The Mathematics of Signal Processing - an Innovative Approach

Peter F. Driessen
Department of Electrical and Computer Engineering
University of Victoria, Canada V3W 3P6
driessen@uvic.ca

ABSTRACT— *The traditional EE math courses in complex variables and z-transforms are taught here in the context of digital filter design. The course material is integrated via a project which requires the design of a simple digital filter, and a complete mathematical analysis and computer simulation of the filter characteristics. The design is facilitated with software that displays the filter impulse and frequency response as the poles and zeros of the transfer function are placed on and moved around the z-plane. The mathematical analysis includes finding the inverse z-transform to obtain the filter impulse response using 5 different mathematical methods, thus covering in one place virtually all of the complex variable and z-transform math learned in the course. The filter design context and the self-checking nature of this project has made it popular with students. A similar approach may also be used with Laplace transforms in the context of analog filter design.*

I. Introduction

The traditional EE math courses in complex variables and z-transforms may seem irrelevant to many students. The traditional complex variables course curriculum includes properties of functions of a complex variable z, complex contour integrals, convergence of sequences and series, power series expansions, and may include properties and calculation of z-transforms. All of this mathematics has its major EE applications in digital filters, digital signal processing and digital control systems. However, these courses typically come later in the program, so the students do not see the relevance of the math at the time they are taking the math course.

These traditional math courses can be made much more interesting and relevant for students by applying this mathematics to digital filter design. In this paper, we present a new approach to the traditional complex variables and z transforms course curriculum. This new approach unifies all of the material under the banner of a simple digital filter design and a complete analysis of it's properties. Similarly, this approach may be used with analog filter design, analog control systems and Laplace transforms.

The paper is organized as follows. In Section 2, the traditional course curriculum is reviewed, and the connections to digital filter design are pointed out. In Section 3, the new course curriculum is described.

Thanks to all the students in my courses who put up with my approach and seemed to enjoy it.

II. Traditional course curriculum and connections to digital filters

A standard course in complex variables, e.g. [1], includes properties of functions of a complex variable z, complex line and contour integrals, convergence of sequences and series, power series expansions and residue theory. A standard course in signals and systems e.g. [2], includes z-transform definitions and properties, methods for taking inverse transforms using long division, partial fractions and tables, and methods for solving difference equations via transform methods.

It is typically not emphasized that the inverse z-transforms can also be taken using contour integrals

$$y[k] = \frac{1}{2\pi j} \oint_c Y(z) z^{k-1} dz \qquad (1)$$

This integral can be evaluated in two ways: using integration along a path which encircles all of the poles of $Y(z)z^{k-1}$, as well as using residue theory to obtain

$$y[k] = \sum Res[Y(z) z^{k-1}] \qquad (2)$$

The inverse z-transforms can also be found using power series expansions in negative powers of z about $z = 0$ (Laurent series)

$$Y(z) = \sum_k y[k] z^{-k} \qquad (3)$$

with a defined radius of convergence (ROC) equal to the magnitude of the pole of $Y(z)$ with the largest absolute value. This ROC can also be found using the ratio or root tests used to test convergence of series.

These 3 methods of taking inverse z-transforms incorporate most of the material of the standard course in complex variables.

If all 6 methods of inverse z-transforms are applied to the same function, then in general, the algebraic expression for the sampled time domain signal $y[k]$ as a function of k will be different. However, the actual numerical values of $y[k]$ for each k will of course be identical.

Digital filter design normally involves selecting the pole and zero locations to obtain a desired transfer function $H(z)$. The filter impulse response (or response to any other input) is found by taking an inverse z-transform. Thus there is a strong connection between a course in complex variables and practical applications. The connection is establisehd by using complex variable theory to take the inverse z-transform to get a practical result (the numerical values of the discrete-time system response) which can be plotted and observed.

III. New course curriculum

The course begins with an introduction to digital audio (CD, DVD, MP3, MP4) and digital video (DVD) as a means to motivate the study of sampled (discrete-time) systems. Other applications of digital signal processing such as digital control systems and systems for manipulation and enhancement of digital (still) images are presented also. We emphasize the key idea that in all DSP, analog input signals (audio, video, control) are sampled and quantized by an A/D to produce numbers, and these numbers are manipulated to yield different numbers which may be input to a control program and/or go to a D/A for analog output.

We then introduce discrete-time systems, linearity and time-invariance, difference equations FIR/IIR and convolution. z-transforms and the transfer function $H(z)$ are introduced as a means to solving difference equations with arbitrary input $X(z)$. At this point, we have motivated the need for taking the inverse z-transform of the output $Y(z) = H(z)X(z)$. We then introduce the (co mplex) inversion integral which defines the inverse z-transform, and are thus confronted with the need to learn about complex contour integrals and integration along a path. We mention how the contour integral which yields $y[k]$ can be simplified using partial fractions, so that it reduces to the sum of integrals of the form

$$\oint_c \frac{Cz}{(z-p)^m} \tag{4}$$

for an mth order pole, where there will be a different set of integrals for each value of k. Next we introduce techniques to simplify the calculation of inverse z-transforms, first using residue theory, and then using power series expansions (Laurent series) thus motivating the requisite knowledge of sequences, series and their convergence properties. In this way, the complex variable theory is put in the context of predicting the outputs of digital filters using inverse z-transforms.

IV. Software

Software is used to aid the visualization of filter impulse and frequency responses as a function of pole and zero locations. There are several versions of such software available, including one called POI [3] developed as a stand-alone package as a student project. POI allows poles and zeros to be dragged and dropped on the z-plane, and displays expressions for the difference equations and transfer function as well as impulse and frequency responses. In addition, Matlab is used to implement the filters. Both POI and Matlab are used to filter real data (speech, music and image) to show the effect of various filter types such as lowpass, highpass, bandpass and bandstop. Block diagram software such as MAX/MSP [5] for music signal processing,and others [6] are mentioned.

V. Term project - Design of digital filters

The contents of the traditional course curriculum in complex variables and z-transforms can be retained in it's entirety and unified via a course project as follows. A key attribute of this project is that it is self-checking: the 8 separate and independent calculations of the filter impulse response must all yield the same result.

Everything about a 2-pole 2-zero bandpass filter.

Design a digital bandpass filter, test the filter characteristics, and document your results in a succinct but complete report. The specifications for the bandpass filter are

- - passband center frequency $f_0 = 1{,}000$ Hz.
- - sampling rate $f_s = 8,000$ Hz
- - 3dB bandwidth $B = 500$ Hz
- - using 2 poles and 2 zeros.

Use any general purpose software such as C, Fortran, Matlab, Maple, and spreadsheets. Title all graphs, label the axes, and include a scale with proper units.

1. Filter design

(a) Design the filter and specify the transfer function H(z). Scale the filter for unity gain at f_1.

(b) An approximate formula for the 3 dB bandwidth B of the filter as a function of the magnitude $|a|$ of the poles (distance of pole from origin) is $\delta\omega = 2|1-a|/\sqrt{a}$, where $\delta\omega$ is the normalized bandwidth B in radians. This shows that the distance of the pole from the unit circle $|1-a|$ controls the bandwidth. Find a numerical value for $|a|$ such that $B = 500$ Hz.

(c) Plot the frequency response (both amplitude and phase) of H(z). Check that the power response is down 3 dB at $f_1 + B/2$, as it should be with the correct value of $|a|$. What is the phase shift at f_1 and at $f_1 + B/2$? Write your own program to make this plot, and compare the results with POI.

(d) Find the difference equations by analysis and compare with POI.

2. Find the impulse response by computer.

(a) Take the IDFT (Inverse Discrete Fourier Transform) of the sampled frequency response to obtain the impulse response $h(n)$. Do this manually with your own program, and with a Matlab, spreadsheet or other standard FFT routine. Use $N = 1024$. Find the frequency resolution.

(b) Find the impulse response $h(n)$ using the difference equations. Obtain numerical results for $h(n)$ for $0 < n < 25$. Spreadsheets are recommended for programming the difference equations, but other types of computer simulation (e.g. C, Fortran) are acceptable. Repeat using MATLAB. For what value of n is $h(n)$ less than one percent of it's maximum value?

3. Find the impulse response $h(n)$ by analysis.

(a) Find the impulse response $h(n)$ by taking the inverse z-transform of $H(z)$ using the inversion integral. Do the integral by two different methods, contour integration along a path, and residues. Obtain results for $0 < n < 3$ in each case.

(b) Find the impulse response $h(n)$ using a Laurent series expansion of $H(z)$ for $0 < n < 3$. Find the radius of convergence of the Laurent series using both the ratio test and the root test.

(c) Take the inverse z-transform using 3 different methods long division, partial fractions with first order factors, partial fractions with quadratic factors. Obtain numerical results for $0 < n < 3$ in each case.

4. Prepare a table with 9 columns, listing n and $h(n)$ for the difference equation calculation, the IDFT calculation, and the 6 analytical methods, for $0 < n < 3$.

5. Demonstrate that the filter works correctly by computer simulation as follows:

(a) Evaluate the filter output $y(n)$ with sinusoidal input $x(n)$ by using the difference equations. Use sampled sine waves at the center frequency f_1, repeat again at the 3 dB down frequency $f_1 + B/2$. Verify that both the amplitude and phase of the $y(n)$ sine wave output are correct relative to the input $x(n)$, by plotting the input and output on the same graph, and measuring the amplitude and phase shift. If the filter is initialed with all zeros, then how long is the transient response before the amplitude and phase reach a steady state?

Note that the resulting points will not look much like a sine wave because there are only about 4 samples per cycle. You can (optionally) use an interpolation routine (a plotting option) to get a smooth sine wave curve through the sample points.

(b) Repeat by computing the convolution of the input $x(n)$ with $h(n)$, and compare the results with the difference equation method. How many terms of $h(n)$ are needed to get reasonable agreement?

(c) Now consider the input $x(n)$ to be three sine waves at $f_1/2, f_1, 3f_1/2$. The bandpass filter output $y(n)$ should be a sine wave at f_1, i.e. the other frequencies are mostly filtered out. Verify this and plot the input and output on the same graph. Take the DFT of $x(n)$ and $y(n)$, and verify that $Y(f) = X(f)H(f)$ at selected sample points in the frequency domain.

(d) Now consider the input $x(n)$ to be $f_s - f_1 = 7,000$ Hz. Compute the filter output $y(n)$ by convolution, and explain the result.

(e) Use the filter to process the a voice audio file and describe the effect of the filter (i.e. how is the output sound different from the input). Is the voice still intelligible after filtering ?

6. Take the DFT of the impulse response to find the frequency response $H(f)$. Choose the DFT size N to obtain a frequency resolution of 5 Hz or better. Use zero-padding if needed. Repeat using a smaller DFT size for which the frequency resolution is about 16 Hz. Explain why $H(f)$ is different for different values of N. Which $H(f)$ is correct, if any ?

7. Consider the filter difference equation with initial conditions $y(-1) = 0, y(-2) = 1$, and input $x(n) = u(n)$. Find the filter output $y(n)$ by z-transform analysis, and confirm the result by computer simulation.

8. Consider an adaptive (time-varying) version of the bandpass filter where the center frequency f_1 changes in response to a control signal. Modify the MATLAB filter implementation to achieve this.

The last item of the project tests understanding of the relationship between the filter coefficients a_k, b_k in the difference equation

$$y[k] = a_1 y[k-1] + a_2 y[k-2] + b_0 x[k] + b_1 x[k-1] + b_2 x[k-2] \quad (5)$$

and the the pole-zero locations p_k, z_k in the transfer function

$$H(z) = \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} \quad (6)$$

where $p_2 = p_1^*, a_1 = p_1 + p_2, a_2 = -p_1 p_2, b_1 = -b_0(z_1 + z_2), b_2 = b_0 z_1 z_2$. The adaptive filter illustrates how to control a_1, a_2, b_1, b_2 so that f_1 (i.e. the phase of p_1) changes in proportion to the control signal.

VI. Summary

We have shown how the traditional course content in complex variables and z-transforms is integrated via a project which requires the design of simple digital filter. The impulse response of this filter is found using a total of 8 different methods, both analysis and computer calculation, thus unifying all of the course material in a design context. The course project has proven to be very popular with students, over the last 5 years. Each year, several students come to visit the author to say that they learned the whole course by doing the project.

References

[1] E. Kreysig, *Advanced Engineering Mathematics*, Wiley, 1998.
[2] B.P. Lathi, *Linear Signals and Systems*, Cambridge, 1994.
[3] http://members.nbci.com/tfletche/poi/
[4] http://www.ece.uvic.ca/ peter/31001/310index.html
[5] http://www.cycling74.com
[6] http://www.cera2.com/WebID/dsp/filters/blank/freeware/a-z.htm