Python 7 assignment
The Laplace Transform

March 2, 2020

In this assignment, we will look at how to analyse “Linear Time-invariant Systems” with numerical tools in Python. LTI systems are what Electrical Engineers spend most of their time thinking about - linear circuit analysis or communication channels for example. In this assignment we will use mostly mechanical examples, but will move on to circuits in the next assignment.

All the problems will be in “continuous time” and will use Laplace Transforms. Python has a Signals toolbox which is very useful and complete. In this assignment, we will explore the following commands:

Command	Description
poly1d	Defines polynomials >>> p = poly1d([1, 2, 3]) >>> print poly1d(p) $1x^2 + 2x + 3$
polyadd	>>> polyadd([1, 2], [9, 5, 4]) array([9, 6, 6])
polymul	>>> polymul([1,2],[9,5,4]) array([9, 23, 14, 8])
	Using poly1d objects: >>> p1 = np.poly1d([1, 2]) >>> p2 = np.poly1d([9, 5, 4]) >>> print p1 + 2 >>> print p2 29 x + 5 x + 4 >>> print p1 + p2 29 x + 6 x + 6
signal toolbox	import scipy.signal as sp
sp.lti	>>> s=sp.lti(Num,Den) defines a transfer function >>> H=sp.lti([1,2,1],[1,2,1,1]) >>> w,S,phi=H.bode() >>> subplot(2,1,1) >>> semilogx(w,S) >>> subplot(2,1,2) >>> semilogx(w,phi)
sp.impulse	>>> t,x=sp.impulse(H,None, linspace(0,10,101)) Computes the impulse response of the transfer function >>> plot(t,x)

Command	Description

sp.lsim	>>> t=linspace(0,10,101)
	>>> u=sin(t)
	>>> t,y,svec=sp.lsim(H,u,t)
	This simulates y=convolution of u and h

Next week we will explore sympy, which has powerful symbolic capabilities. For now we will use the simple tools in scipy. Unfortunately, scipy has poly1d but cannot define rational functions. That is why we define transfer functions using Num,Den rather than as a rational function of s.

1 The Assignment

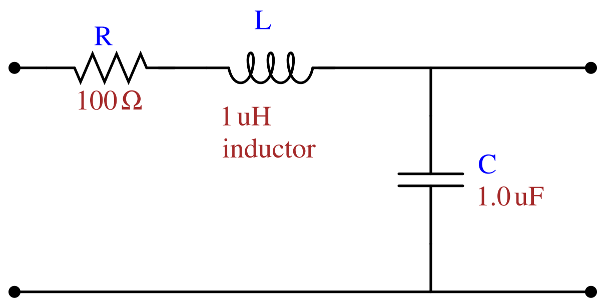
1. The Laplace transform of $f(t) = \cos(1.5t)\, e^{-0.5t} u_0(t)$ is given by
$$F(s) = \frac{s+0.5}{(s+0.5)^2+2.25}$$
Solve for the time response of a spring satisfying
$$\ddot{x} + 2.25x = f(t)$$
with $x(0) = 0$ and $\dot{x} = 0$ for t going from zero to 50 seconds. Use system.impulse to do the computation

2. Solve the above problem with a much smaller decay:
$$f(t) = \cos(1.5t)\, e^{-0.05t} u_0(t)$$

3. Consider the problem to be an LTI system. $f(t)$ is the input, and $x(t)$ is the output. Obtain the system transfer function $X(s)/F(s)$. Now use *signal.lsim* to simulate the problem. In a for loop, vary the frequency of the cosine in $f(t)$ from 1.4 to 1.6 in steps of 0.05 keeping the exponent as $\exp(-0.05t)$ and plot the resulting responses. Explain what is happening.

4. Solve for a coupled spring problem:
$$\begin{aligned} \ddot{x} + (x-y) &= 0 \\ \ddot{y} + 2(y-x) &= 0 \end{aligned}$$
where the initial condition is $x(0) = 1$, $\dot{x}(0) = y(0) = \dot{y}(0) = 0$. Substitute for y from the first equation into the second and get a fourth order equation. Solve for its time evolution, and from it obtain $x(t)$ and $y(t)$ for $0 \le t \le 20$.

5. Obtain the magnitude and phase response of the Steady State Transfer function of the following two-port network.



6. Consider the problem in Q5. suppose the input signal $v_i(t)$ is given by

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

Obtain the output voltage $v_0(t)$ by defining the transfer function as a system and obtaining the output using *signal.lsim*.

How do you explain the output signal for $0 < t < 30\mu s$? Can you explain the long term response on the msec timescale?

Note: You need to capture the fast variation, which means your time step should be smaller than 10^{-6}. Yet you need to see the slow time, which means you must simulate till about 10 msec. Use an appropriate time vector.

Note: Since the network is resistive, the current at $t = 0^-$ will have decayed to zero, which gives you your initial condition.