Solutions For Lab 2: Model Identification and Transient Response of a DC Servo Motor System

1 Procedure

3.1 Model Identification The Simulink model Lab2_Identification was run. A unit step function was given as an input to the DC model block producing the following trajectories:

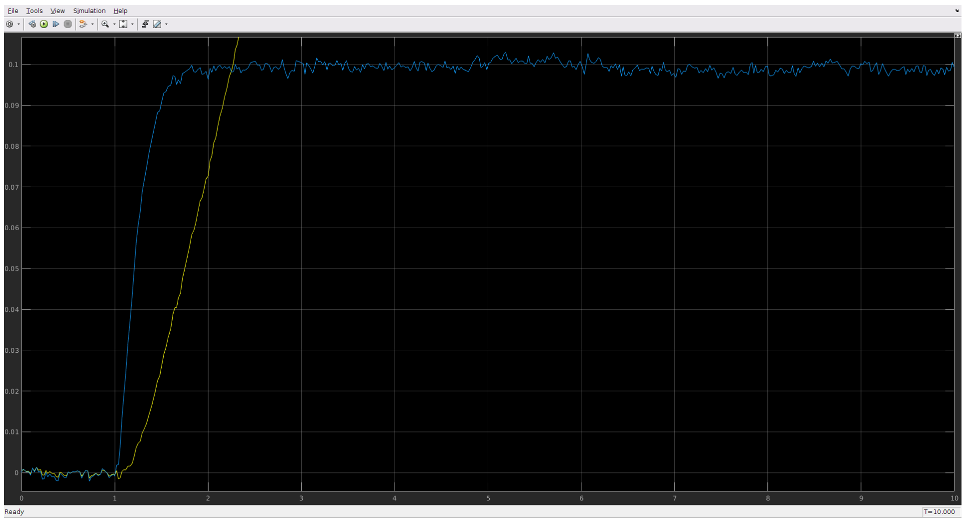

Figure 1: Typical impulse response for the DCservo model. Omega in blue.

Our model of the DC servo system has an input-speed transfer function

$$\Omega(s) = \frac{K_s}{\tau s + 1} \tag{1}$$

and input-position transfer function

$$\Theta(s) = \frac{K_s}{s(\tau s + 1)} \tag{2}$$

To determine K_s, we find the steady state value of the step response. By inspection, it seems to be about.1. Noise makes a more precise estimate difficult. We can do some processing to come up with a better estimate. The model created a data structure in the workspace containing the scope data. We can take an a average over the latter part of the data to estimate the steady-state value:

```
omega = Lab2_Identificationdata.signals(2).values;
avgomega = omega(200:length(omega));
ks = mean(avgomega)
```

which gives a steady state value of about .0993 in this case. Since our input was a unit step function, our estimate for K_s is .0933/1 = .0993. Next, we attempt to estimate τ. First, we find the time at which the response is 63.2% of its final value, or .0628. We can do this by inspection using the scope's magnification cursor:

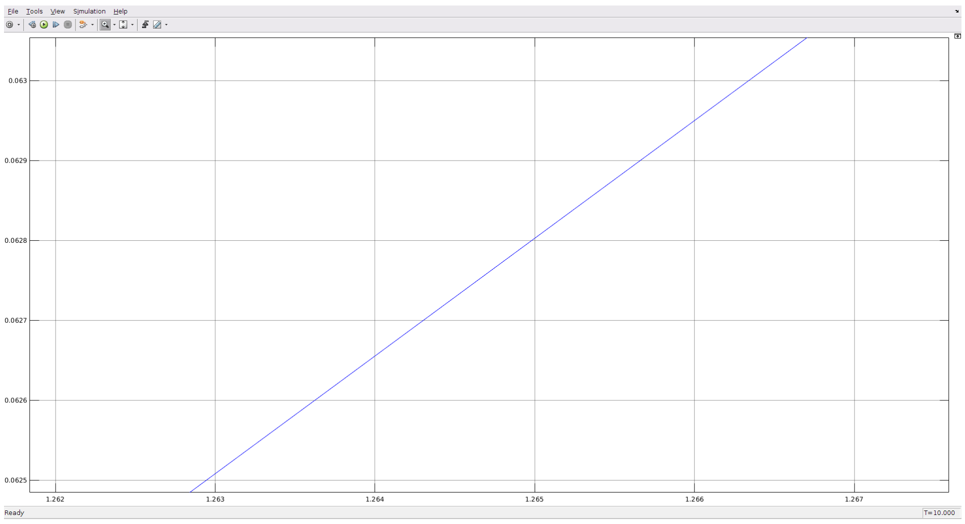

Figure 2: Zooming in on the omega trajectory near the point at which it is equal to 63.2% of the final value.

This time is about 1.265, .265 seconds after the application of the step function. This gives us a value of .265 for τ.

In the Simulink model Lab2_Comparison, we update the transfer function block according the parameters determined for our model of the DC servo system.

We compare the velocity trajectory of our model to that of the DC servo system.

Our model matches the behavior of the system reasonably well, though our model doesn't seem to completely describe the dynamics of the system. Though the curves intersect around one time constant, the model curve is substantially higher before and substantially lower after, until both curves settle at the final value.

It doesn't seem that we can do much better by tweaking parameters, a shorter time constant may reduce error after the curves intersect at the cost of more error before and vice-versa for a longer time constant. We will stick with the paramter values determined.

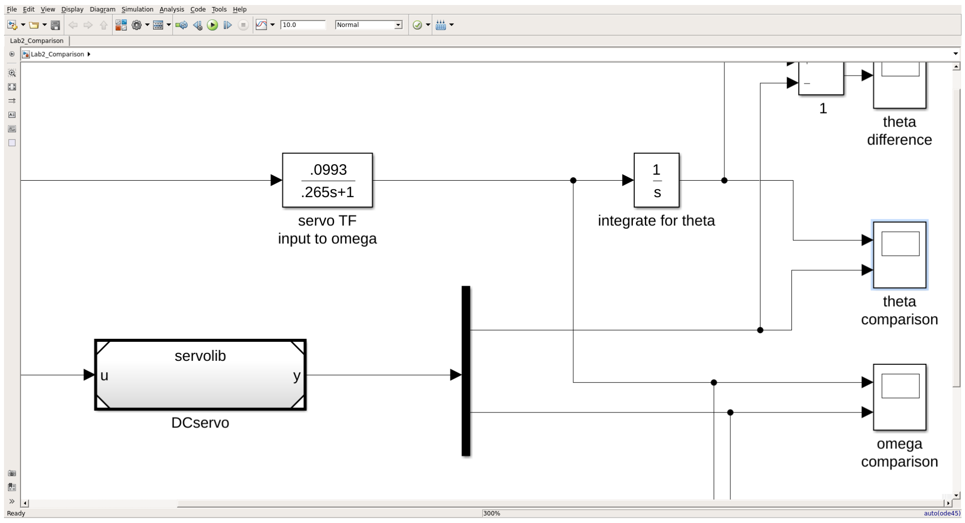

Figure 3: Lab2_Comparison.slx with updated model transfer function block.

3.2 Transient Response

The model Lab2_Transient contains 7 DC servo systems each with a different position feedback gain. We run the model, producing a plot with a family of position trajectories:

2 Exercises

1. From the data of section 3.2, we are able to fill the following table:

Gain	Poles	%overshoot	ζ	ω_n	$T_{settling}$	T_{peak}	T_r
1	-3.67, -1.02,1	0					
2	-3.56,-.2103	0					
4	-3.32, -.451	0					
8	-2.64, -1.14	0					
16	-1.89 ± 1.56i	1					
32	-1.89 ± 2.90i	9					
64	-1.89 ± 4.52i	13					

The gain values were read from the Simulink model diagram. The poles were calculated from our model transfer function. For a system with transfer function $\Theta(s)$, the closed loop transfer function with error feedback gain K is given by:

$$\frac{K\Theta(s)}{1+K\Theta(s)} \tag{3}$$

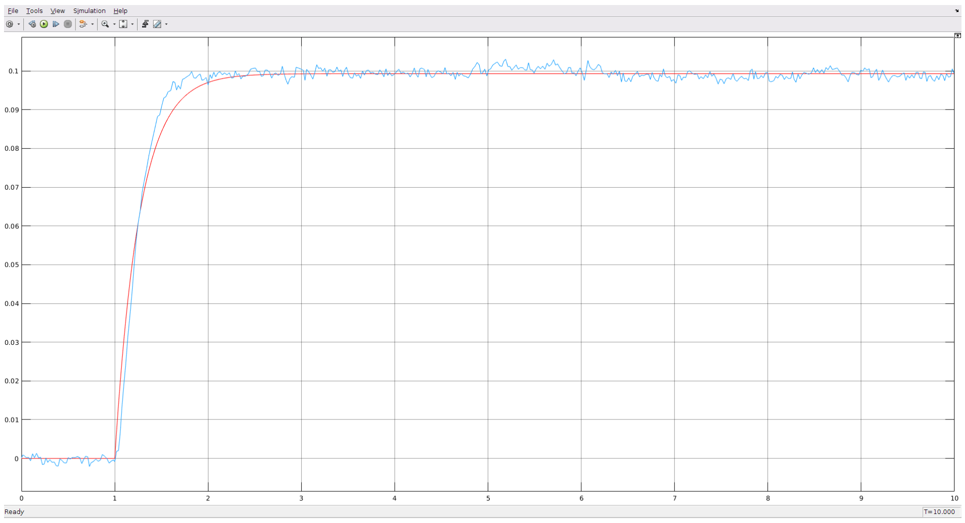

Figure 4: Omega trajectory comparison. Our model in red and the DC servo system in blue.

the following MATLAB script was used to calculate the poles of the closed loop transfer function for each gain.

```
%define constants
ks = .0993;
tau = .265;
K = [1,2,4,8,16,32,64];

%create transfer function object
s = tf('s');
systf = ks/(s*(tau*s + 1));

%for each gain value
for i = 1:length(K)
%find the closed loop transfer function
     fbsystf = feedback(K(i)*systf,1);
     display(['poles for ' num2str(K(i)) ' : '])
     %calculate poles for closed loop system
     pole(fbsystf)
end
```

The percent overshoot was measured by inspection using the scope.

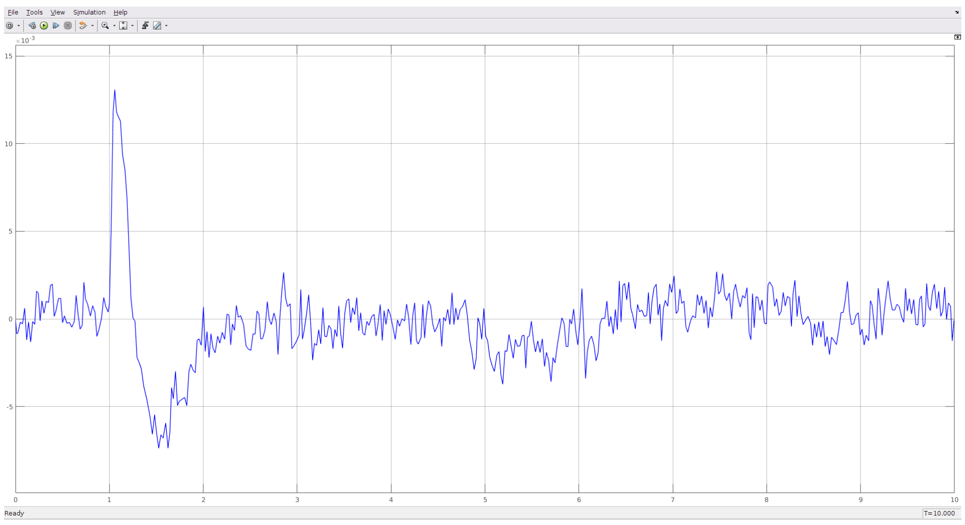

Figure 5: The difference in omega trajectories between our model and the system. While not very large, the difference during the rise is greater than can be explained by noise alone.

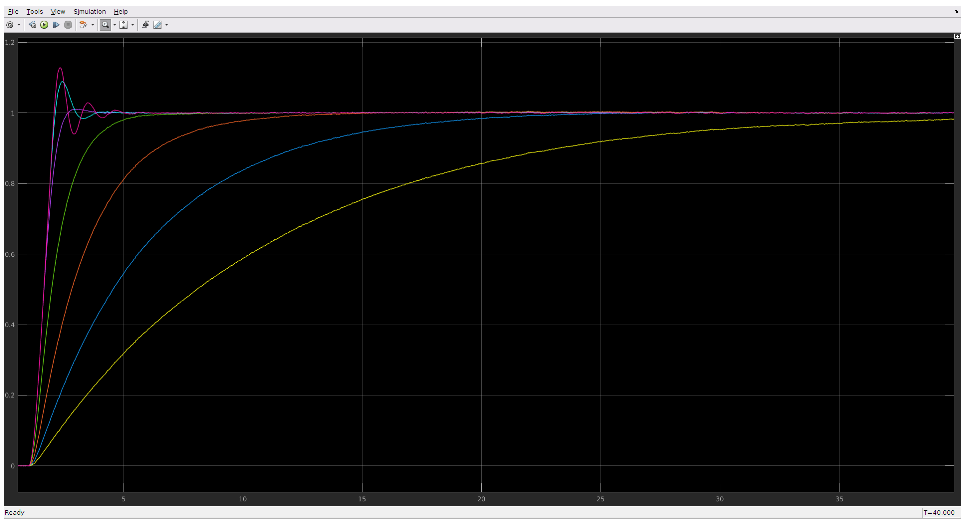

Figure 6: The family of curves produced from the set of feedback gains (1,2,4,8,16,32,64) with their respective colors yellow, blue, red, green, purple, cyan, and magenta