Digital Signal Processing Laboratory

Laboratory 3 Discrete-Time Systems

3.1 Introduction

The purpose of this lab[1,2] is to explore the characteristics of discrete-time systems.

3.2 Discrete-time Systems

The following continuous-time systems are commonly used in electrical systems:

$$\text{differentiator:} \qquad y(t) = \frac{d}{dt}x(t) \tag{3.1}$$

$$\text{integrator:} \qquad y(t) = \int_{-\infty}^{t} x(\tau)d\tau \tag{3.2}$$

3.2.1 EXERCISE

1. For each of the two systems described in 3.2:

   a) Formulate a discrete-time system that approximates the continuous-time functions. Write the difference equation that describes the discrete-time systems.

   b) Draw the block diagram of the discrete-time systems.

2. Write two functions that will apply the differentiator (function ~3212d.py) and integrator (function ~3212i.py) systems. Apply the differentiator and integrator to the following two signals for $-10 \le n \le 20$.

   a) $x_a(n) = \delta(n) - \delta(n-7)$ (Figures ~3212ad.png, and ~3212ai.png)

   b) $x_b(n) = u(n) - u(n-(N+1))$ with $N = 8$ (Figures ~3212bd.png, and ~3212bi.png)

   To compute $u(n)$ for $-10 \le n \le 20$, set `n = -10, -9, .., 20`, and use the boolean expression `u = (n>=0)`. Use the `subplot` and `stem` commands to co-plot the input and output signals.

[1] C. Bouman, *Digital Signal Processing with Applications*, School of Electrical and Computer Engineering, Purdue University

[2] S. Burrus, etal., *Computer-Based Exercises for Signal Processing using Matlab*, Prentice-Hall: Englewood Cliffs, NJ, 1994, pp.8-10

Hint: When implementing a difference equation using `for` loops, pre-define the output vector before entering the loop. If you are using a `for` loop to filter the signal $x(n)$ and yield an output $y(n)$, place the following command before the `for` loop

```
y = zeros((1,N),float)
```

where N is the length that y should be after filtering.

3.3 Difference Equations

3.3.1 EXERCISE

1. Write Python functions to implement two discrete-time filters S_1 and S_2 described by the following difference equations:
   a) $y_1(n) = \frac{1}{2}x(n) - \frac{1}{2}x(n-1)$ (Function `s1(..)`)
   b) $y_2(n) = \frac{1}{2}y(n-1) + x(n)$ (Function `s2(..)`)

   Place the two functions in a module named ∼3.py.

2. Use the functions to plot the impulse response of the following systems:
   a) S_1 (Figure ∼3312a.png)
   b) S_2 (Figure ∼3312b.png)
   c) $S_2(S_1)$ the series connection with S_1 followed by S_2 (Figure ∼3312c.png)
   d) $S_1(S_2)$ (Figure ∼3312d.png)

3.4 Audio Filtering

3.4.1 EXERCISE

Listen to the file `music.wav` using a sound player. Load this file into Python using `scipy.io`.

Filter the audio signal with each of the two systems S_1 and S_2 in section 3.3.1. How do the filters change the sound of the audio signal? Explain your observation (File ∼341.txt).

3.5 Infinite Impulse Response Difference Equations

An IIR(infinite impulse response) filter is an LTI system expressed as a linear constant-coefficient difference equation:

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k) \tag{3.3}$$

In `scipy.signal`, difference equations are represented by two vectors: one vector containing the feedforward coefficient, b_k for the x terms, and the other vector containing the feedback coefficients, a_k for the y terms. The coefficient a_0 is usually taken to be 1, so that when $y(n)$ is written in terms of past values it drops out:

$$y(n) = -\frac{1}{a_0}\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

In `scipy.signal` the `lfilter` function will divide out a_0 so it must not be zero.

Hints: The function `y = lfilter(b,a,x)` implements a digital filter defines by the `a` and `b` coefficients as in (3.3) to filter the data stored in x. If x is the unit impulse signal, then y will be the impulse response $h(n)$. Note that the function `lfilter` returns only as many samples into y as there are in x (the impulse response is truncated to the length of the unit impulse vector, x).

3.5.1 EXERCISE: Simple Difference Equation

1. Create vectors b and a that contain the coefficients of $x[n]$ and $y[n]$, respectively, in the following difference equation:

$$y(n) + 0.8y(n-2) = 0.2x(n) + 0.4x(n-1) + 0.2x(n-2) \tag{3.4}$$

2. Calculate $y(n)$ analytically for $x(n) = \delta(n)$

3. Create a unit impulse vector, `imp`, of length 128. Generate the first 128 points of the impulse response of the filter in (3.4). Use `stem` to plot these values as a discrete-time signal versus time. Plot the first 20 points (Figure ∼351.png).

3.5.2 EXERCISE: Impulse Response with filter

1. Use the `lfilter` function to generate and plot the impulse response $h(n)$ of the following difference equation. Plot $h(n)$ in the range $-10 \le n \le 100$ (Figure ∼352.png).

$$y(n) - 1.8\cos(\tfrac{\pi}{16})y(n-1) + 0.81y(n-2) = x(n) + \tfrac{1}{2}x(n-1) \tag{3.5}$$

2. Determine the impulse response analytically and confirm your results.

I will highly appreciate your comments and suggestions to improve this material.
R. Stephen L. Ruiz rslruiz@gmail.com