

Assignment 6: The Laplace Transform

Aravint Annamalai, EE18B125

March 8, 2020

1 Abstract

In this assignment, the Laplace Transform approach to analyse Linear Time-Invariant Systems (often abbreviated as LTI Systems). LTI Systems are commonly used by electrical engineers as circuits, communication channels etc. can be modelled as LTI Systems. Python has a Signals toolbox which are a set of functions in the libraries `numpy` and `scipy.signal`. The various functions in this toolbox is used in appropriate locations in this assignment.

2 Introduction

If an LTI system has a **transfer function** $H(s)$, then the output $X(s)$, if the input is $F(s)$ is given by

$$X(s) = H(s)F(s)$$

Then, we use the `sp.impulse()` function in order to obtain the time domain output, and this can be plotted for some specific time interval of our choice.

The above algorithm is used for different scenarios in this assignment.

3 Task 1

Here, we are given a time-domain input

$$f(t) = \cos(1.5t)e^{-0.5t}u(t)$$

having Laplace transform as:

$$F(s) = \frac{s + 0.5}{(s + 0.5)^2 + 2.25}$$

Now, this input is applied to the spring which satisfies:

$$\frac{d^2x}{dt^2} + 2.25x = f(t)$$

In Laplace domain, the above equation becomes:

$$\begin{aligned}
s^2 X(s) + 2.25 X(s) &= F(s) \\
\implies X(s) &= \frac{F(s)}{s^2 + 2.25} \\
\implies X(s) &= \frac{s + 0.5}{((s + 0.5)^2 + 2.25)(s^2 + 2.25)}
\end{aligned}$$

This value of $X(s)$ is hard-coded into the program, with the use of the function `np.polymul()` in order to multiply the factors in the denominator and `sp.lti()` in order to assign the value of $X(s)$. Using `sp.impulse()`, the output in the time domain is got for 0-50sec, which is declared as a time vector using `np.linspace()`. The time-domain output is plotted.

4 Task 2

In this case, the input decays much slower as

$$\begin{aligned}
f(t) &= \cos(1.5t)e^{-0.05t}u(t) \\
\implies F(s) &= \frac{s + 0.05}{(s + 0.05)^2 + 2.25} \\
\implies X(s) &= \frac{s + 0.05}{((s + 0.05)^2 + 2.25)(s^2 + 2.25)}
\end{aligned}$$

A similar procedure as above is followed, with appropriate modifications due to the slow decay in the exponential of the input function.

5 Task 3

Here, we are going to vary the frequency of the cosine function in $f(t)$ from 1.4 to 1.6, in steps of 0.05. This is done in a *for* loop. The decaying exponential part is kept the same as Task 2 above (i.e) $\exp(-0.05t)$. Even here, the procedure is similar to the previous 2 scenarios, as the input function did not change much in form. Let the frequency of the cosine function be k , then the output function in the Laplace domain is as follows:

$$X(s) = \frac{s + 0.05}{((s + 0.05)^2 + k^2)(s^2 + 2.25)}$$

In this task, the value of k varies from 1.4 to 1.6 in steps of 0.05. `np.arange()` is used for the purpose. The time-domain output is plotted for each value of k , and the legend is given appropriately.

6 Task 4

In this task, we are solving a two-variable coupled differential equation in the Laplace domain. The differential equation in the time-domain is as follows:

$$\begin{aligned}\ddot{x} + (x - y) &= 0 \\ \ddot{y} + 2(y - x) &= 0\end{aligned}$$

with the initial conditions

$$x(0) = 1, \dot{x}(0) = \dot{y}(0) = y(0) = 0$$

In Laplace domain, the above set of equations become:

$$\begin{aligned}(s^2 + 2)Y(s) &= 2X(s) \\ (s^2 + 1)X(s) &= s + Y(s)\end{aligned}$$

On solving, the values of $X(s)$ and $Y(s)$ obtained are:

$$\begin{aligned}X(s) &= \frac{s^2 + 2}{s^3 + 3s} \\ Y(s) &= \frac{2}{s^3 + 3s}\end{aligned}$$

The above transfer functions are hard coded using `sp.lti()`. Then, they are converted into time domain with the time vector running from 0 to 20 secs using the function `sp.impulse()`. The two functions are plotted against time.

7 Task 5

In this task, the Bode plot of a transfer function is to be plotted. The circuit is an RLC circuit with the transfer function given by

$$H(s) = \frac{10^6}{10^{-6}s^2 + 100s + 10^6}$$

The manitude and phase vectors to be used for the Bode plot is obtained using the function `H.bode()`. In order to get the Bode plot familiar to us, we use the `semilogx` plot since the Bode plot is plotted with the frequency in the log scale. In this way, the Bode magnitude and the hase plots are plotted.

8 Task 6

Now, to the above network, we feed an input to it. The input in time domain is given by

$$v_i(t) = \cos(10^3 t)u(t) - \cos(10^6 t)u(t)$$

The transfer function, the time-domain input function and the time vector can be passed as arguments to the function `sp.lsim()` function and it gives the output function in time domain. (convolution of the input function and the transfer function in time domain. This can be plotted against time with the time vector of our choice. Here, since we need to observe the transient state as well as the steady state output, we plot the output for 0 to 30us and 0 to 10msec.

9 Conclusion

In this assignment, I had my first experience with the tools of `scipy` library and I have learnt to compute Laplace transform and related analysis of LTI systems in the Laplace domain using the simple tools available in the library. This technique is a really convenient one as sometimes, analytically calculating the time-domain output of a system might be non-trivial, whereas calculating it in the s-domain might be relatively much simpler. So, this assignment is a great learning experience for me and I would use the tools learnt here in the future as well.