

Using methods learned in class, we found the open-loop transfer function of the Motomatic to be:

$$\frac{V_{\text{out}}}{V_{\text{in}}} = \frac{210.9}{s^2 + 5.27s}$$

1. In HW#9, we used Matlab's `tf2ss()` command to find a continuous-time state space representation of the form $\dot{x} = Ax + Bw, x(0^+)$ for this system. We also verified that the resulting system was completely controllable. Now, we want to use optimal control theory (LQR), to find an optimal feedback gain, $K(\infty)$, for this system.

- a) Use Matlab's `lqr()` command in the form $[K,S,E] = \text{LQR}(A,B,Q,R)$ to find the optimal value of the feed back control, $w = -Kx$, that will minimize the linear quadratic cost function, $J = \int_0^{\infty} 1/2 x^T Q x + 1/2 w^T R w dt$ using a 2x2 identity matrix for Q and R=1.

Ans: Using the same A and B from HW#9, we have

>> `[K,S,E] = lqr(A,B,Q,R)`

K =

0.2773 1.0000

S =

0.2773 1.0000
1.0000 5.5473

E =

-5.3608
-0.1865

>>

So, $w = -K(\infty)x = -0.2773x_1 - x_2$

- b) What are the closed-loop eigenvalues (E) for your LQR system? Is the closed-loop system stable?

Ans: E =

-5.3608
-0.1865

Yes, they are in the LHP

- c) Go back and repeat part a) but this time substitute $A + \alpha I$ for A where $\alpha = 16$. Now check the eigenvalues of the closed-loop system. Are the real parts less than -16?

```
>> [K,S,E] = lqr(A+16*eye(2),B,Q,R)
```

```
K =
```

```
53.5066 856.8526
```

```
S =
```

```
1.0e+004 *
```

```
0.0054 0.0857  
0.0857 2.2944
```

```
E =
```

```
-10.7768  
-15.9998
```

```
>> eig(A-B*K)
```

```
ans =
```

```
-31.9998  
-26.7768
```

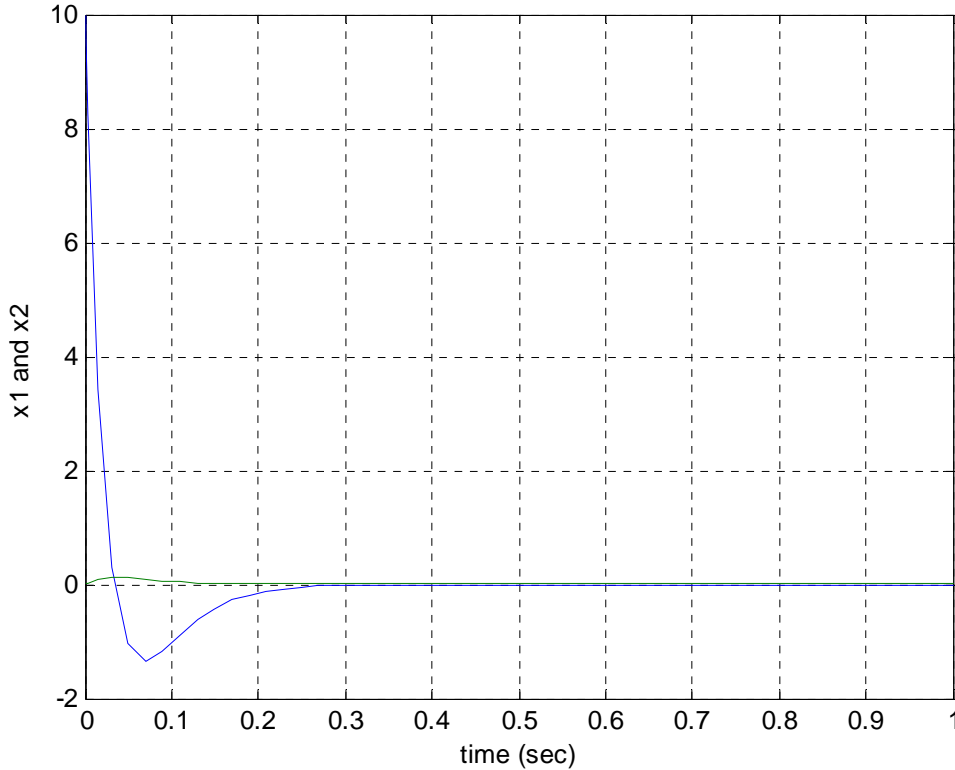
```
>>
```

Yes, the real parts of the eigenvalues of $A-BK(\infty)$ are much less than -16

- d) Simulate your LQR closed-loop state-space system from part c) using Simulink using an initial state of $x(0) = [10 \ 0]^T$ (hint: you will need to connect a 2x1 gain block -K to the output of your Simulink model and feed the output of this block it back to the input)

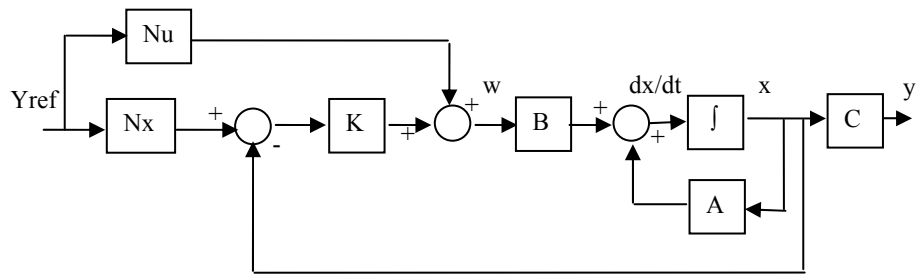
Ans: The following is a Simulink simulation using the optimal $w = -K(\infty)x$ feedback matrix from part c)

EE/MFS 599: Optimal Response with the initial state = [10;0] for HW10



As we can see, the states “settle” in about 0.2 seconds or so. Better than HW9 but we would expect this because the closed-loop eigenvalues are much deeper into the RHP.

- The ‘R’ in LQR stands for regulator (dRoop). Most industrial control applications require that we track a reference signal, not just regulate (droop). To this end, consider the tracking architecture:



- For your servo system state space model, what are the dimensions of y_{ref} , N_x , and N_u ?

Ans y_{ref} is a p -vector, N_x is $n \times p$ and N_u is $p \times m$. Recall we assumed the same number of inputs as output ($m=p=1$ in our case)

- Use the equations given today in class to find the values of y_{ref} , N_x , and N_u

Ans:

```
>> inv([C 0;A B])*[1;0;0]
```

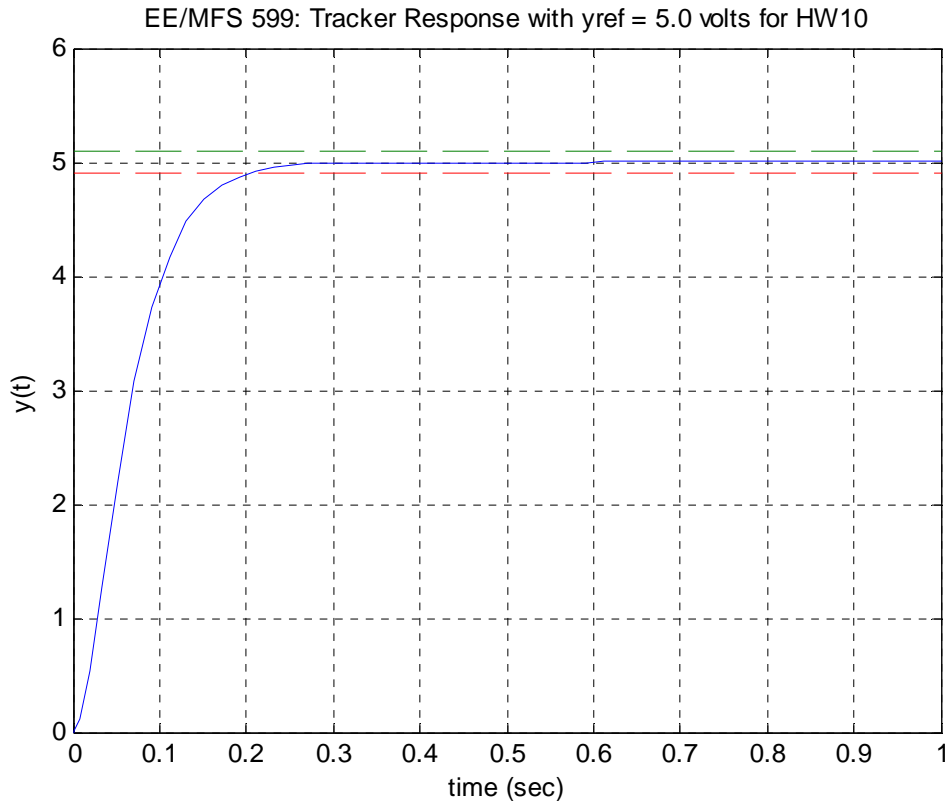
ans =

0
0.0047

0
So, $Nx=[0 \ 0.0047]^T$ and $Nu = [0]$

c) Simulate your tracker architecture in Simulink using an initial state of $x(0) = [0 \ 0]^T$ and a $y_{ref} = 5.0$ volts. How long does it take $y = x_1 = V_{out}$ to settle within 2% of its final value? What is the overshoot on y ?

Ans: Using the tracking architecture defined above, we obtain the following output from Simulink:

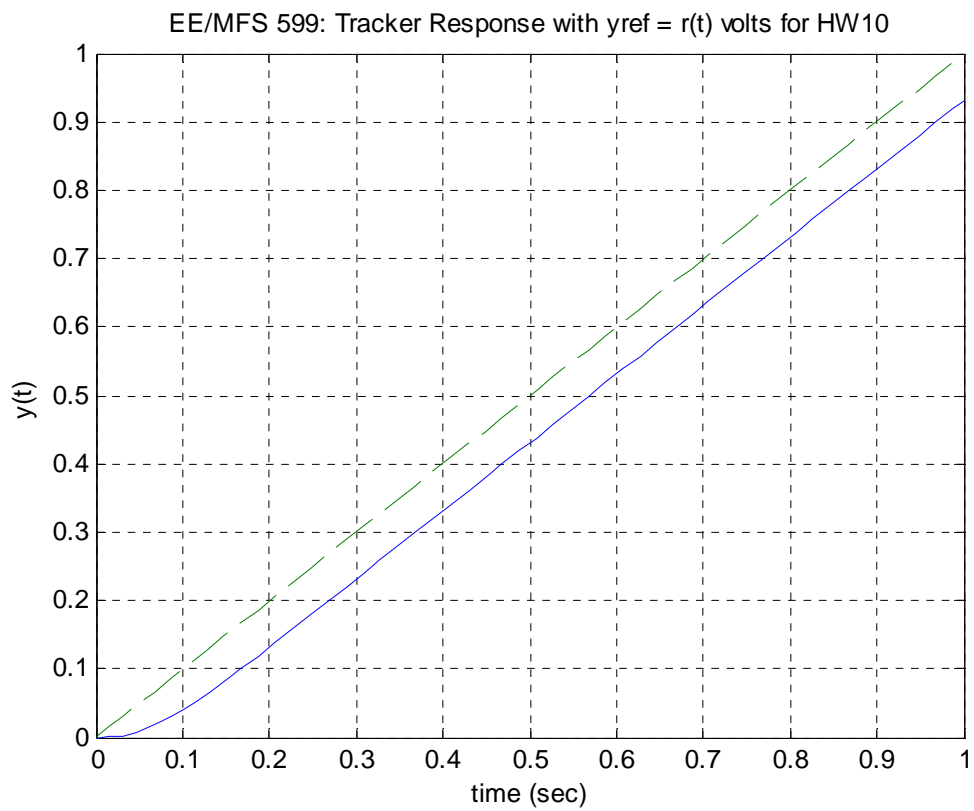


Note that there is no perceptible overshoot and the output settles at $t_s = 0.2$ seconds

d) Now, let $y_{ref} = r(t) =$ a unit ramp. Does your tracker follow a ramp? What is the steady-state error for a ramp?

Ans:

The following is the Simulink simulation for $y_{ref}=r(t)=$ ramp:



The tracker does follow a ramp but not perfectly. The steady-state error is 0.0686