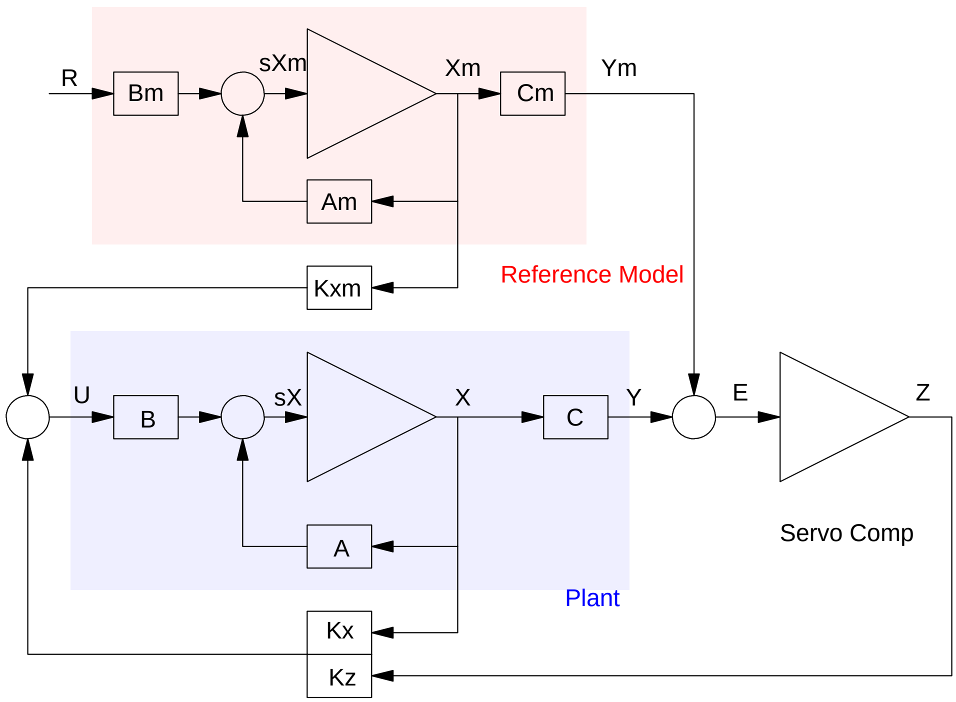
LQG/LTR with a Servo Compensator

LQG/LTR is a convenient way to force a plant to behave like a reference mode. The DC gain was slightly off, however, and required a pre-gain to compensate. Instead, if the set point is a constant, you can add a servo compensator at the output. This forces the error to zero as well as assures perfect tracking at DC.



A Servo Compensator is Added to the LQG/LTR Design to Assure Tracking

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

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

If you find feedback gains which force Z to zero, you force the plant to behave like the reference model.

Example: 4th-Order Heat Equation

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

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

Here, an underdamped system is used just to challenge the design method. First generate the augmented system in SciLab (or MATLAB)

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]

           plant       servo   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
   - - - - - - - - - - - - - - - - - - -
    0    0    0    1 :    0 : - 1     0
   - - - - - - - - - - - - - - - - - - -
    0    0    0    0 :    0 :   0     1
    0    0    0    0 :    0 : - 17  - 2
```

```
-->B7 = [B;0;0;0]
-->Br = [zeros(5,1);Bm]
-->C7 = [   0  0  0  1  0  0  0;
            0  0  0  0  1  0  0 ]

        0   0   0   1   0   0   0      plant output y
        0   0   0   0   0   1   0      model output

--> D7 = [0  ;  0  ]

        0
        0
```

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

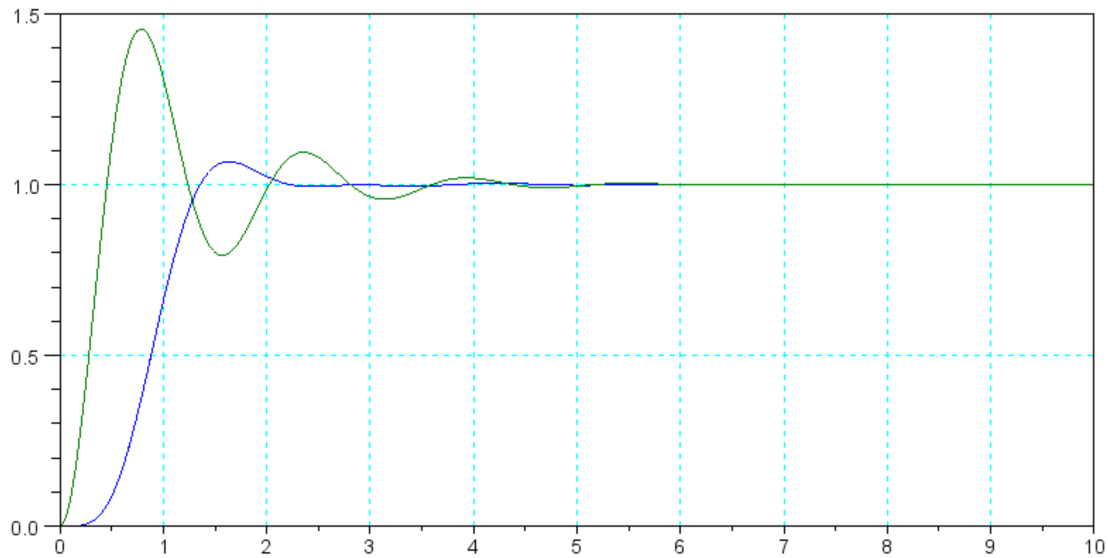
```
-->K7 = lqr(A7, B7, Q7*1e6, R7)

 7.3587544    41.793142    181.61608    618.11264   1000.   -125.24099   -71.358383
```

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
```

Plotting the response to a step in R:

```
-->G7 = ss(A7-B7*K7,Br,C7, D7);
-->t = [0:0.001:1]';
-->y = step(G7,t);
-->plot(t,y)
```

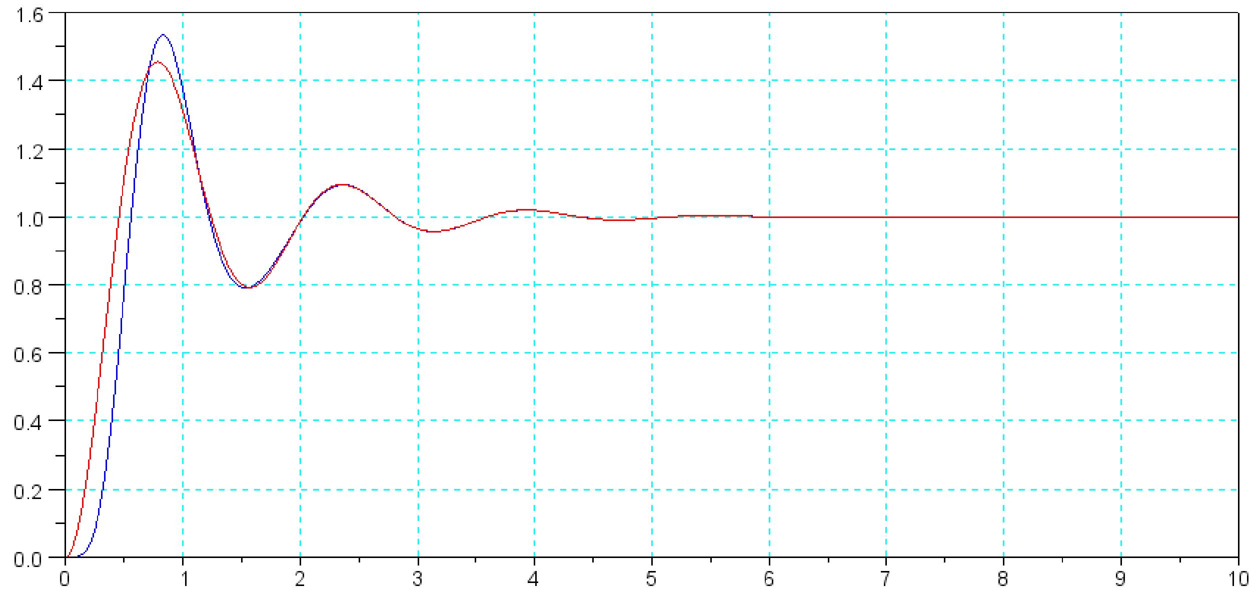


Step Response of the Plant (blue) and Reference Model (green) for Q = 1e6 $C^T C$

The plant (blue) doesn't track the reference model (green) all that well. This is the best tradeoff between tracking (Q) and input (R) for this Q and R however. For better tracking, increase Q:

Repeat with Q = 1e9 $C^T C$

```
-->K7 = lqr(A7,B7,Q7*1e9,R7);

19.469654   228.47301   1872.6106    10909.359   31622.777   -8360.5346   -1715.4934


-->eig(A7-B7*K7)

  - 2.5224807 + 7.3316288i       Dominant Pole:  Faster (better tracking)
  - 2.5224807 - 7.3316288i
  - 8.1906629
  - 6.6170147 + 4.5208949i
  - 6.6170147 - 4.5208949i
  - 1. + 4.i                     Reference Model
  - 1. - 4.i

-->G7 = ss(A7-B7*K7,Br,C7,D7);
-->y = step(G7,t);

-->plot(t,y)
```

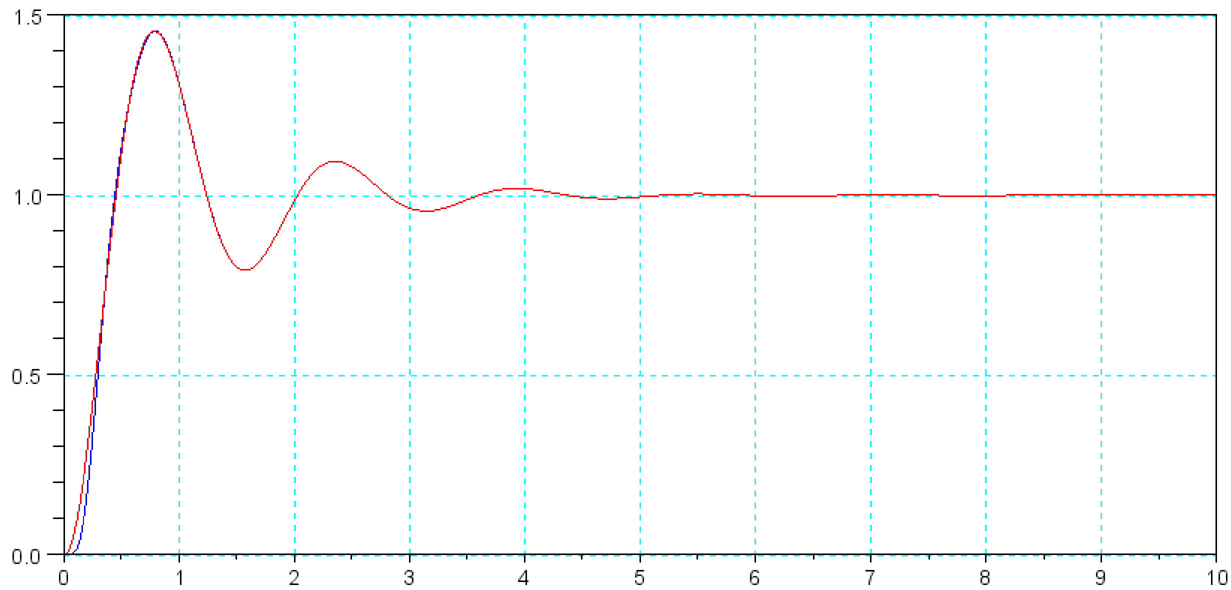


Step Response of the Plant (blue) and Reference Model (red) for Q = 1e9 $C^T C$

Note that with a higher weighting on Q (the error in the servo state), tracking gets better. The input gets larger too, however. For even better tracking, increase Q further:

Repeat with Q = 1e12 $C^T C$

```
-->K7 = lqr(A7,B7,Q7*1e12,R7);

44.674955    1087.2757    17425.846    186212.9   1000000.   -184213.15   -17702.505


-->eig(A7-B7*K7)

  - 4.9338452 + 14.959796i
  - 4.9338452 - 14.959796i
  - 15.969812
  - 12.918726 + 9.2443637i
  - 12.918726 - 9.2443637i
  - 1. + 4.i
  - 1. - 4.i

-->G7 = ss(A7-B7*K7,Br,C7, D7);

-->y = step(G7,t);

-->plot(t,y)
```



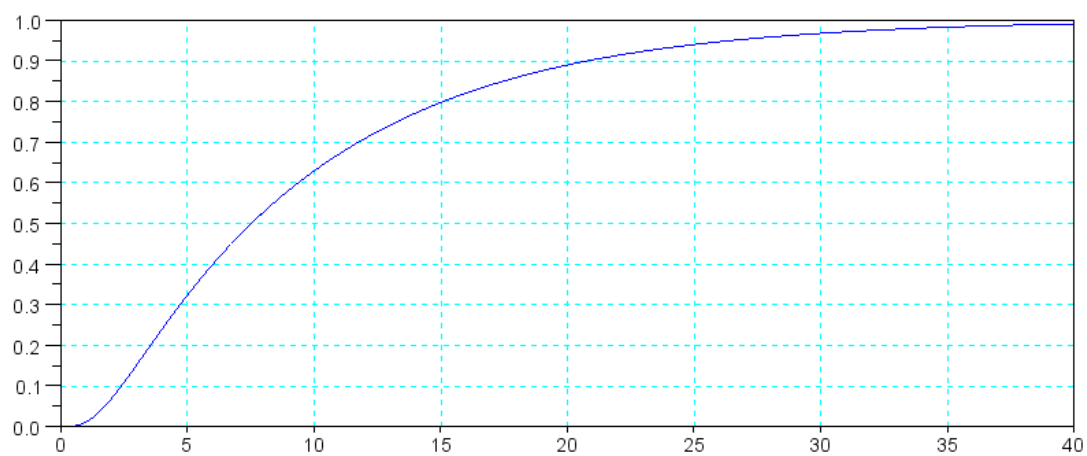
Step Response of the Plant (blue) and Reference Model (red) for Q = 1e12 CTC

Now the system is tracking the reference model fairly well. The cost is high feedback gains.

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.

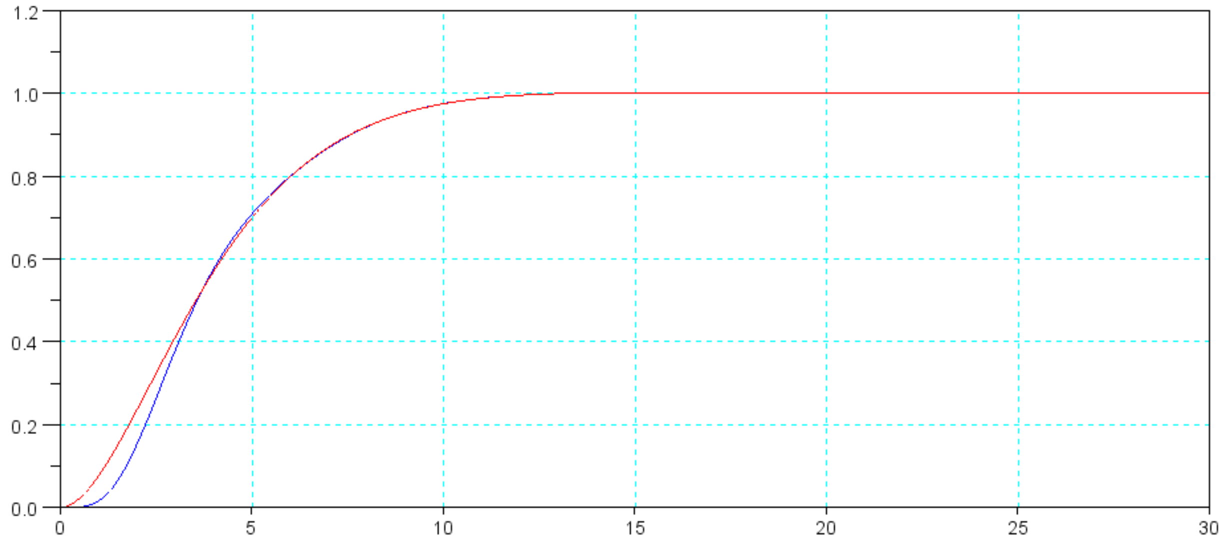
It works better if you try to make the system behave the way it wants. If you look at the open-loop step response, you see how it *wants* to behave:



Open-Loop Response

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

```
-->poly([-0.4+j*0.2,-0.4-j*0.2])

    1.    0.8    0.2

-->Am = [0,1;-0.2,-0.8];
-->Bm = [0;0.2];
-->t = [0:0.01:30]';

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

          plant        servo    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
   - - - - - - - - - - - - - - - - - - - -
    0    0    0    1 :   0 :  -1     0
   - - - - - - - - - - - - - - - - - - - -
    0    0    0    0 :   0 :   0     1
    0    0    0    0 :   0 : -0.2  -0.8

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

2.1105308   6.4482317   16.072671    34.984873  31.622777  -55.521102  -36.011681


-->eig(A7-B7*K7)

  - 3.5431217
  - 2.0849937 + 0.2969885i
  - 2.0849937 - 0.2969885i
  - 0.6987109 + 1.2345238i
  - 0.6987109 - 1.2345238i
  - 0.4 + 0.2i
  - 0.4 - 0.2i

-->G7 = ss(A7-B7*K7,Br,C7, D7);

-->y = step(G7,t);

-->plot(t,y)

-->xgrid(4)
```



Reference Model (red) and plant output (blue)

Now the gains are much more reasonable and the tracking is good.

It helps to know how a system wants to behave when choosing the reference model.