Lab 2 ELEC4400: Z Transform and Convolution

Thomas Fuller

May 21, 2019

Part 1: Fibonacci

Last time we used Z Transform Techniques to determine a closed form expression for the Fibonacci Sequence, shown in equation form below.

$$y[n] = \frac{1}{\sqrt{5}}\left([\frac{1+\sqrt{5}}{2}]^n - [\frac{1-\sqrt{5}}{2}]^n\right)$$

This was derived via Z transform from the difference equation

$$y[n+2] = y[n+1] + y[n]$$

Problem 1

Verify that both of these equations output Fibonacci Numbers, then use the python `time`" built in library to time how long it takes to solve for the n-th Fibonacci Number using both versions of the Fibonacci sequence.

Please Generate a plot with n on the horizontal axis, and time on the vertical axis, comparing the speed of each method for computing the first 30 fibonacci numbers

Some Helpful Things

To write to a file

```
f = open('data.csv','w')

f = write('%s,%s,%s'%(t,x1,x2))
```

texttf.close()

Convolution

Today's event is all about convolution in discrete time and some of its applications. We said that $x[n]$ was a vector of input values, and that $y[n]$ was a vector of output values. $h[n]$ is a function associated with a system which is the "unit sample response of the system". Another way of phrasing this is to say that

$h[n]$ is a function that describes what the system WILL DO to a unit sample in $x[n]$. Here is this big equation defining the convolution.

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

What I want to tell you, and what I hope to show you by the end of this activity is that convolution is multiplying **each** element in the input $x[n]$ by **every** element in a time shifted version of $h[n]$, and then adding all the results together.

The property of time invariance allows us to use shifted versions of $h[n]$, and the property of linearity allows us to sum the results.

This idea is important because it allows us to determine the output response $y[n]$ of any system (described by $h[n]$) to any input $x[n]$, so long as the system obeys the property of linearity and time invariance.

You've done Convolution a 10^6 times already

We've all done convolution from a young age, and we did it when we performed the operation of multiplying two polynomials together. Consider two polynomials which are dear to my own heart,

$$x = s^2 + 4s + 3 \tag{1}$$

and also

$$h = s + 5 \tag{2}$$

Problem 1

By the traditional FOIL method, carefully multiply these two polynomials together.

Convolution for Polynomial

Multiplying polynomials can be thought of as a convolution of the vectors containing the coefficients of each polynomial. So say I want to FOIL (or from now on let's say "convolve" or "convolute") the polynomials $q = 2s + 4$ and also $v = 5s + 3$ together. I can list their coefficients into a vector in the following way

$$q = \begin{bmatrix} 2 & 4 \end{bmatrix} \tag{3}$$

$$v = \begin{bmatrix} 5 & 3 \end{bmatrix} \tag{4}$$

Problem 2

Multiply the polynomials q and v by the convolution method.

Problem 3

Multiply my favorite two polynomials (x and h from problem 1) via the convolution method.

Part 3

Math describes parts of our lives in a rigorous way, and convolution is the math that describes what the output of a digital signal is when it passes through a system. I want to try and show you that that's the truth now. Consider the RC circuit below, with $R = 1k\Omega$, and $C = 1nF$. There are various ways of figuring

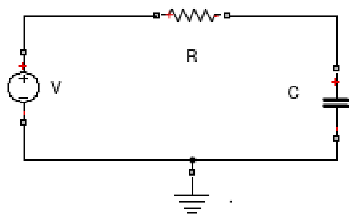

Figure 1: Figure of Part 3

out what the impulse response of this circuit is, but I'm here to just tell you that its

$$h(t) = 10^6 e^{-10^6 t} \tag{5}$$

Problem 4

I'd like to ask you to please find the response of this circuit to a constant input (the step response) using the `np.conv()`. Set $t = nT$ with $T = 10^{-8}$, and use an input vector that is 1000 samples long, and plot the first 10 microseconds of the response. One small subtlety of this problem; since we're using discrete math to approximate the continuous time response of the function, instead of using $u[k]$, we'll use $Tu[k]$ where T is the width between samples.

Hand in a printed copy of your answers to each problem, as well as the code used to generate the final result.