Electrical & Electronic Engineering, NUI, Galway

EE445 DIGITAL SIGNAL PROCESSING

Tutorial DSP-0

DISCRETE-TIME SIGNALS AND SYSTEMS

Objective
The objective of this tutorial is to use Octave to investigate the generation of some common discrete-time signals, and additionally to examine the time-domain behaviour of discrete-time systems, in particular the impulse responses of such systems.

Completion of this "tutorial" is optional (there is no submission required, and no credit associated with it), and it is mainly intended to provide some additional familiarity with Octave in the context of signal processing.

PROGRAMMING TIPS

1. ***You should always make use of appropriate comments in your code, for clarity.***
2. ***Axes on plots should be labeled appropriately (including units, if applicable). Also, make sure that you include a caption with each figure. Figures without these elements are meaningless!***
3. ***Remember that Octave is simply a software tool, and is just a means to an end – as with any programming environment, it's easy to make logical (not just syntax) errors that will give you the wrong result. So, you should always try to have some idea of what the answer should be, so that you can identify the presence of logical errors in the code that you've written.***

Background Material
DSP Notes, Sections 1 and 2

Examples
Before attempting the exercises below, you should refer to the examples covered in the lectures, the code for which is located on the EE445 subject page. Also, remember that to obtain help on any Octave function or code construct, you can use the online Help menu in the Octave application itself. Alternatively, you can simply type "help <function name>" at the Octave prompt ">>".

Exercises

1. Write Octave code to generate and plot against time (for a period corresponding to 50 msec) an AM voltage waveform with the following parameters:

   Carrier frequency = 2 kHz
   Modulating signal = a sine wave of frequency 100 Hz
   Carrier amplitude = 8 V
   Modulation index = 0.75.
   Sampling frequency = 10 kHz

   (Note: you should revisit your EE308 Signals and Communications notes for details on modulated waveforms).

2. Two first-order digital filters are described by the following transfer functions:

$$H_1(z) = 1 - 0.9z^{-1}$$
$$H_2(z) = \frac{1}{1 - 0.9z^{-1}}$$

   Write Octave code to compute and plot the impulse responses, $h_1(n)$ and $h_2(n)$ of the two filters, for $0<=n<100$. You should demonstrate the calculation of the impulse response using both the Octave *impz* function, as well as an explicit implementation of the difference equation (e.g. using a *for* loop).

3. For the following difference equation:

$$y(n) - 0.9y(n-2) = 0.3x(n) + 0.6x(n-1) + 0.3x(n-2)$$

calculate and plot the following using Octave:
   (i) The impulse response $h(n)$ for -20<=n<=100
   (ii) The unit step response for -20<=n<=100

4. A second-order system is described by the following transfer function:

$$H(z) = \frac{1 + b_1 + b_2}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

The poles of this filter can be represented in polar form with a frequency of θ_o and a radius of r in the z-plane. Furthermore, the coefficients are related to the poles by the following equations:

- $b_1 = -2r\cos(\theta_o)$
- $b_2 = r^2$

Calculate the filter coefficients for an "analogue" frequency equal to $1/10$ of the sampling frequency (remember that you need to convert this to digital frequency – use the fact that the sampling frequency corresponds to 2π in digital frequency), and $r = 0.85$, and plot the impulse response for 0<=n<100. Also, use the `zplane` function to plot the pole-zero map of the filter, and verify that the poles are where you expect them to be.