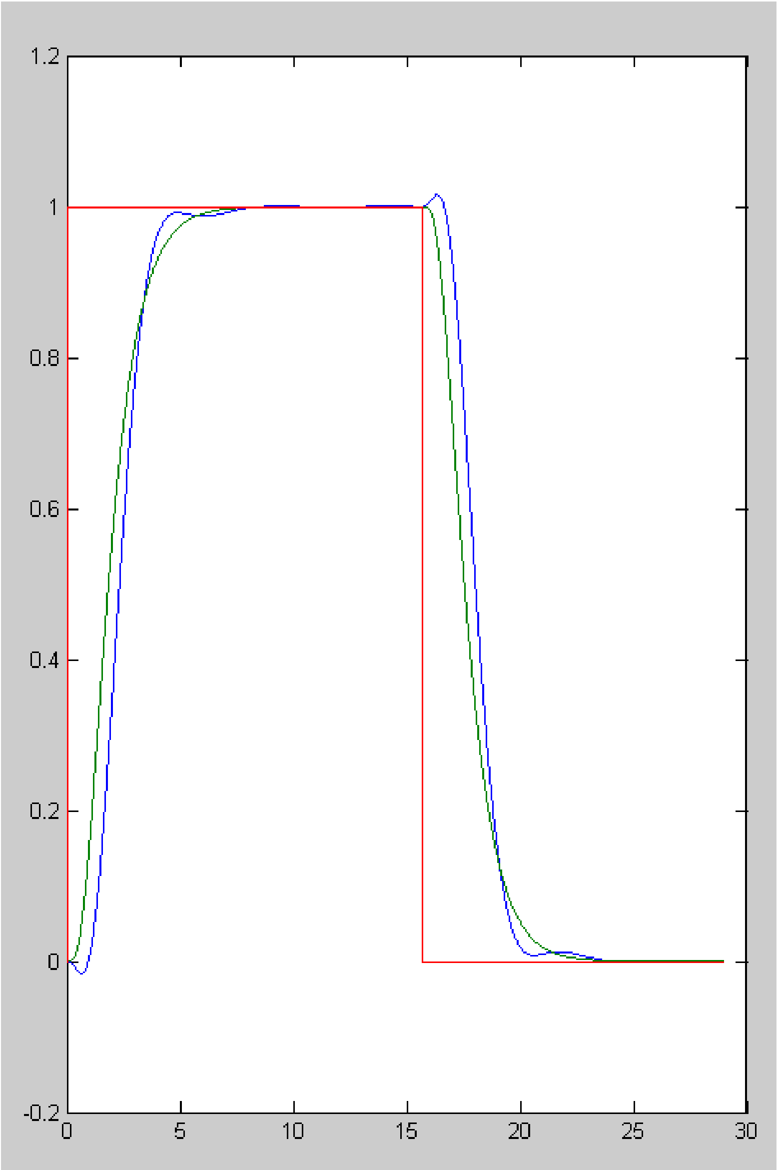


Response #3:

$$G_m = \left(\frac{-}{(s+1)(s+2)(s+3)(s+4)} \right)$$

Speed up the reference model

- Plant tries to follow
- Larger Q will result in better tracking

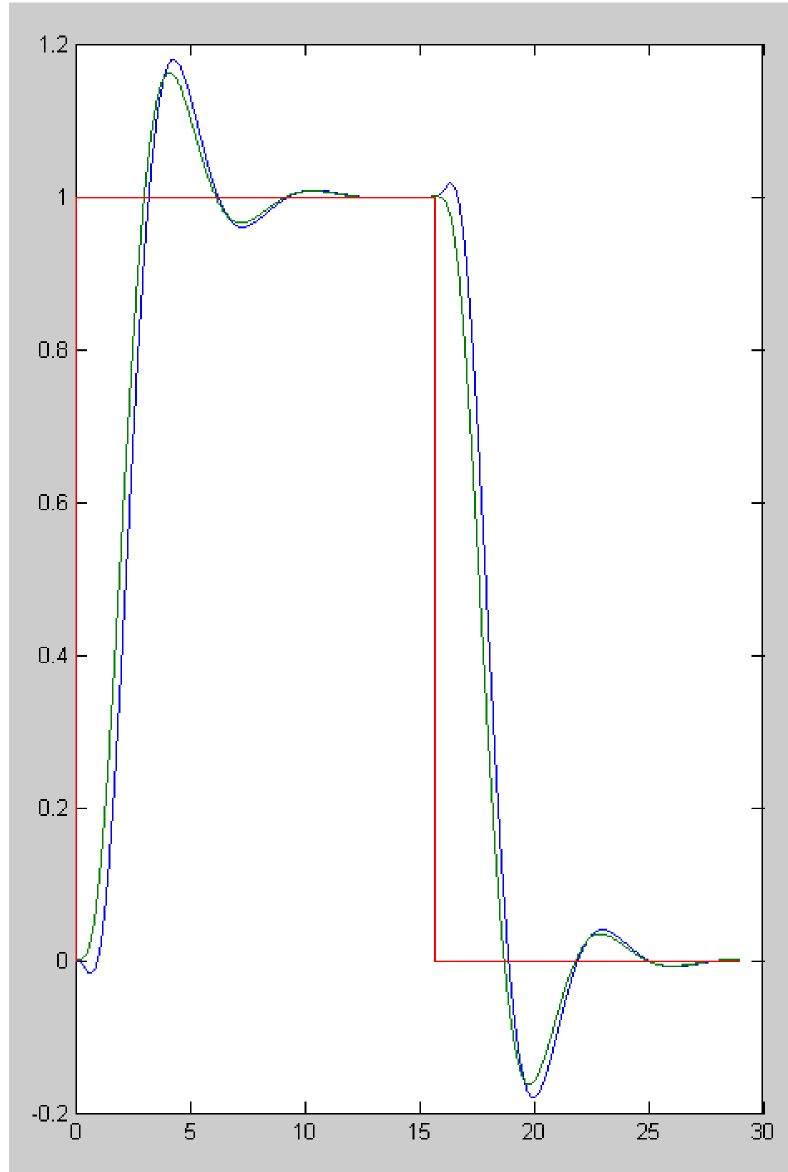


Response #4

$$G_m = \left(\frac{-}{(s+0.5+j)(s+0.5-j)(s+2)(s+3)}\right)$$

By adjusting the reference model, you can get whatever response you want

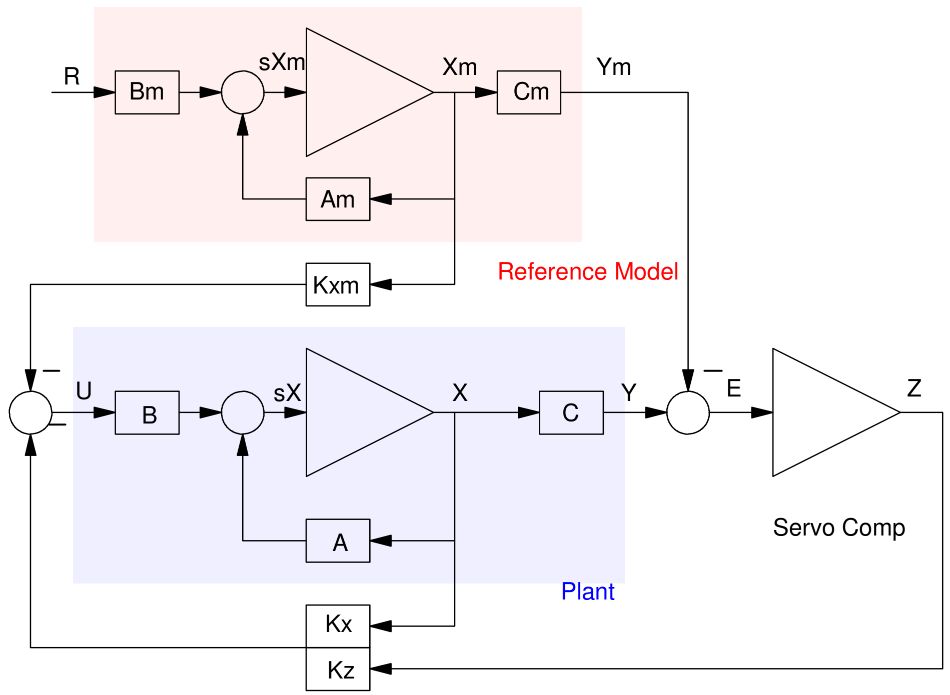
- Overdamp
- Underdamped
- As long as you don't try to make the system too fast



Summary

LQG/LTR is another way to design feedback controllers

- The reference model defines how the system should behave
- The control law tries to make the plant behave like the reference model
- A servo compensator forces the two outputs to match at DC



With a servo compensator, the DC gain of the plant matches the DC gain of the reference model