

ECE 351 - Lab 6 - Partial Fraction Expansion

Ethan Reeder

2 March 2021

Contents

1	Purpose	3
2	Procedure	3
3	Results	4
4	Questions	5
5	GitHub Link	5
6	Conclusion	5
7	Appendix	6

1 Purpose

The purpose of this lab was to use Laplace transformations to find the time-domain response of an RLC bandpass filter to both impulse and step inputs.

2 Procedure

This lab began with a prelab calculating the transfer function and the step response of a differential equation. These are shown in Equations 1 and 2 below:

Transfer Function:

$$H_t(s) = \frac{s^2 + 6s + 12}{s^2 + 10s + 24} \quad (1)$$

Step Response:

$$y(t) = \left[-\frac{1}{2} + e^{-6t} - \frac{1}{2}e^{-4t}\right] * u(t) \quad (2)$$

Then, during the lab time, the step response function was graphed with the step command from the scipy.signal library and compared, as can be seen in Figure 1 in the Results section. Next, the residue command from the scipy.signal library was used to find the R, P, and K values necessary for partial fraction expansion. The output in the console is shown in the Appendix.

Next, a more complex differential equation was given:

$$y^{(5)}(t) + 18y^{(4)}(t) + 218y^{(3)}(t) + 2036y^{(2)}(t) + 9085y^{(1)}(t) + 25250y(t) = 25250x(t) \quad (3)$$

This differential equation was then put into the residue function, and the results were then input into the cosine method function in order to find the step response. This derived step response was then plotted on the same plot as the step response attained from using the step command, as shown in Figure 2 in the Results section. The output from the console for the residue function is shown in the Appendix.

3 Results

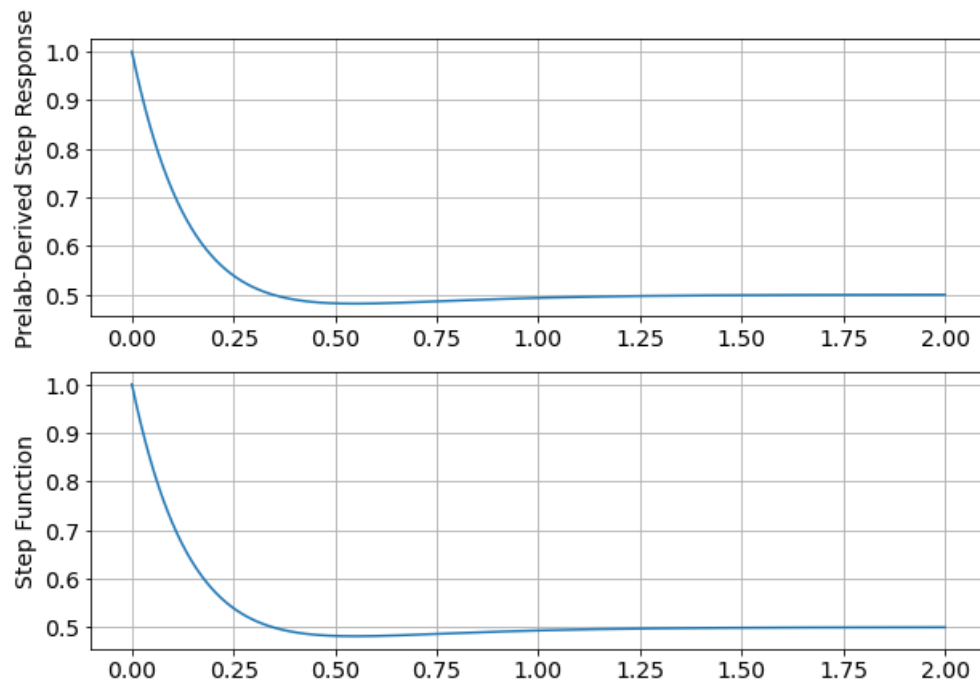


Figure 1: Part 1 Step Response

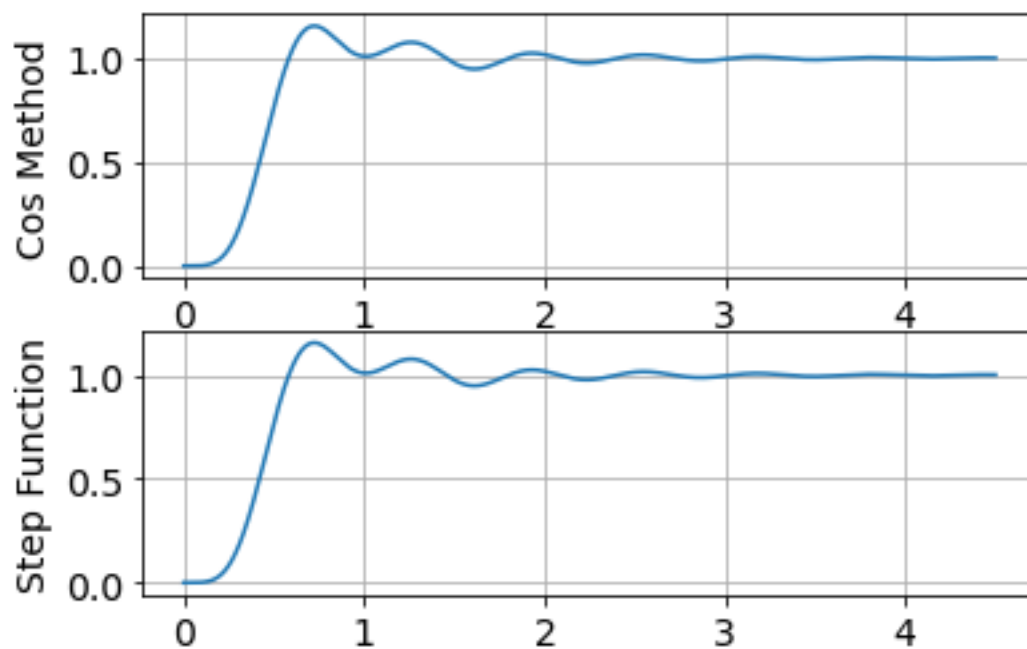


Figure 2: Part 2 Step Response

4 Questions

For a non-complex pole-residue term, you can still use the cosine method, explain why this works.

In the cosine method, the terms inside the cosine function are all due to imaginary parts, and thus they are 0 in a non-complex pole-residue term. Since $\cos(0) = 1$, the cosine method still is valid and works, though it might be a little more involved than doing simple partial fraction expansion if all the terms are real.

Leave any feedback on the clarity of lab tasks, expectations, and deliverables.

The lab was clear and the expectations were understood.

5 GitHub Link

<https://github.com/EReeder35>

6 Conclusion

In conclusion, this was another educational lab. In addition to being able to practice making my own function to implement the cosine method, I also enjoyed working with arrays to pass into functions with the R, P, and K values from the residue function.

7 Appendix

Console Output:

Part 1 R, P, **and** K Values:

R: [0.5 -0.5 1.]

P: [0. -4. -6.]

K: []

Part 2 R, P, **and** K Values:

R: [1. -7.20391843e-17j -0.48557692+7.28365385e-01j

-0.48557692-7.28365385e-01j -0.21461963+0.00000000e+00j

0.09288674-4.76519337e-02j 0.09288674+4.76519337e-02j]

P: [0. +0.j -3. +4.j -3. -4.j -10. +0.j -1.+10.j -1.-10.j]

K: []