EECS 3602 Lab 3 : Discrete Time Systems

Submission details: Write your responses to the following questions and submit them electronically as a lab report, along with any code that you write. If your responses are handwritten, scan them for electronic submission. Submission is via Moodle. Due date: October 29, 2015.

Grading details: 70% of your lab grade is for correctly completing the lab requirements; 20% is for clear writing and good presentation, including readable and well-documented code; 10% is for extra work or analysis that expands on or goes beyond the lab requirements.

Part 1: Discrete-time convolution. MATLAB has a `conv()` command to implement discrete-time convolution. In this part of the lab, you will implement your own convolution command.

Let $x[k]$ and $h[k]$ represent signals, where $x[k]$ is supported[1] on $k \in \{0, 1, \ldots, \ell_x - 1\}$, and where $h[k]$ is supported on $k \in \{0, 1, \ldots, \ell_h - 1\}$. The convolution $x[k] \star h[k]$ is given by

$$y[k] = x[k] \star h[k] = \sum_{\tau=-\infty}^{\infty} x[\tau] h[k - \tau].$$

Let `x` and `h` represent MATLAB vectors which contain the ℓ_x elements of $x[k]$ and the ℓ_h elements of $h[k]$, respectively.

Write a command `myconv()` that:

- Takes as arguments twi vectors, `x` and `h`, representing discrete-time signals $x[k]$ and $h[k]$;
- Returns a vector `y` of length $\ell_x + \ell_h - 1$, representing the convolution of `x` and `h`.

When writing `myconv()`, use operations that are as basic as possible: do not use `conv()` or other toolbox functions. The idea is for you to implement this function by yourself from scratch. (Using vector slices, e.g. `x[2:4]`, is ok.)

[1]The *support* of a function is the range over which it is nonzero.

For the report: Discuss how you wrote `myconv()`. Compare the output of `myconv()` to the output of `conv()` for the same inputs.

Part 2 : Discrete-time systems. Consider a system with input $x[k]$ and output $y[k]$, related by the difference equation

$$y[k] + \sum_{i=1}^{n_b} b[i]y[k-i] = \sum_{j=0}^{n_c} c[i]x[k-i].$$

Note that $b[i]$ is defined for $i \in \{1, 2, \ldots, n_b\}$, while $c[i]$ is defined for $i \in \{0, 1, \ldots, n_c\}$. (It may help to think that $b[0] = 1$). Let `x`, `b`, and `c` represent MATLAB vectors contaning $x[k]$, $b[i]$, and $c[i]$, respectively.

Implement a function `mysystem()` which takes `x`, `b`, and `c` as arguments, and returns the system output `y`. As the system output may be infinitely long, `y` should be long enough to capture "most" of the output. (Use your judgment.)

For the report:

- Discuss how you wrote `mysystem()`.
- How would you get the impulse response of a given system? Discuss and give a few examples via MATLAB plots.
- Compare the output of `mysystem()` and `myconv()` for both finite impulse response (FIR) and infinite impulse response (IIR) systems.
- How would you obtain the (Fourier) transfer function of a given system? Discuss and give some examples via MATLAB plots, including the response to sinusoidal signals of various sizes.