ECE 3500: Fundamentals of Signals and Systems (Spring 2019)

Lab 2: Convolution, Correlation, and Finding Signals

Files necessary to compute this assignment: rudenko_01.mp4 to rudenko_39.mp4, get_tune.m, get_tune.p

Deliverables Due: The midnight of the day for your assigned lab session between Feb. 5 and Feb. 7.

Introduction

Discrete-time convolution is defined by

$$y[n] = x[n] * h[n] = \sum_{m=-\infty}^{\infty} x[n-m]h[m] \ .$$

Convolution describes the process of a signal $x[n]$ passing through a linear, time-invariant system defined by an impulse response $h[n]$. Convolution is a central concept that describes every linear, time-invariant system that we create and use. Convolution describes how the circuits affect a given voltage input, how radio waves travel through the atmosphere, how to blur and sharpen images, how to recognize images/faces videos, etc. While many of these systems do not use convolution directly (we often use faster algorithms that indirectly perform convolution), understanding how and why convolution works is essential to all of them.

In this lab assignment, we will focus on understanding convolution and how to use it to design and analyze systems. We will create several system impulse responses and study how each system affects a given signal. We then study a widely used application of convolution, known as "template matching" or "matched filtering," that is often used to locate images, radar or sonar signals, and musical segments in noisy data sets.

This lab assignment has three learning objectives:

1. To improve your understanding of convolution
2. To introduce you to the concept of correlation and template matching
3. To design your own "Name that tune" algorithm

Your experiment will be divided into three parts, each associated with an individual objective.

Deliverables

This lab assignment has two deliverables:

1. Your custom "lastname_firstname_lab2.m" file
2. Your answers to questions / inclusions of plots specified throughout the lab
   (see grading rubric at the end of this document for a complete list of the questions)
   Note: **there is no special format for this submission, but be clear regarding what answer corresponds to what question.**

Please electronically submit these deliverables to your lab teaching assistant via canvas.

Part 1: Convolution for Signal Manipulation

Create a Signal: Open the MATLAB editor and create a script file. In your script, create a discrete-time signal $x[n]$ that is 20 samples in length. Start with this signal containing all zeros (i.e., we are "initializing" the signal). The easiest way to do this is to write:

```
>> x = zeros(20,1);
```

In your script file, make your signal $x[n]$ equal to

$$x[n] = u[n-3] - u[n-8] - \delta[n-17] \ .$$

There are multiple ways that you can do this. Choose the way that works best for you.

Manually plot $x[n]$ and then plot `x` in MATLAB using the commands

```
>> n = 0:19;
>> stem(n, x);
>> xlabel('Samples');
>> ylabel('Amplitude');
```

Confirm that your plot and MATLAB plot match.

Note: **The first index in MATLAB (i.e., `x(1)`) corresponds to time $n = 0$ (i.e., $x[n]$). Therefore, keep in mind that our MATLAB index is always one value ahead of our n.**

Convolve with a Delta Function: Now create a system impulse response $h[n]$ that is also 20 samples in length. Create the impulse response `h` in MATLAB such that $h[n]$ and $y[n]$ are defined by

$$\begin{aligned} h[n] &= \delta[n] \\ y[n] &= x[n] * h[n]. \end{aligned}$$

Perform the convolution in MATLAB by using

```
>> y = conv(x, h)
```

Note: **The output signal $y[n]$ should now have a length of $N = 39$. We will see why in the next subsection.**

Convolve with a Shifted Delta Function: Create a new impulse response $h[n]$ that is also 20 samples in length. Create the impulse response `h` in MATLAB such that

$$h[n] = \delta[n - k_1] \ ,$$

for $k_1 = 5$. Determine and plot the convolution $y[n] = x[n] * h[n]$ in MATLAB. Repeat this for $k_1 = 6$, $k_1 = 10$, and $k_1 = 19$.

1.1. Answer: What is relationship between your input $x[n]$, your output $y[n]$, and k_1?

1.2. Answer: Why is it important to have an output length of $N = 39$?

Convolve with Two Delta Functions: Create a new impulse response $h[n]$ that is also 20 samples in length. Create the impulse response `h` in MATLAB such that

$$h[n] = \delta[n] + \delta[n - k_2] \ ,$$

for $k_2 = 19$. Determine and plot the convolution $y[n] = x[n] * h[n]$ in MATLAB. Repeat this for $k_2 = 17$, $k_2 = 14$, and $k_2 = 11$.

1.3. Answer: What is relationship between your input $x[n]$, your output $y[n]$, and k_2?

1.4. Answer: What does this system do?

Convolve with a Box (Running Averaging): Create a new impulse response $h[n]$ that is also 20 samples in length. Create the impulse response `h` in MATLAB such that

$$h[n] = (1/3)(u[n] - u[n - 3]) \ .$$

Determine and plot the convolution $y[n] = x[n] * h[n]$ in MATLAB.

1.5. Answer: What does this system do? What application might this be used in?

Convolve with a Difference (Edge Detection): Create a new impulse response $h[n]$ that is also 20 samples in length. Create the impulse response `h` in MATLAB such that

$$h[n] = -\delta[n] + 2\delta[n - 1] - \delta[n - 2] \ .$$

Determine and plot the convolution $y[n] = x[n] * h[n]$ in MATLAB.

1.6. Answer: What does this system do? What application might this be used in?

Part 2: Convolution for Finding Signals

Consider the following small change to the convolution equation

$$y[n] = x[-n] * h[n] = \sum_{m=-\infty}^{\infty} x[n+m]h[m]$$

This is often known correlation (or cross-correlation). The correlation has several very useful properties. It is the key component of many pattern recognition algorithms, such as facial recognition. We will illustrate how this pattern recognition works in the following subsections.

Creating Signals: Create the following signals in MATLAB:

```
>> x1 = sin(pi/10*(0:19));
>> x2 = sin(pi/5*(0:19));
>> x3 = sin(pi/2*(0:19));
>> x4 = (-1).^(0:19);
```

Plot each of these signals individually with the `stem` command.

Now concatenate each of these four signals:

```
>> z = [x1 x2 x3 x4];
```

The Auto-Correlation: First, perform the correlation of `x1` with itself. This is known as the auto-correlation. Perform the auto-correlation with the following two sets of commands:

```
>> a1 = conv(fliplr(x1), x1);
>> b1 = xcorr(x1, x1);
```

The results from `conv` should be equivalent to the results from `xcorr`, but with one small difference: `xcorr` is much faster for large signals. You may want to remember this for Part 3 of the lab.

The correlation can be thought of as a ***measure of similarity*** between $x[m]$ with a version of $h[m+n]$ that is delayed by $-n$ samples. The auto-correlation is always maximum at $n = 0$ because the signals are identical at this delay. The x-axis for a correlation is referred to as the "lag" or "delay." The correlation can be plotted (with the correct x-axis) through the following commands.

```
>> n = -(20-1):(20-1);  % 10 is from the maximum length between x1 and h
>> stem(n, a1)
>> xlabel('Lag [samples]')
>> ylabel('Amplitude')
```

Plot the result for both `a1` and `b1` to confirm that they are equal. Also plot the auto-correlations for `x2`, `x3`, and `x4`.

2.1. Include: The four plots resulting from the auto-correlation of `x1`, `x2`, `x3`, `x4`. Include correct axes and labels.

The Cross-Correlation (Template Matching): Now perform a correlation between `z` and `x1` with the command:

```
>> z1 = xcorr(z, x1);
```

Plot the result for `z1` using

```
>> n = -(80-1):(80-1);  % 40 is from the maximum length between z and h
>> stem(n, z1)
>> xlabel('Lag [samples]')
>> ylabel('Amplitude')
```

Note that this process is also known as "template matching" (particularly in image processing) or "matched filtering" (particularly in communications). Replace `x1`, `x2`, `x3`, `x4` and repeat the `xcorr` and plotting process.

2.2. Include: The four plots resulting from "template matching" `x1`, `x2`, `x3`, `x4` with `z`. Include correct axes and labels.

2.3. Answer : How do these results compare with the autocorrelations?

2.4. Answer : What is the significance of the maximum values?

Verify your conclusions with your TA before moving on.

Part 3: Name that Tune

Reading audio files: Included with this lab is 39 mp4 files by Peter Rudenko from the free music archive (http://freemusicarchive.org/). To load these files into MATLAB use the `audioread` command:

```
>> [y, Fs] = audioread(['rudenko_01.mp4']);
```

If you want to use a variable to control the file number, you can use the commands:

```
>> ii = 1;
>> [y, Fs] = audioread(['rudenko_' num2str(ii, '%02i') '.mp4']);
```

The output of the `audioread` function provides `y` (the audio signal) and `Fs` (the sampling rate of the audio signal). The sampling rate is the number of discrete values (i.e., samples) per second in the signal.

To visually see the audio signal with correct axis labels, type the commands

```
>> t = 1/Fs:1/Fs:length(y)/Fs;
>> plot(t, y);
>> xlabel('Time [sec]')
>> ylabel('Amplitude')
```

The variable `t` describes the time-axis of the audio.

Running P-files: Included with this lab is a p-file (an obfuscated m-file) function

```
>> [findme, Fs] = get_tune(uid);
```

Note that to run this function, all 39 Rudenko files must be in the current directory. The input to this function `uid` will be a string with your uid. The output of the function `findme` is an audio segment from the one of the 39 audio files. The second output `Fs` is the sampling rate of the audio (this should be equal to the sampling rate of the full audio).

Replace `uid` with your uID (in single quotes) and run this function to extract an audio segment that is unique to you. Use the command

```
>> sound(findme, Fs)
```

to play the audio segment.

Your Task: Name that Tune: Write a MATLAB script that utilizes the capabilities of convolution that we learned in the previous exercises to determine which music file your unique audio segment `findme` originates. Note that even when using the `xcorr` function, your script may take a few minutes to fully execute. Save this script in a MATLAB m-file named "lastname_firstname_lab2.m", where "lastname" is replaced with you last name and "firstname" is replaced with your first name.

3.1 Submit: Your custom "lastname_firstname_lab2.m" file

3.2 Answer: From which music file did your music segment originate?

3.3 Answer: Describe your system / algorithm with one-three paragraphs. Do not refer to ***any*** specific MATLAB code. Your description should be ***specific*** enough that a reader could replicate your algorithm but ***general*** enough that a reader could replicate the algorithm in any programming language (C++, Java, etc.) while not knowing MATLAB. Assume your audience has the technical knowledge of a typical electrical engineering sophomore (with the exception of MATLAB knowledge).

I suggest you include two paragraphs with the following information:

> **Paragraph 1:** Describe your data (music) set, your music segment, and how they are related. This paragraph should give readers an understanding of what type of data you are working so that they can collect a similar (or possibly identical) data set. You should provide some summary specifics about the data (the author of the music, the type of music, etc.).
>
> **Paragraph 2:** Describe your algorithm and how you applied it. You may want to include the equation for correlation to succulently describe it. You may then want to describe how you applied the correlation equation and then how you determined from which music file your music segment originated.

3.4 Answer: Describe your result from Part 3 in one paragraph. Does your algorithm yield the correct result? Justify your claim that your algorithm did well or poorly (i.e., how do you know?). Also, discuss one potential scenario in which your algorithm would fail in a real application and explain why this scenario causes it to fail.

Grading Rubric

Criteria	Points Possible	Points
Part 1 Deliverables	**30**	
Include Answer to 1.1: What is relationship between your input $x[n]$, your output $y[n]$, and k_1?	5	
Include Answer to 1.2: Why is it important to have an output length of $N = 39$?	5	
Include Answer to 1.3: What is relationship between your input $x[n]$, your output $y[n]$, and k_2?	5	
Include Answer to 1.4: What does this system do?	5	
Include Answer to 1.5: What does this system do? What application might this be used in?	5	
Include Answer to 1.6: What does this system do? What application might this be used in?	5	
Part 2 Deliverables	**24**	
Include 2.1: The four plots resulting from the auto-correlation of `x1` with `x2`, `x3`, `x4`. Include correct axes and labels.	7	
Include 2.2: The four plots resulting from "template matching" `x1`, `x2`, `x3`, `x4` with `z`. Include correct axes and labels.	7	
Include Answer to 2.3: How do these results compare with the autocorrelations?	5	
Include Answer to 2.4: What is the significance of the maximum values?	5	

Continued on next page...

Criteria	Points Possible	Points
Part 3 Deliverables	**46**	
Submit: Your custom "lastname_firstname_lab2.m" file.	6	
Include Answer to 3.1: From which music file did your music segment originate?	10	
Include a 1-3 paragraphs on your methodology: Describe your system / algorithm. Do not refer to ***any*** specific MATLAB code. Your description should be ***specific*** enough that a reader could replicate your algorithm but ***general*** enough that a reader could replicate the algorithm in any programming language (C++, Java, etc.) while not knowing MATLAB. Assume your audience has the technical knowledge of a typical electrical engineering sophomore (with the exception of MATLAB knowledge). I suggest you include two paragraphs with the following information: **Paragraph 1:** Describe your data (music) set, your music segment, and how they are related. This paragraph should gives readers an understanding of what type of data you are working so that they can collect a similar (or possibly identical) data set. You should provide some summary specifics about the data (the author of the music, the type of music, etc.). **Paragraph 2:** Describe your algorithm and how you applied it. You may want to include the equation for correlation to succulently describe it. You may then want to describe how you applied the correlation equation and then how you determined from which music file your music segment originated.	20	
Include one paragraph on your results: Describe your result from Part 3. Does your algorithm yield the correct result? Justify your claim that your algorithm did well or poorly (i.e., how do you know?). Also, discuss one potential scenario in which your algorithm would fail in a real application and explain why this scenario causes it to fail.	10	