Zewail City of Science and Technology

CIE-442

Project
Spectrum Analyzer & Filter Design

1.1 Objectives

- Understand the operation of Spectrum Analyzer.
- Understand the need of Discrete convolution.
- Understand the concept of correlation.
- Understand the effect of Windowing, averaging and so forth on both the time domain and frequency domain signals.
- Understand the least square method for designing an FIR filter.
- Use MATLAB as a simulation software.

1.2 Introduction

Fourier transform is one of the most important tools, if not the most important, in communications. It is used to transform the time domain signals to their frequency counterparts which helps in analyzing the spectrum of the information signals. Knowing the spectrum of the signal is crucial to be able to apply the correct digital signal processing algorithms without wrongly affecting the information signal and keep the info intact.

One of the devices used in real-life to visualize the signal's frequency contents is the Spectrum analyzer which is the main topic of this project. One of the signal types of interest is audio files. Audio files are analog in their nature; hence, they require an analog to digital converter to be able to process it digitally. In this project, the audio signal is given to you sampled or in other words, it is ready for signal processing in the digital domain (No ADC is required in this project). The spectrum analyzer, then, takes the sampled audio file and perform all the necessary measurements to visualize the different frequency components and the effect of windowing and averaging on such signal.

This project aims at understanding the need for spectrum analyzers and their use in real-life applications. Furthermore, it will help you understand the DFT process, windowing and the leakage concept. In addition, it will help in looking at different methods in visualizing your signal and compare different methods against each other. Moreover, the discrete convolution feature will be of an asset to understand how convolution works and can aid you in understanding the channel effects on signal transmission. Furthermore, the filter design section will help you in understanding the design procedure of digital filters and how they affect signals.

1.3 Project statement

This project is divided into three main sections: **(Note: The Bold items are related to Part 2)**

- Discrete Convolution.
- Spectrum Analyzer.
- FIR filter Design using Least Squares method.

It is required to implement a GUI in MATLAB to mimic the operation of real-life spectrum analyzer with the Discrete Convolution being an additional feature.

1.3.1 Draft Layout

- Your GUI should start by giving three choices for the user to either perform Discrete Convolution or use Spectrum Analyzer or design an FIR filter.
- If the choice is Discrete Convolution, then a new GUI window will open and:
    - You Should give the user a drop-down list of your available functions to perform convolution with.
    - It is expected to have two drop-down lists to allow the user to choose the two functions that will be convolved together.
    - Additional features can be added to allow the user to choose different parameters like the signal amplitude, the signal width or period, the exponential power and so forth.
    - You should provide at least four plots on your GUI in this section, the main signal, the signal to be convolved with, the result of the convolution and **the animated steps for performing your convolution (Bonus).**
    - Hint: You must include at least three different functions in this section.
- If the choice is Spectrum Analyzer, then a new GUI window will open and:
    - The Layout should be as follows:
        - **Inputs: (Part 2)**
            1. An input file option should be available to allow the user to choose an audio file (that will be given to you to test on).
            2. Built-in functions to try them as examples or for the purpose of demonstration (two functions will be enough).
        - Two plots to plot the time domain signal and the frequency domain signal.
        - A comparison mode tab.
        - **Display options for the plots (linear and logarithmic, magnitude and phase, real and imaginary). (Part 2)**
        - **Calculation Tab. (Part 2)**
        - Window choice and window length.
        - N-DFT choice (Maximum is 64K, one of your choices should be the maximum and you can choose any other three values of the form 2^n).
        - **Markers. (Part 2)**

    - **Option to choose the frequency bands (Spans). (Part 2)**
  - The frequency plots should be drawn versus your actual frequency not the frequency bins ($f_{analysis}(m) = \frac{mf_s}{N}$).
  - The windowing option:
    - You should try Rectangular, Triangular, Hanning and Hamming windows.
    - You should give the user different choices for the window's length.
  - **The Calculation mode: (Part 2)**
    - **Calculate the power (within a certain span).**
    - **Perform RMS averaging.**
    - **Indicate the presence or absence of a DC bias.**
    - **Find the maximum point in your spectrum (Peak).**
  - N-DFT choice:
    - You should give the user valid N-DFT choices or allow the user to enter it with a maximum value of 64K.
    - The N is related to RBW and f_s. You should view your RBW.
  - Comparison mode:
    - You should have at least four plots in this mode (It is recommended that this mode opens up in a new a GUI). The four plots are two for the time domain comparisons and two for the frequency domain comparisons.
    - The idea of this mode is to compare the same signal under different windowing and show the results side by side for both the time domain and the frequency domain.
    - The effect of the window length and the N should be also available in the comparison mode. In other words, you should be able to compare the same window but with different length on the same signal and same wise for the N.
- **If the choice is design an FIR filter, then a new GUI will open and: (Part 2)**
  - **You should give the user the option to enter the length of the desired filter response (for example denoted L).**
  - **You should give the user the option to enter the length of the designed filter response (for example denoted N).**
  - **You should plot the magnitude spectrum of both the desired and the designed filters on the same plot with different colors to show the difference.**
  - **You should give the user the option to show the magnitude response in both linear and log scale.**
  - **You should calculate your least square error.**
  - **You should give the user the option to design the filter using the weighted least squares method by giving him a set of choices (minimum three choices) for the weighting coefficients. (Bonus)**
  - **Note: L > N.**

1.3.2 Features:

- Time and Frequency domain analysis.
- Discrete time convolution.
- Windowing effect.
- Comparison mode.
- **Calculation mode. (Part 2)**
- **RMS averaging. (Part 2)**
- **Peak finding. (Part 2)**
- **Bias finding. (Part 2)**
- **Different display options. (Part 2)**
- **Markers. (Part 2)**
- **Frequency band choices. (Part 2)**
- **Different input options. (Part 2)**
- **FIR filter design using least square method. (Part 2)**

1.3.3 Bonus

- Implement your own code to perform Fast Fourier Transform (FFT) and use it in your project. You should verify your code by making sure that it performs the correct FFT and can be compared to the built-in function.
- Convolution Animation.
- Error Handling.
- Weighted least square method for FIR filter design.

1.4 Requirements

- Implement the MATLAB code for all the above tasks and include all the MATLAB files in your online submission.
- Write a report that explains the operation theory and include your code and documentation within your report.

1.5 Important notes

- The code should be readable and all the variable names are meaningful.
- The code should be commented to explain the functionality of your code.
- The report should be formatted in an organized way (i.e: it should include table of contents, introduction, theory, results and discussions, conclusion, code section and so forth).
- The pages should be numbered.
- Don’t forget to include your references.
- The quality of your report will be assessed.
- Documentation should be clear and any used variables within the theory should reflect on your code.