Electrical Communications Systems

Course No. ECE.09.433

Laboratory Project 1

Waveform Synthesis and Spectral Analysis

Objectives

This project has 4 parts. In Part 1, you will generate arbitrary waveforms with specified SNRs. In Part 2, you will study the differences between the Continuous Fourier Transform (CFT) and the Discrete Fourier Transform (DFT). In Part 3, you will synthesize AM and FM bandpass signals and analyze their spectra. In Part 4, you will capture and analyze the spectra of a segment of any piece of music that you like.

Software
***Matlab:* https://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf**

Part 1: Digital synthesis of arbitrary waveforms with specified SNR

Background

- SNR in dB = $10 \log_{10} (\sigma_s^2/\sigma_n^2)$; where σ_s^2:signal variance, σ_n^2: noise variance
- Given signal, s(t), find σ_s^2
- Compute required σ_n^2
- Generate noise signal, $n(t) = \sigma_n N(0,1)$, where N(0,1) is a Normally (Gaussian) distributed random variable with Zero mean and Unit variance
- Message signal with desired SNR, m(t) = s(t) + n(t)

Procedure Overview

- Synthesize a 1 second A-Sharp sinusoidal tone (466.16 Hz) sampled at 8 kHz.
- Plot this waveform; observe and listen using the *sound* command in Matlab.
- Corrupt this signal with a Gaussian noise source to get an SNR of 10 dB. Observe, listen and save waveform as before.
- Synthesize 1 cycle of the noisy waveform.
- Vary the frequency, amplitude.
  Repeat experiment with various SNRs.

- Experiment with other waveforms:
  - $s(t) = A_c[1 + \cos(2\pi f_m t)]\cos(2\pi f_c t)$ with varying A_c, f_m and f_c.
  - $s(t) = A_c\cos[2\pi f_c t + \beta\sin(2\pi f_m t)]$ with varying A_c, β, f_m and f_c.

Example Matlab Code

```
%ECOMMS Lab Project 1 Example
%S. Mandayam, Rowan University
%This program generates a 1-second duration
%Asharp signal (466.16 Hz) with a specified SNR

%Specify SNR
snr=10;

%Generate Asharp signal
t=[0:1/8e3:1.0]';
s = 0.5*sin(2*pi*466.16*t);
sound(s);


%Compute signal variance
var_s = cov(s);

%Calculate required noise variance
var_noise=var_s/(10^(snr/10));

%Generate noise
n=sqrt(var_noise)*randn(length(s),1);
sound(n);

%Add signal to noise and generate message
m=s+n;
sound(m);
```

Part 2: Comparison between CFT and DFT (FFT)

Consider the signal shown in Figure 1:

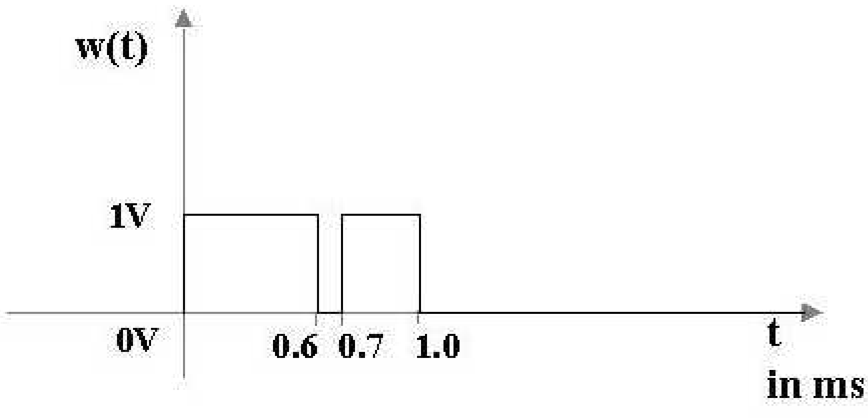

Figure 1: Time-domain signal

- Model the signal as a continuous-time function and plot.
- Obtain, **analytically**, the CFT of the signal, and plot.
- Based on your observations of the frequency components of the signal, determine the maximum sampling period/minimum sampling frequency that will allow reconstruction of the continuous-time signal.
- Plot the DFT magnitude spectrum using Matlab's *fft* function.
- Attempt to reconstruct the original continuous-time function from its samples, either by:
    - Convolving the time domain samples with appropriate Sinc function (difficult), or
    - Windowing the Fourier transform and taking the inverse Fourier transform (easier).

  Comment on your results.

- What is the maximum sampling period/minimum sampling frequency that will allow reconstruction of the discrete-time signal from its DFT, which adequately represents the original continuous-time signal? Show plots of the discrete-time signals, the corresponding DFTs and reconstructions, for sampling intervals at, above and below this maximum allowed sampling period.

Part 3: Spectral Analysis of AM and FM Signals

- Synthesize a bandpass AM signal, $s(t) = A_c[1 + A_m\cos(2\pi f_m t)]\cos(2\pi f_c t)$ where $f_m = 5$ kHz and $f_c = 25$ kHz.
- Obtain and plot the spectral components of this signal using Matlab's *fft* function.
- Add noise to the RF signal, observe the signal in the time and spectral domains.
- Experiment with various f_ms, f_cs and SNRs.
- Synthesize a bandpass FM signal, $s(t) = A_c\cos[2\pi f_c t + \beta_f A_m\sin(2\pi f_m t)]$ where β_f = Frequency Modulation Index (choose initially = 10).
- Repeat earlier experiments.

Part 4: Spectral Analysis of Music Signal

Legally, download a segment of your favorite piece of music, import into Matlab and listen. Compute its spectrum in Matlab, observe.