ELEC 3004/7312: Digital Linear Systems: Signals and Control!

Tutorial 3 [Week 6] - z-Transform, FIR vs IIR, FIR filter design and Q4

z-Transform

Using the table from the end of this sheet, we will work through one example, ask you to solve one yourself and go over the solution. Then we will discuss the concept of region of convergence, and whether or not the transform is unique.

- Examples
  - Ex 1:
  $$x[n] = 0.2^n \cdot u[n] + 4\delta[n-4]$$
  - Ex 2:
  $$x[n] = 0.2^n \cdot u[n] \cdot 4\delta[n-4]$$
- Region of Convergence?
- IS the Z-transform Unique?
- Please see also p. 674 of Lathi for a z-Transform Table

FIR vs IIR

What is the form of an FIR filter?

What is the form of an IIR filter, and why does this result in an infinte impulse response?

Considering these forms, what is the stability of the filters for both cases,IE given a bounded input, can we guarantee one filter topology will have a bounded output?

How does Rads/Sample relate to sampling rate and cut-off frequency?

FIR filter design

This part will cover the design of an FIR filter. in tutorial one we asked you to use matlab to produce a low pass filter. that filter had no specifications. here we will introduce specifications to the design process, and a tool which will make this much easier.
open SPTOOL in matlab, and use it to design a notch filter.
assuming a sampling rate of 44khz, design a notch filter for 50hz with the low pass cut off set to 40hz and the high pass cutoff set to 70hz. Maximum passband ripple is 3db at both sides.[1]

[1]This is a typical mains filter design for an audio setup, IE the 'easy' approach to removing the 50hz hum from Eleda's tune in problem set 1 Q2

PS2 Q4

This is going to be a discussion on the three points from part A. While the discussion won't directly answer the problem, the answers to the questions below can be used to formulate an argument for why they break the perfect inverse claim.

- What is Quantisation?
- can we correct Quantisation Error?
- What happens to a signal when we try to use sinc reconstruction on a signal of finite length?
- Why do we get clock jitter?
- What does this mean for a sampling system?

Extension - 2D and Image processing

Included is a challenge question in the nature of material in Problem Set 2. It was decided that it was out of scope for the course, but as I have already developed a model solution, If time permits we will solve part a in the tutorial, and the solution to parts a and b will be released on the last day of class before Easter. The solutions to part b is very similar to part a, part c is a bit different, but not much more code than the other two.

Jo Ogle Picasa is an engineer looking to dabble in the fine arts. As a busy engineer with a lot on her plate, she has contracted you to implement her vision using some signal processing techniques. She has provided you with a famous painting, and tasked you with implementing a number of 2D filters and algorithms to "enhance" the work:

a) Form a 2D kernel as follows:

$$Sobel_x = \begin{bmatrix} 1 & 0 & -1 \\ -2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$Sobel = Sobel_x * Sobel_y = \begin{bmatrix} 0.25 & 0.5 & 0 & -0.5 & -0.25 \\ 0.5 & 1 & 0 & -1 & -0.5 \\ 0 & 0 & -0.25 & 0 & 0.25 \\ -0.5 & -1 & -0.5 & 1 & 1 \\ -0.25 & -0.5 & -0.25 & 0.5 & 0.5 \end{bmatrix}$$

note that the convolution here is what matlab refers to as the 'full' convolution. Apply the 5x5 Sobel operator to the image by convolving the image with the operator. conv2 may be useful.

b) Generate a 9x9 approximation of a 2D Gaussian use the following formula to generate the weights

$$Gauss(x, y) = \exp \frac{\sqrt{(5-x)^2 + (5-y)^2}^2}{a}$$

$$Gauss_{norm} = \frac{Gauss}{\sum_x \sum_y Gauss(x, y)}$$

The second line normalises the convolution kernel so the sum of all the components is one and therefore the kernel will not introduce a gain.
use the values a = 2 and a = 0.2. Again conv2 may come in handy.

c) Take the direct cosine transform of the image, and set to zero 95 percent of the least significant terms (based on the absolute value of the DCT weighting)[2]. dct2 and idct2 will come in handy (these are closely related to the fft function, computing the 2 dimensional Discrete Cosine Transform (and inverse DCT) of the image rather than the Discrete Fourier Transform)

Download the supplementary materials folder. inside you will find a small piece of code which will load the images, and print them in Matlab.

[2]this is analogous to what JPEG does to compress an image. For those in CSSE3010 you will cover Huffman coding soon, and it will hopefully become apparent that the new image is a prime candidate for huffman coding, as there are so many repeated symbols to exploit.