EECS351 Discussion 1 with MATLAB demo, 09/08/16

Nate Sawicki

1 What is a digital signal? Discrete Time vs. Continuous Time

A digital signal is simply a sequence of discrete values. A digital signal x[n] is only valid for integer n. A continuous signal x(t) is valid for all t.

Digital signals can be denoted using brackets, braces, or parenthesis.

Example: x[n] =[$\underline{1}$, 2, 3, 3, 5, 1]

NOTE: We underline a value to denote the 0th index of a signal (n = 0)

2 Kronecker Delta (a.k.a. Delta Sequence, Unit Sample Sequence)

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

2.1 Decomposition of Discrete Signals with Shifted Deltas

Any sequence $x[n]$ can be written as $x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$

2.2 Dirac Delta (a.k.a. Unit Impulse Function)

A loose definition:

$$\delta(t) = \begin{cases} +\infty, & t = 0 \\ 0, & t \neq 0 \end{cases} \quad \text{s.t. } \int_{-\infty}^{\infty} \delta(t)dt = 1.$$

A more precise definition:

$\delta(t) = \lim_{\Delta \to 0} \frac{1}{\Delta} rect(\frac{t}{\Delta})$

2.3 Decomposition of Continuous Signals with Shifted Deltas

Any continuous-domain function $x(t)$ can be written as $x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)d\tau$

2.4 Kronecker vs. Dirac

What's the deal with these different delta functions?

The Kronecker delta, denoted $\delta[n]$, is a sequence indexed over a discrete domain, in other words, $n \in \mathbb{Z}$. The Dirac delta, denoted $\delta(t)$ or drawn as an upward pointing arrow, is a function over a continuous domain, in other words $t \in \mathbb{R}$. In this class, we're dealing exclusively with discrete-time signals, so in the time domain, we'll have Kronecker deltas. However, when we take the DTFT of a discrete-time signal, we get a result in the continuous frequency domain (ω), so if we have an impulse in the continuous frequency domain, it will be a Dirac delta, $\delta(\omega)$.

Spiritually, these functions are the same though.

3 Properties of Discrete Signals

3.1 Periodicity

As signal $x[n]$ is periodic if there exists a number n_0 such that $x[n] = x[n - n_0]$ for all n. The period of the signal is n_0.

For sinusoidal functions (i.e. $cos\,(\omega n + \phi)$), the function is periodic iff ω is a rational multiple of π, i.e. $\omega = \frac{M}{N}\pi$ for integers M and N.

3.2 Boundedness

A signal is bounded if there exists a positive, finite number B such that $|x[n]| \leq B$ for all n.

3.3 Causality

A signal $x[n]$ is causal if $x[n] = 0$ for $n < 0$.

3.4 Symmetry

A signal $x[n]$ is symmetric if $x[n] = x[-n]$. Signals with symmetry are also called "even".

If instead a signal has the property that $-x[n] = x[-n]$, then it is called "odd".

4 Properties of the Delta Functions

4.1 Sampling Property of the Kronecker Delta

$\delta[n - n_0]x[n] = \delta[n - n_0]x[n_0]$

4.2 Sifting Property of the Kronecker Delta

$$\sum_{n=-\infty}^{\infty} \delta[n - n_0]x[n] = x[n_0]$$

4.3 Scaling Property of the Dirac Delta

$\delta[2n] = \delta[n]$ Note that there is no scaling factor!

4.4 Sampling Property of the Dirac Delta

$\delta(t - t_0)x(t) = \delta(t - t_0)x(t_0)$

note: The result is a function!

4.5 Sifting Property of the Dirac Delta

$$\int_{-\infty}^{\infty} \delta(t - t_0)x(t) = x(t_0)$$

note: The result is a scalar!

4.6 Scaling Property of the Dirac Delta

$\delta(\alpha t) = \frac{1}{|\alpha|}\delta(t)$ for any scalar α

5 Importance of Norms and Inner Products

The above delta properties will prove very useful in this course for proving digital Fourier Transform properties. These properties are especially useful in DSP but are used less often in higher lever classes.
In the second lecture, the definitions of l2 norm and euclidean inner product are given. Though these definitions are seemingly dry, norms and inner products are extremely useful. Nate uses norms and inner products quite regularly. Websites like Twitter and Netflix use different norms to determine a measure of similarity in datasets. In Machine Learning (ML), inner products are of utmost importance. If a dataset can be represented using inner products, ML algorithms can efficiently compute results that would other require infinite memory/resources.

6 MATLAB basics

6.1 How to create a digital signal

To create a signal (vector) called sigVec in MATLAB:

» sigVec = [1 2 3 4 8 9 12 8]

6.2 How to transpose a signal

To flip the orientation of sigVec, use an apostrophe to denote a transpose

» sigVecFlipped = sigVec'

6.3 MATLAB indexing

To index a matrix called sigVec, use parenthesis:

sigVec(row,column)

sigVec is the 1 x 8 matrix from 0.1. Suppose we want to index the 3rd entry.

» thirdEntry = sigVec(3)

Alternatively

» thirdEntry = sigVec(1,3)

6.4 Using functions

MATLAB has many built in functions for things that you might want to do. Often, these functions will require an input and produce an output. Function calls in MATLAB usually take the form:

» output = function(input)

OR

» [output1, output2, ...] = function(input1, input2, ...)

6.5 Colon operator as a counting tool

The colon operator is "one of the most useful operators in MATLAB" according to Mathworks. This operator does not have a strict definition, but becomes very intuitive after using it a few times. The colon operator can be used in the following ways:

To count up or down by 1

» startingNumber : endingNumber

To count up or down by Increment

» startingNumber : Increment : endingNumber

Example: The colon operator can be used to create large signal vectors. Suppose we want to create a signal called n, which simply counts to 30 starting from 0. We could type out:

» n = [0 1 2 3 4 5 6 7 ...]

Unfortunately, your GSI gets a severe hand cramp from typing every digit manually. He explains the colon operator can be used to count up more efficiently than typing by hand:

» n = 0:30;

Now suppose we want to create an array that counts backwards from 30 to 0, skipping every odd number. The colon operator can be used for signals of this type. We are not limited to counting up or down by one.

n = 30:-2:0;

6.6 Colon operator for indexing

The colon operator can also be used to index or select certain portions of a signal. Recall sigVec from 0.1

sigVec = [1 2 3 4 8 9 12 8]

The colon operator can be used as a shorthand to select all terms in a row or column.
Example:

» x = sigVec(:)

// x = [1 2 3 4 8 9 12 8]

» x = sigVec(:,1)

// x = [1]

» x = sigVec(1,:)

// x = [1 2 3 4 8 9 12 8]

Example:

Create a matrix in MATLAB using the following command:

» Matrix = [1 2 3; 3 5 1; 1 5 3];

Using the colon operator we can select parts of Matrix.

» x = Matrix(2,:)

// x = [3 5 1]

» x = Matrix(:,3)

// x = [3 1 3]

» x = Matrix(1:2:3,3)

// x = [3 3]

7 MATLAB Problems

7.1 Generate a Signal

Using MATLAB, generate a signal of length 3 using the number 3, 5, and 1.

7.2 Plot a signal

Using MATLAB, plot the signal you generated in 1.1 as a stem plot.

7.3 Generate a longer signal

First type the following commands into MATLAB:
(Note: put a semicolon at the end of your command to suppress output)

» fs = 44100;

» duration = 3;

» t = ??

In MATLAB, create the variable t using two colon operators, fs, and duration.

Using the new variable t, create a cosine wave using the cos( ) function. Save the cosine wave as a new variable.

7.4 Music Processing Basics! (Using the sound function)

$$x[n] = \begin{cases} x(n/(fs)), & n \,\epsilon\, \mathbb{Z} \text{ (fs is the sampling frequency)} \\ 0, & otherwise \end{cases}$$

Suppose x(t) = cos(2π*400t)

What is x[n]?
Create x[n] in MATLAB using the variable t from 1.3. Name the variable x.

Type the command plot(t,x) (you may need to zoom in or out)
Type the command sound(x,44100) or sound(x,fs)

7.5 Analyze the signal

Use Google to search for the Fast Fourier Transform MATLAB command.
Figure out how to use this function on the cosine wave x from 1.4.

The Fast Fourier Transform function will return an output variable, name this variable "Fourier".
Now type the following command:

» plot(abs(fftshift(Fourier)))

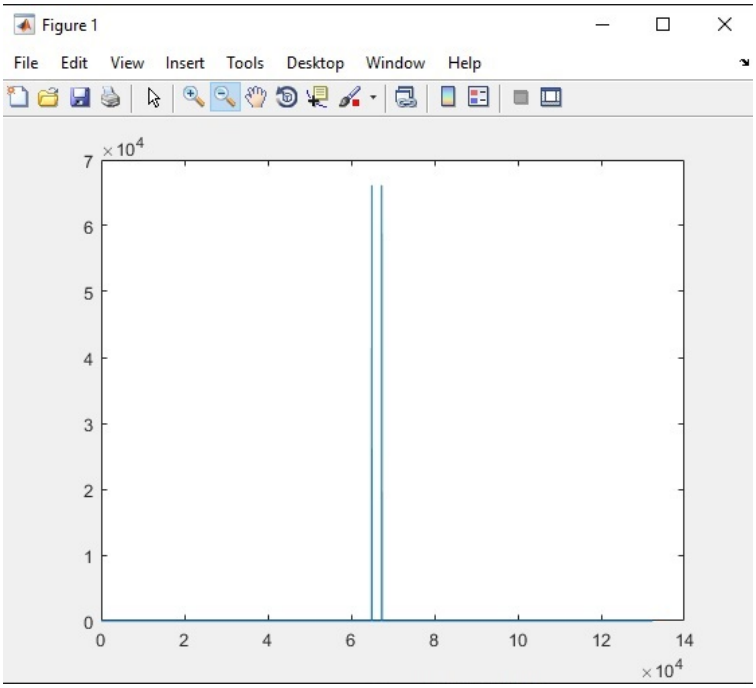

Figure 1: If you did everything right, you should end up with a plot like this

Theoretical Problems (optional)

The following problem was compiled by Mai Le

8 Alternate Expressions for Sequences

Let $x[n] = \begin{cases} \left(\frac{1}{2}\right)^n, & n \text{ nonnegative multiple of } 4 \\ -\left(\frac{1}{2}\right)^n, & n \text{ nonnegative multiple of } 2, \text{ but not a nonnegative multiple of } 4 \\ 0, & \text{otherwise} \end{cases}$

Express $x[n]$ mathematically in three different ways