ECE 220-203

Lab 3

Jordan Ditzler

10/14/19

Jordan Ditzler
ECE 220-203

Objective

The objective of this lab is to reinforce our knowledge of LTI systems described by linear constant-coefficient differential equations and how to plot them in MATLAB.

MATLAB Code

See Appendix A.

Results

Part ai:

$$y' + 3y = x \qquad s + 3 = 0 \qquad s = -3$$

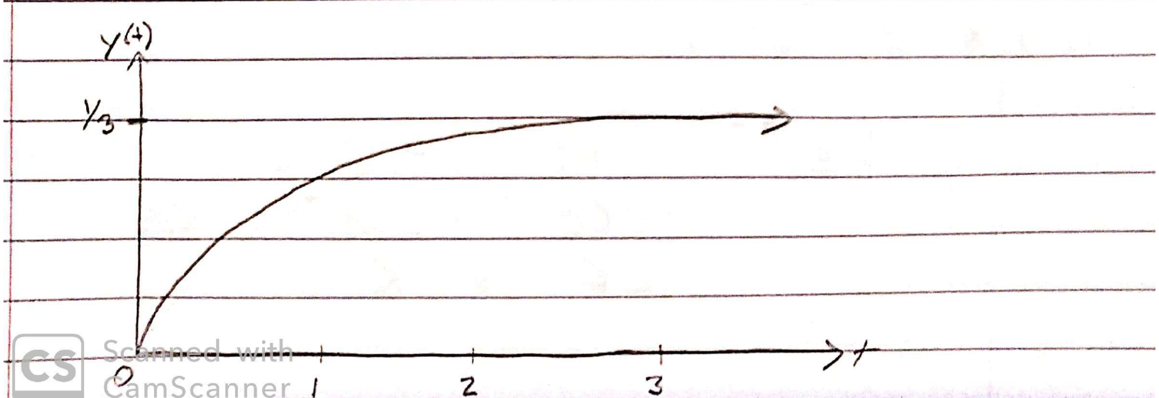
$$y_h = Ae^{-3t} \qquad y_p = \overset{=0}{\cancel{K'}} + 3k = K \Rightarrow K = 1/3$$

$$= 1/3$$

$$y(t) = Ae^{-3t} + 1/3 \qquad y(0) = 0$$

$$0 = A + 1/3 \qquad A = -1/3$$

$$y(t) = (1/3 - 1/3\, e^{-3t})\, u(t)$$



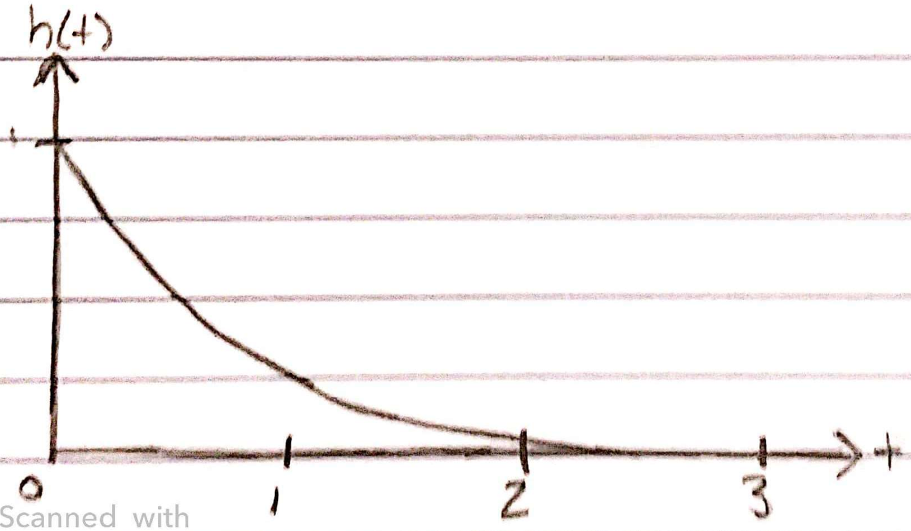
Jordan Ditzler
ECE 220-203

Part aii:

$$y(t) = \left(\tfrac{1}{3} - \tfrac{1}{3}e^{-3t}\right)' u(t)$$

$$= \frac{-3}{-3}e^{-3t} = e^{-3t}u(t)$$

$$h(t) = e^{-3t}u(t)$$

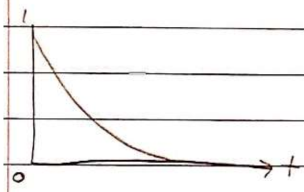


Part aiii:

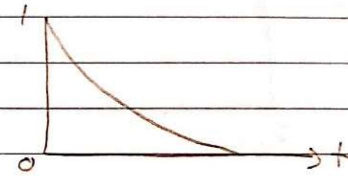
$x(t) = e^{-2t}u(t)$

$h(t) = e^{-3t}u(t)$

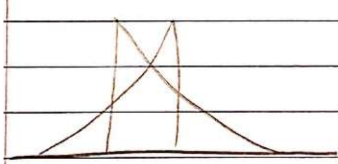
$x(t)$



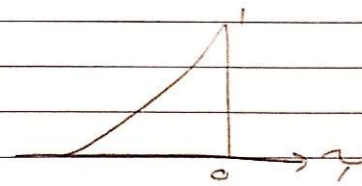
$h(t)$



$t > 0$



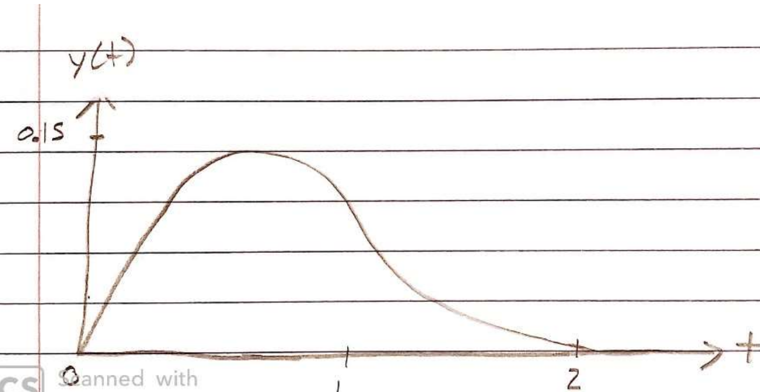
$h(t-\tau)$



$$\int_0^t e^{-2\tau} e^{-3(t-\tau)} = e^{-3t}\int_0^t e^{-2\tau} e^{3\tau}$$

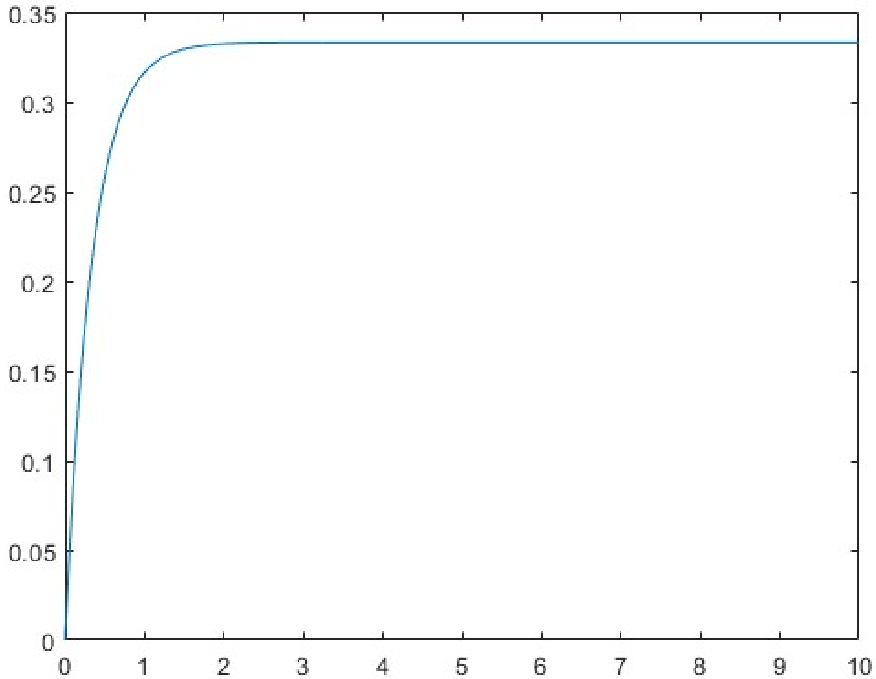
$$= e^{-3t}\int_0^t e^{\tau} = e^{-3t} e^{\tau}\Big|_0^t = e^{-3t}(e^{t}-1)$$

$$= e^{-2t} - e^{-3t}$$

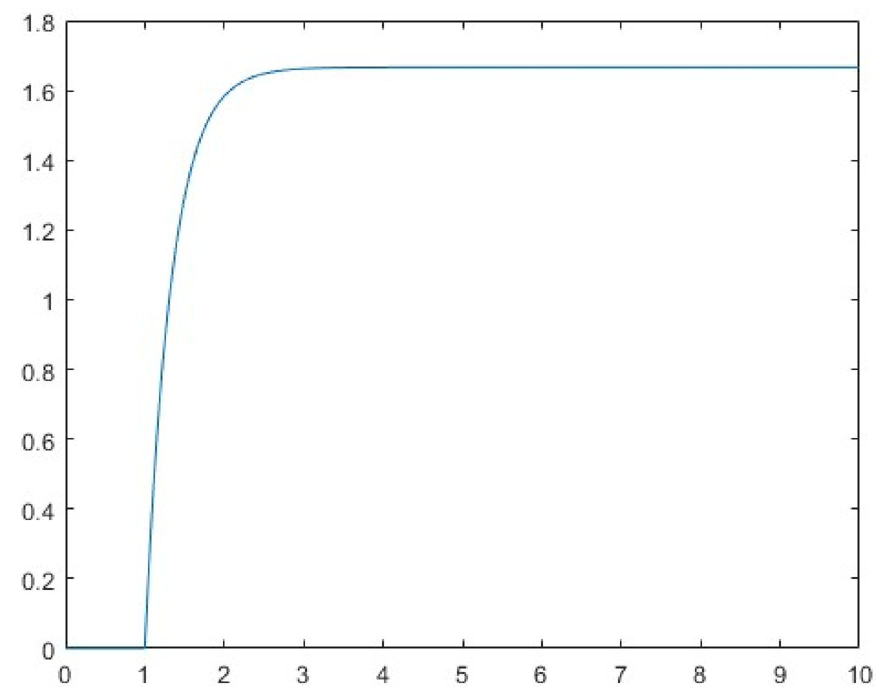


Jordan Ditzler
ECE 220-203

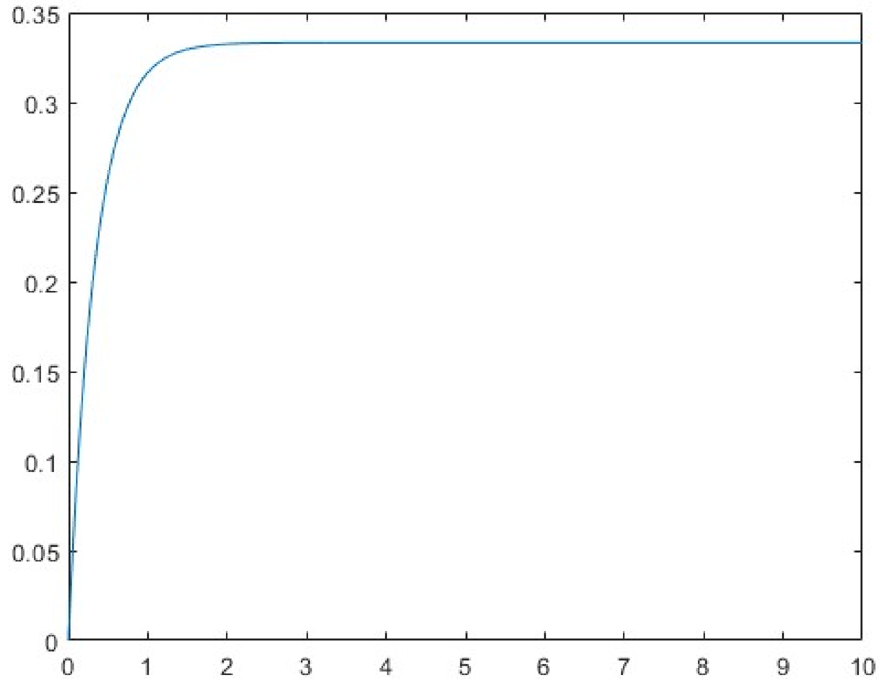
Part b:



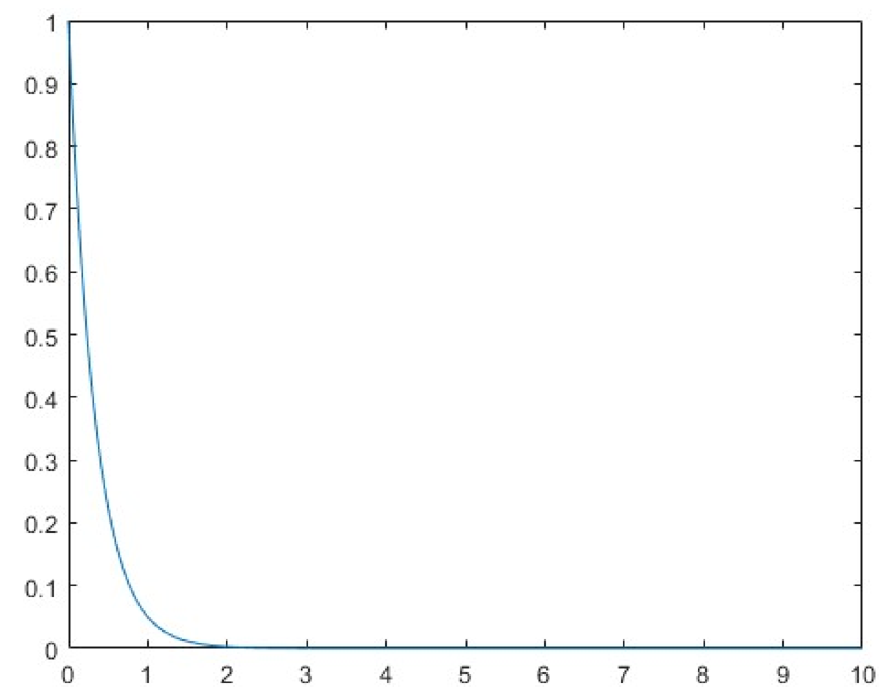
Part c:



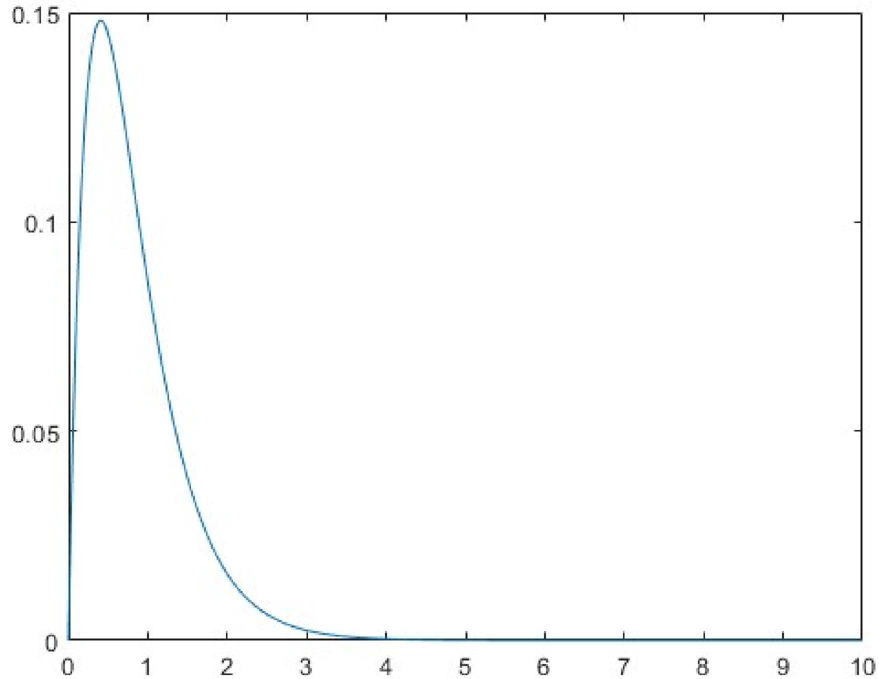
Part d:


0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 1 2 3 4 5 6 7 8 9 10

Part e:


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8 9 10

Part f:



Questions

b) Yes, my result of part (b) matches the plot of my analytical prediction in part (a)-(i), but the MATLAB plot is very jagged.
   Yes, after increasing the sampling size of vector t, the MATLAB plot looks the same as my analytical prediction.
c) The result in part (c) is the same as part (b), but its magnitude is multiplied by 5 and shifted to the right by 1.
d) Yes, part (d) has the same answer as part (b).
e) My result of part (e) matches the plot of my analytical prediction in part (a)-(ii); they are the same graph and they match.
f) My result of part (f) matches my analytical prediction in part (a)-(iii).

Discussion/Conclusion

This lab went smoothly and was completed with few issues. The issues stemmed mainly from figuring out how to create time delayed unit steps (from part c), which in of itself wasn't a very complex problem. Most of the lab was learning new functions and how their plots look like with our given inputs. The analytical prediction was straight-forward, now that I've become much better at convolution, and the comparisons were the easiest part of this lab. It was interesting to see the relationships between the "on-paper" plots and the MATLAB plots with each function. For instance, how the function lsim() can find $y(t)$ with just the roots of the differential equation, impulse() can find its impulse response $h(t)$, and exp() can find $y(t)$ from an exponential input. I believe my results to be accurate and I found no errors when writing my MATLAB script.

Appendix A

```
%% b
figure(1)
b = [1];
a = [1 3];
t = 0:0.01:10;
sys = tf(b,a);
x = ones(1,length(t));
y = lsim(sys,x,t);
plot(t,y);
%% c
figure(2)
x = zeros(1,length(t));
x(t>=1) = 5;
y = lsim(sys,x,t);
plot(t,y);
%% d
figure(3)
y = step(sys,t);
plot(t,y);
%% e
figure(4)
y = impulse(sys,t);
plot(t,y);
%% f
figure(5)
x = exp(-2*t) .* (ones(1, length(t)));
y = lsim(sys,x,t);
plot(t,y);
```