Project 2 - Speech Detection with FIR Filters

ECE505, Spring 2020
EECS, University of Tennessee
(Due 03/03)

1 Objective

The project introduces a practical application where sinusoidal signals are used to transmit information, i.e., amplitude modulation (AM) for radio (e.g., speech). In this application, FIR filters can be applied to extract the information encoded in the waveforms. Through this project, you should have in-depth understanding of characteristics of FIR filters and review techniques like generating envelope, tone synthesis, amplitude modulation, etc.

2 Background

You need a little bit background on how amplitude modulation (AM) is used.

AM is often used to transmit a signal with low-frequency content using a high-frequency transmission channel. A common example is AM radio, where a relatively low-frequency signal such as a speech signal, with frequencies between 50 Hz and 4 kHz, is transmitted by radio waves at frequencies around 1 MHz. AM is performed by multiplying a high-frequency signal, referred to the *carrier*, by a low-frequency *message* signal, $m(t)$,

$$\begin{aligned} x(t) &= (1 + A\cos(2\pi f_m t))\cos(2\pi f_c t) \\ &= \cos(2\pi f_c t) + \tfrac{A}{2}\cos(2\pi(f_c + f_m)t) + \tfrac{A}{2}\cos(2\pi(f_c - f_m)t) \end{aligned} \tag{1}$$

where $m(t) = A\cos(2\pi f_m t)$ is the message signal, $\cos(2\pi f_c t + \theta)$ is the carrier signal. More complicated message signals may be analyzed if $m(t)$ can be written as summations of multiple sinusoidal components

$$\begin{aligned} x(t) &= (1 + \textstyle\sum_k A_k\cos(2\pi f_k t))\cos(2\pi f_c t) \\ &= \cos(2\pi f_c t) + \textstyle\sum_k \tfrac{A_k}{2}\cos(2\pi(f_c + f_k)t) + \textstyle\sum_k \tfrac{A_k}{2}\cos(2\pi(f_c - f_k)t) \end{aligned}$$

In communications jargon, the signal components at $f = f_c \pm f_m$ are called the *sidebands* and the signal component with a frequency of f_c is called the *carrier*.

In general, in AM, $f_c \gg f_m$. Note that an AM signal is very similar to the beat signals except for the carrier term.

AM demodulation is the process of recovering the message waveform from a modulated signal such as AM. There are numerous methods of demodulating a signal. In the following, we only discuss three and the project is focused on the LTI filter-based method.

Envelope detection (peak tracking). This method tries to return a waveform that passes through the peaks of the AM waveform using an RC circuit shown in Fig. 1, where the capacitor is charged through the diode during each positive cycle of the AM signal and the resistor discharges the capacitor during the negative part of the cycle. Fig. 2 shows a demodulated signal as compared to the message signal. The demodulated signal approximates the sinusoidal shape of the message signal after removing the DC level of one. The jagged appearance is due to the exponential discharge of the RC circuit. The MATLAB function `amdemod()` performs this type of demodulation.

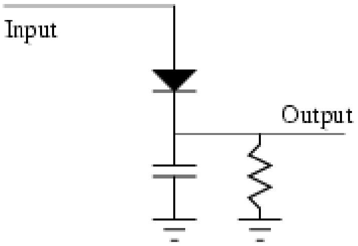

Figure 1: The demodulated AM signal using MATLAB's `amdemod()` routine.

LTI filter-based demodulation. It is possible to recover the message signal by *modulating the modulated signal* and then filtering. This is the basic principle used in commercial radios nowadays. Given an AM signal of the form Eq. 1, we can isolate the message signal by multiplying $x(t)$ by $\cos(2\pi f_c t)$,

$$\begin{aligned} x(t)\cos(2\pi f_c t) &= (1+m(t))\cos(2\pi f_c t)\cos(2\pi f_c t) \\ &= (1+m(t))(\tfrac{1}{2}+\tfrac{1}{2}\cos(2\pi(2f_c)t)) \\ &= \tfrac{1}{2}m(t)+\tfrac{1}{2}+\tfrac{1}{2}(1+m(t))\cos(2\pi(2f_c)t) \end{aligned} \tag{2}$$

based on which, by applying a filter to eliminate the high-frequency component and the DC component, the message signal $m(t)$ can be extracted.

Notch filters for demodulation. Notch filters can completely eliminate some frequency other than $\omega = 0$ or $\omega = \pi$. It is possible to make a notch filter with as few as three coefficients. If ω_{not} is the desired notch frequency, then the following

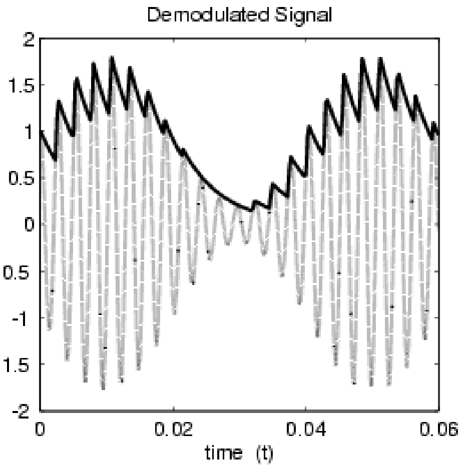

Figure 2: The demodulated AM signal using MATLAB's `amdemod()` routine.

length-3 FIR filter

$$y[n] = x[n] - 2\cos(\omega_{not})x[n-1] + x[n-2]$$

will have a zero at $\omega = \omega_{not}$ in its frequency response. For example, a filter designed to completely eliminate signals of the form $Ae^{j0.5\pi n}$ would have coefficients

$$b_0 = 1, b_1 = 2\cos(0.5\pi) = 0, b_2 = 1$$

3 Tasks

3.1 Task 1: Filter Design and Behavior Study (50 pt)

- Task 1.1 (5 pt): Write a function, `function [y] = firfilter(b, x)` to implement an FIR filter, where `b` is the filter coefficient vector, and `x` and `y` are the input and the resulting signal, respectively. MATLAB has a build-in function, `filter()`, but it is more generic and can implement a wide class of filters.

- Task 1.2 (5 pt): Write a function, `function [H] = freqres(b, omega)` to generate the frequency response of an FIR system, where `b` is, again, the filter coefficient, and `omega` is the radian frequency vector. You can also save the argument of `omega` as it is normally from $-\pi$ to π. MATLAB also has a build-in function, `freqz()`, but again is more generic than needed here.

- Task 1.3 (3 pt): Write a function, `function [x] = xn(A, omega, theta, n)` to generate a discrete-time cosine wave, where `A, omega, theta, n` are the amplitude, frequency in radian, phase shift, and length of the discrete-time signal, respectively.

- Task 1.4 (5 pt): In homework 3, you've written `nwz.m` that outputs the impulse response, magnitude response, phase response, and pole-zero plots given the pole and zero locations. The sample source code can be located at `http://web.eecs.utk.edu/~hqi/ece505/code/nwz.m`. Now write two other functions `n2z.m` that generates the pole-zero locations given the coefficients `b` and `a`, i.e., `[Z, P] = n2z(b,a);` and `z2n.m` that generates the coefficients given the pole-zero locations, i.e., `[b, a] = z2n(Z,P).`

- Task 1.5 (5 pt): Generate a 30-sample cosine wave, `xn1`, with $A = 7$, $theta = \pi/3$, and $omega = \pi/8$. Use `firfilter()` to implement the following first-difference filter with a gain of 5,

  $$y[n] = 5x[n] - 5x[n-1].$$

  In a figure with 3 × 2 subplots

  - Plot $xn1$ using MATLAB's built-in function `subplot()` and `stem()`.
  - Plot $yn1$ in another subplot.
  - Plot the frequency response of the filter, generated from `freqres()` in a 3rd `subplot()`.
  - Plot the phase response of the filter in a 4th subplot
  - Use `n2z` to find the pole/zero locations and provide the pole-zero plot in the 5th subplot
  - Plot the impulse response in the last subplot.

  Explain how the amplitude, frequency, and phase of the input signal should be changed by the frequency response and compare it with the plot of `yn1` to see if the plot confirms your analysis. Explain what the zero locations should be distributed and confirm that with the plot.

- Task 1.6 (7 pt): If a system is made of two subsystems cascaded together with the first subsystem being a 5-point averager, i.e., $w[n] = x^2[n]$, and the second subsystem the first difference FIR filter as in Task 1.5, i.e., $y[n] = 5w[n] - 5w[n-1]$, if we still use `xn1` as the input, plot all three waveforms, `xn1`, `wn`, and `yn` on the same figure with `subplot`. Make a sketch of the spectrum of these three signals. Observe signal `wn` and see if you can "see" the additional frequencies introduced by the squaring operation. Now replace the first-difference filter with the second-order FIR filter

$$y_2(n) = w(n) - 2\cos(0.25\pi)w(n-1) + w(n-2)$$

and observe how this new filter is able to remove certain frequency component(s). What type of FIR filter is it?

- Task 1.7 (10 pt): Use the same signal $w[n]$ generated from Task 1.6, design an 8-point bandpass filter to just pass the extra frequency components due to the square operation but eliminate all other frequencies (including the DC components). You can either start by specifying the pole/zero locations and find the filter coefficients, or specifying the filter coefficients and using which to find the pole/zero locations. What type of FIR filter is it?

- Task 1.8 (10 pt): Is the bandpass filter a minimum-phase system? If not, implement the system as a minimum-phase system cascaded by an allpass system. Provide the frequency response, impulse response, and pole-zero plots of both subsystems, as well as the input and output waveform with their spectrum. Assume the input is still `wn[n]`. Are the subsystems FIR filters? If so, what type?

3.2 Task 2: AM Waveform Detection and Filtering (50 pt)

- Task 2.1 (8 pt): Write a function `mx = ampMod(x, fs, fc)` that would modulate a speech signal `x` with a carrier signal of frequency `fc` and `fs` is the sampling frequency. The speech can be downloaded at `http://web.eecs.utk.edu/~hqi/ece505/project/proj2speech.mat`. The file contains three signals, `cc, mm, ss` representing the carrier signal, the modulated signal and the speech signal. Plot the waveform and the spectrum of the three signal, estimate the carrier frequency and perform modulation using `ss`.

- Task 2.2 (12 pt): Write a function `ampDemod_LTI` that would demodulate the modulated speech signal using LTI filters. Compare results with the one from `amdemod()`.

- Task 2.3 (15 pt): Filter the demodulated speech signal using two systems cascaded together with the first system being a 5-point average and the second system a first difference filter. Do it in both the time and the frequency domain and compare the difference.

- Task 2.4 (15 pt): Write a function `ampDemod_Notch` to demodulate the speech signal using notch filters then compare results with the one from `amdemod()` and LTI filters.

Acknowledgement

Part of this project is designed based on labs provided in McClellan, Schafer, and Yoder’s *DSP FIRST: A Multimedia Approach*.