EEE 304 – Lab 4

Amplitude Modulation and Demodulation

Introduction

This lab introduces one of the basic modulation schemes known as the Amplitude Modulation (AM), through a simple and intuitive example in LabVIEW. AM is a technique used in electronic communication, most commonly for transmitting information via a radio carrier wave In addition to the graphical approach followed to perform a certain task in the previous labs, this exercise introduces the MathScript functionality in LabVIEW. MathScript adds math-oriented, textual programming to LabVIEW that utilizes .m file script syntax, the syntax widely used by alternative technical computing software such as MATLAB. It is possible to combine the intuitive LabVIEW graphical programming with MathScript.

This lab contains two parts. The first part introduces Amplitude Modulation using a single tone (Sine Wave) as a message signal, which makes the understanding of the modulation technique very straight forward. The second part introduces the MathScript functionality and performs modulation of a speech signal. This will be helpful to understand the effects of modulation-demodulation on the quality of the reconstructed speech.

1. Basic Concepts

1.1. Modulation

Modulation is a process that causes a shift in the range of frequencies of a signal. There are many applications in which shifting the spectrum is necessary. Audio signal frequencies are so low that impracticably large antennas are required for effective power radiation. In this case, shifting the spectrum to higher frequencies by modulation would be a simpler solution. Another application where modulation-demodulation is necessary is Frequency Division Multiplexing (FDM), where several signals are transmitted simultaneously over a channel by sharing its frequency bands.

Communication that uses modulation to shift the frequency spectrum of a signal is known as carrier communication. In this mode, one of the basic parameters (amplitude, frequency, or phase) of a sinusoidal carrier of high frequency ω_c is varied in proportion to baseband signal $m(t)$.

1.2. Amplitude Modulation

Amplitude modulation is characterized by the fact that the amplitude A_c of the carrier $A_c cos(\omega_c t + \theta_c)$ is varied in proportion to the baseband (message) signal $m(t)$, the modulating signal. The frequency and the phase of the carrier are fixed. Here for simplicity, we can assume that $\theta_c = 0$. If the carrier amplitude A_c is made directly proportional to the modulating signal $m(t)$, we obtain the signal $m(t)cos(\omega_c t)$. The carrier signal $A_c cos(\omega_c t)$ is added to this signal, and the resulting signal $(A_c + m(t))cos(\omega_c t))$ is referred as the amplitude modulated signal. The reason for adding the carrier signal is to make the demodulation process easier.

Let the Fourier transform of the message signal be denoted by $M(\omega)$, i.e.,

$$F[m(t)] = M(\omega)$$

From properties of Fourier Transform, it can be easily observed that

$$F[m(t)cos(\omega_c t)] = [M(\omega+\omega_c) + M(\omega-\omega_c)] / 2$$

Hence, the modulated signal and its Fourier transform are given by

$$\phi_{AM}(t) = [(A_c+m(t))cos(\omega_c t)]$$

$$F[\phi_{AM}(t)] = [M(\omega+\omega_c)+M(\omega-\omega_c)] / 2 + \pi A_c [\delta(\omega+\omega_c)+ \delta(\omega-\omega_c)]$$

Recall that $M(\omega-\omega_c)$ is $M(\omega)$ shifted to the right by ω_c and $M(\omega+\omega_c)$ is $M(\omega)$ shifted to the left by ω_c. The process of modulation is described in Figure 1(a).

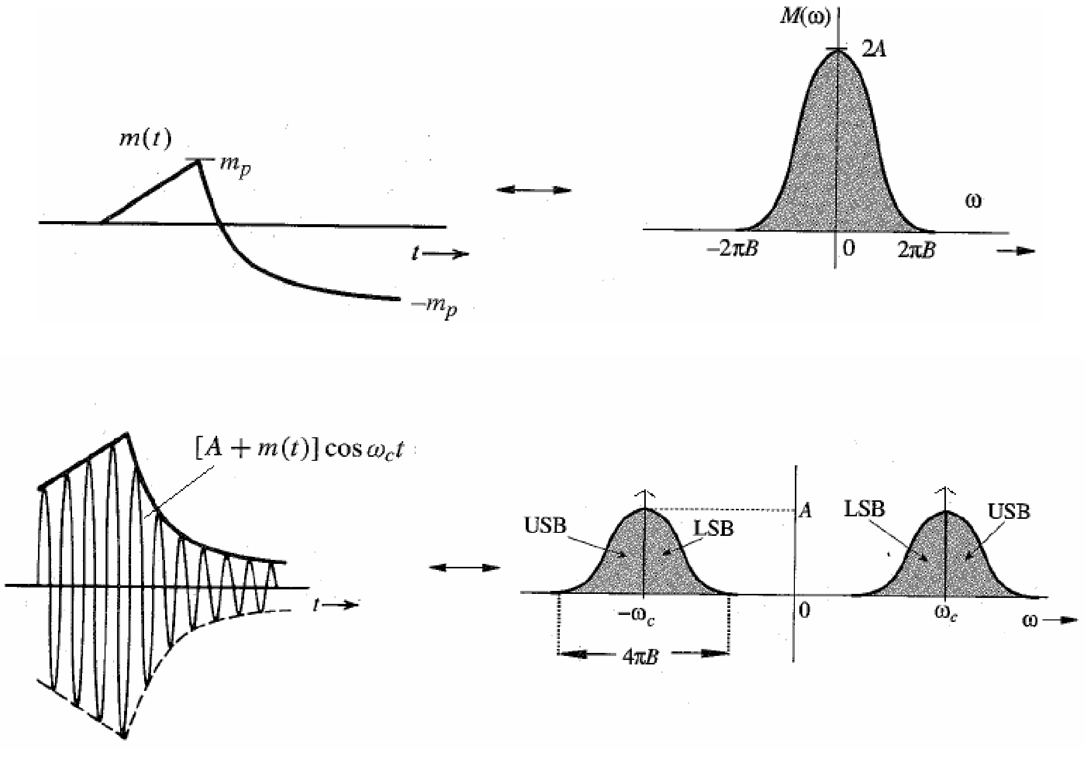


Figure 1 (a). Amplitude Modulation

If the bandwidth of the message signal, $m(t)$, is B Hz, then the bandwidth of the modulated signal is $2B$ Hz. The other important observation from the Figure 1 is that, the signal spectrum centered at ω_c contains three parts. A carrier centered at ω_c, a portion that lies above the ω_c (USB) and a portion lies below ω_c (LSB).

1.3. Amplitude Demodulation

Demodulation can be performed in two different ways.

- Coherent Demodulation: This assumes that the carrier signal with same frequency and phase can be generated at the receiver for demodulation.
- Non-Coherent Demodulation: This directly demodulates the received signal from its envelope without requiring the carrier signal.

In this exercise, we will use coherent demodulation, as it is very similar to the modulation process and easy to demonstrate and understand. As described, the scheme of modulation shifts the spectrum of the message signal by multiplying it with the carrier signal. Hence, demodulation requires the shifting of the spectrum back to the original location. To achieve this, we multiply again the modulated signal with the carrier signal.

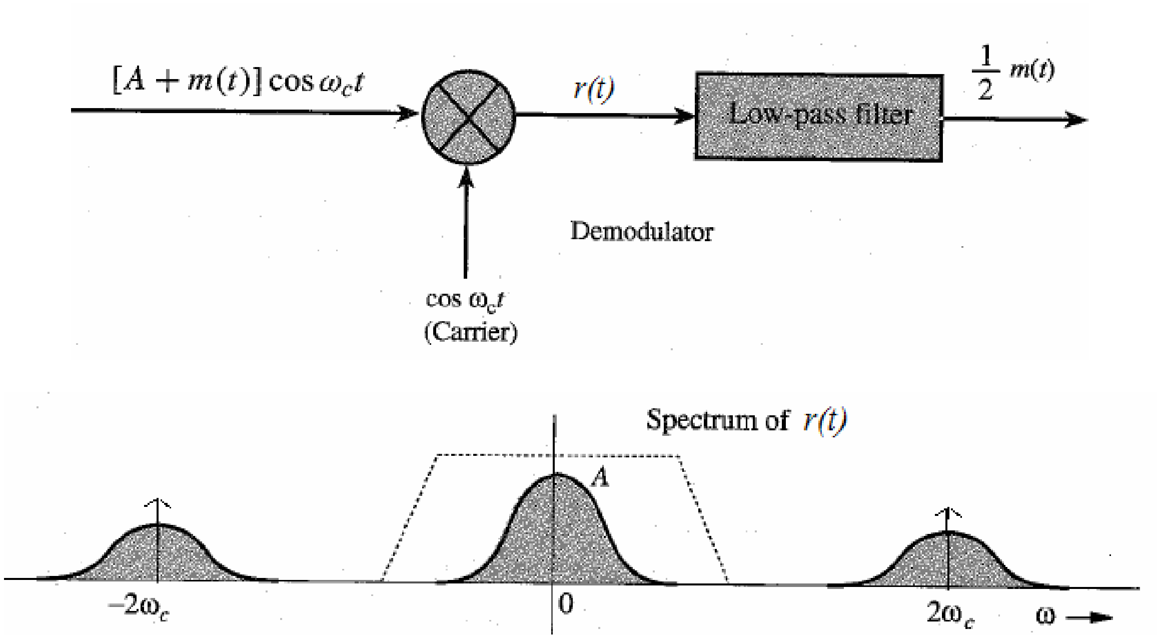


Figure 1(b) Demodulation

The received signal is $\phi_{AM}(t) = [(A_c + m(t))\cos(\omega_c t)]$. Multiplying it with $\cos(\omega_c t)$ results in the signal r(t), where r(t) is given by

$$r(t) = [(A_c + m(t))]\cos^2(\omega_c t) = [(A_c + m(t))](1 + \cos 2\omega_c t)/2.$$

and its Fourier transform,

$$R(\omega) = \frac{A_c}{2}\delta(\omega)+\frac{\pi A_c}{2}[\delta(\omega+2\omega_c)+\delta(\omega-2\omega_c)]+\frac{1}{2}M(\omega)+\frac{1}{4}[M(\omega+2\omega_c)+M(\omega-2\omega_c)].$$

It is evident from the equation above that the one part of the demodulated signal spectrum is centered at zero frequency and the other part is centered at $2\omega_c$. In order to retrieve the original message signal $M(\omega)$, and to remove the high frequency components, $r(t)$ is passed through a low pass filter with a cutoff frequency greater than B Hz.

1.4 Modulation Index

Modulation index (μ) is defined as the ratio between the peak value of the message signal and the amplitude of the carrier signal. This indicates by how much the modulated signal varies around its original level. The value of $\mu < 1$ results in under-modulation and $\mu > 1$ results in over-modulation. Over modulation results in erroneous signal reconstruction if non-coherent demodulation (envelope detection) is used, but in case of coherent demodulation, any value of μ provides perfect reconstruction.

2. Exercise – Amplitude Modulation and Demodulation

2.1. Using a sine wave as the message signal.

Now, as we have enough background of modulation theory, it is very straight forward to implement in LabVIEW. Figure 2 illustrates the basic block diagram for this modulation exercise.

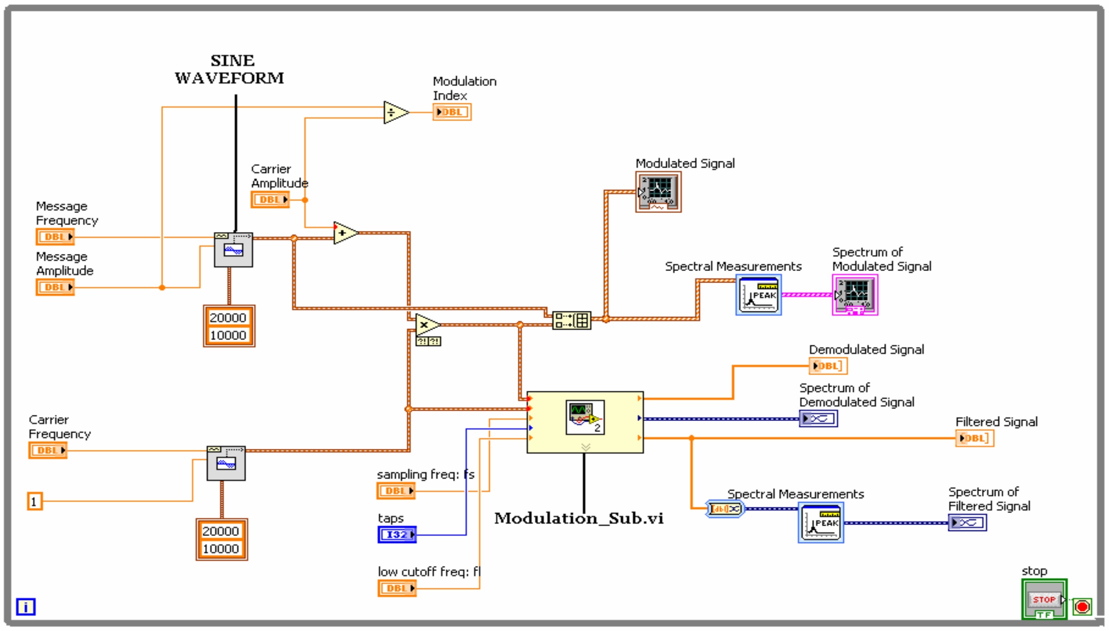
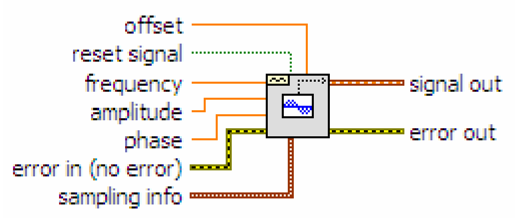


Figure 2. Basic block diagram for Amplitude Modulation

2.1.1 Generating message and carrier signals

In this exercise, a sine wave is used as the message signal and the carrier is also a sine wave. The first step in this exercise is to generate the waveforms with desired input frequencies and a specific sampling rate. Place two ***Sine waveform*** generator functions in your block diagram.



The sampling info to both the sine waveform generators use the following parameters.

- Sampling frequency: 20 kHz.
- # of Samples: 10000.

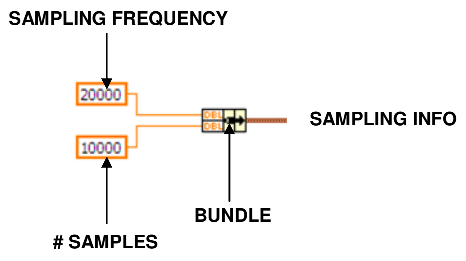


Figure 3. Building sampling info.

Now, in order to study the effect of modulation index, generate controls for the amplitudes of message and carrier signal. Calculate the modulation index from these two amplitudes, as mentioned previously and display it in the front panel. To examine the effect of carrier frequency with respect to message frequency, generate controls corresponding to these two parameters. Notice that, these parameters are inputs to the sine wave generators. The range of the message signal frequency is set to be from 10Hz to 40Hz and the range of carrier frequency is set from 10Hz to 500Hz.

The initial parameters to be chosen for the message signal are

- Frequency: 10 Hz
- Amplitude: 1

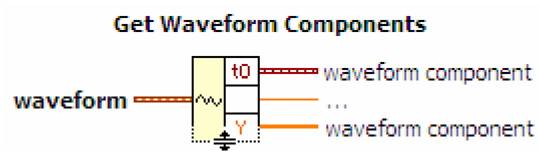
The initial parameters to be chosen for the carrier signal are

- Frequency: 200 Hz
- Amplitude: 1

Notice here, that the carrier frequency is much higher compared to the frequency of the message signal.

2.1.2 Get Waveform Components

In order to use the signal outputs from the sine wave generators, it may be required to convert the output signal into an array of double values. This is accomplished by the function ***Get Waveform Components***. This function can found in the functions palette (*Programming* >> *Waveform* >> *Get Wfm Components*).



2.1.3 Modulation

The modulation process is accomplished in two steps

- multiply the message signal to the carrier signal
- Add the carrier signal to the result.

In Figure 2, the modulated signal is provided to the ***Build Array*** function as one of the inputs, and the second input to this block is the message signal. This facilitates to view the input signal and modulated signal simultaneously. Now switch to the front panel and place a ***Waveform graph***. Switch to the block diagram and connect the output of Build array block to the waveform graph. Also compute the FFT of the two time domain signals and view their frequency responses on another ***Waveform graph*** in the front panel.

2.1.4 Demodulation

The task of the demodulation is left as an exercise for the students. As can be seen from the block diagram, only the inputs and the outputs of the demodulator are shown in the block diagram, as the underlying functionality has been grouped into a **subvi**.

Hint:

From the block diagram, it can be seen that the inputs to the demodulator are the carrier and the modulated signals. As described earlier, the demodulation is identical to the modulation process and it involves two steps.

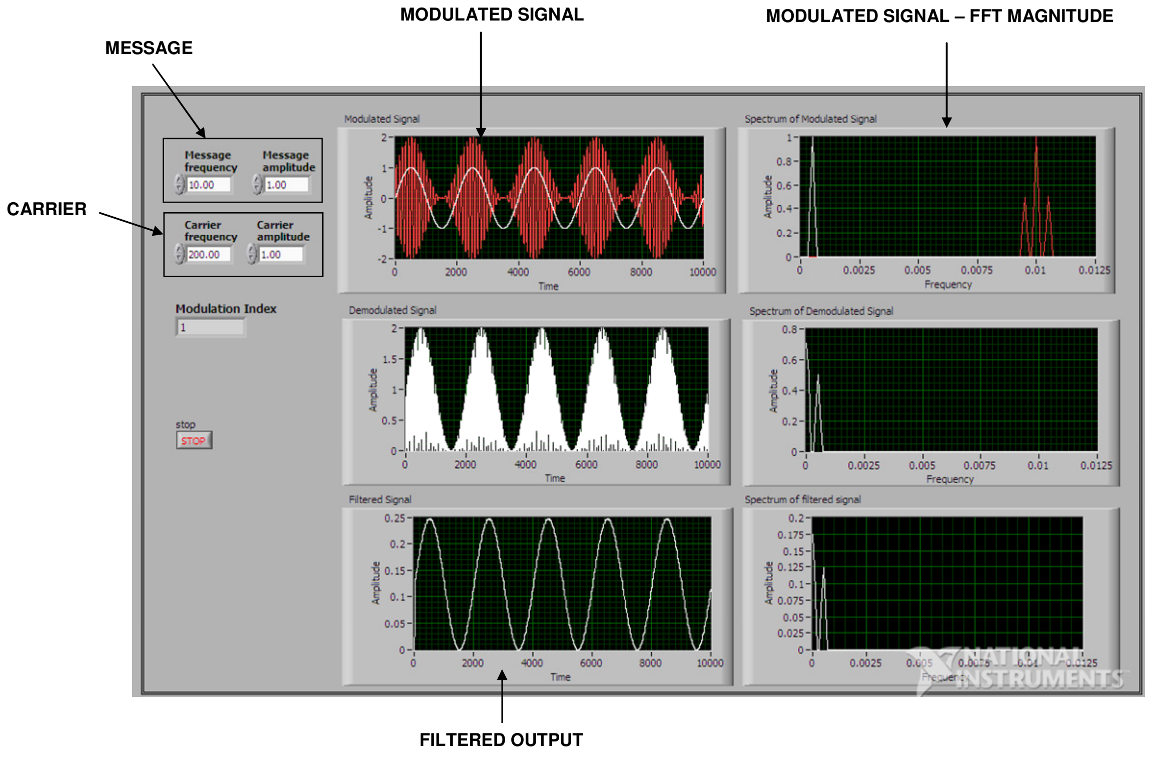
- Multiply the modulated signal to the carrier signal.

- Generate the time and frequency domain plots of the demodulated signal.

- Use a low-pass filter on the demodulated signal to remove the high frequency components generated due to the multiplication of the two signals

- Design a low-pass filter using the function ***FIR filter*** (*Signal Processing >> Filters >> Advanced FIR Filtering*). The filter coefficients are designed using the function ***FIR Windowed Coefficients***. The parameters that you can control are the sampling frequency, # taps and low cut-off frequency. Fix the sampling frequency to 20kHz. Now the number of taps defines the order of the FIR filter. So it is required to adjust two parameters (Taps and Low Cutoff) to get the perfect reconstruction of the message signal. As done earlier plot the filtered signal in time and frequency domains. That would help to tune the two parameters of the FIR Coefficients. Figure 4 shows the front panel of this exercise.

- Fix the number of taps to be 50. Choose a cut-off frequency such that the demodulated waveform is similar to the message signal.


MODULATED SIGNAL
MODULATED SIGNAL – FFT MAGNITUDE
MESSAGE
CARRIER
Modulated Signal
Spectrum of Modulated Signal
Demodulated Signal
Spectrum of Demodulated Signal
Filtered Signal
Spectrum of filtered signal
Amplitude
Time
Frequency
FILTERED OUTPUT

Message frequency 10.00
Message amplitude 1.00
Carrier frequency 200.00
Carrier amplitude 1.00
Modulation Index 1
stop STOP

Figure 4. Block diagram for Exercise 2.1

Now, examine the effect of different parameters of the sine waveform generators on the modulation-demodulation process.

First of all, fix the message frequency (10 Hz) and start changing the carrier frequency between its minimum and maximum value. What do you observe in the modulated signal (in time and frequency domains)?

Exercise 1 (40 pts)

1(a) Provide the block diagram and front panel snapshot with following parameters

Message signal

- Frequency: 10 Hz
- Amplitude: 1

Carrier signal

- Frequency: 200 Hz
- Amplitude: 2

1(b) The next step is to examine the effect of modulation index. To perform this task fix the amplitude of carrier signal at 2. Provide snapshots of the front panel with amplitude of the *message signal*

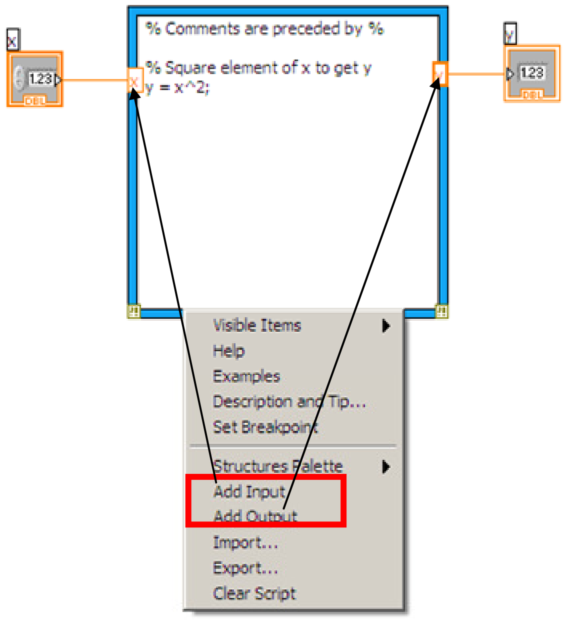
i) 2

ii) 3

What are the modulation indices ? Does changing the value of modulation index affect the quality of the reconstructed signal? (If yes, give reasons)

2.2. MathScript Node

LabVIEW graphical programming paradigm can be used with LabVIEW MathScript, a math-oriented textual programming language that is generally compatible with the widely used .m file script syntax. A ***MathScript Node*** can be placed in your block diagram by selecting it from *Programming>>Structures>>Mathscript Node*. In this exercise we will perform a simple example using MathScript Node.


% Comments are preceded by %

% Square element of x to get y
y = x^2;

Figure 5. Illustration of MathScript Node.

You can type the MathScript commands inside the blue structure or import an .m file directly. By right-clicking on the boundary, you can find options to create input and output nodes to the Mathscript node, which can in turn be used to communicate with other LabVIEW functions. In Figure 5, one input variable *x* and one output variable *y* have been created.

Create a new VI and place a Mathscript node in it. Add one input variable *x* and one output variable *y* as shown in Figure 6. Type the following code inside the MathScript node.

```
% Comments are preceded by %

% Input Vector A

A = [0 1 2 3 0];

% Square element of x to get y

y = A.^x;   % Notice that there is a dot after A

% Plot x and y

stem(x);

hold on;
```

```
stem(y,'r');
```

Pass a constant value to the input variable x (=2) and display the output variable y. Save and execute the VI. What do you observe?

2.3. Using an input speech file

In this exercise you will design a modulator-demodulator for the input speech file provided (*y.wav*). You can reuse most part of the block diagram from Exercise 2.1 except for the following changes. Figure 6. shows the block diagram for this exercise.

2.3.1 Message and Carrier Signals

The message signal is not a sinusoid but the input speech. You will make use of Mathscript nodes to read the .wav file and also playback the reconstructed speech. But you will use the same carrier signal of different frequency

Exercise 2 (60 pts)

2(a) Write a mathscript "*Mathscript node 1*" with following parameters

Input

- **Filename (String):** Get the path of the input file and convert it into a string.

Outputs

- **y (Dbl Array):** Contains the contents of the speech file.
- **fs (Dbl):** Sampling Frequency.
- **Am (Dbl):** Peak value of y (use the command **max(abs(y))**).
- **len (Dbl):** Number of samples in y (use the command **length(y)**);

[You can read a .wav file in MathScript using the command **[y,fs,nb] = wavread(Filename).** nb is the number of bits, but we do not use that in this exercise.]

2(b) Write a mathscript "*Mathscript node 2*" with following parameters to playback the reconstructed speech.

Input

- **y (Dbl Array):** Output from the demodulator.

- **fs (Dbl):** Sampling frequency.

In general, to playback speech you can use the command **sound(y,fs)**. In this exercise, use the command **sound(4*(y-0.5),fs)**. Here, 4 is an amplification factor and 0.5 is an offset.

2(c) Why is there the amplification factor and the offset? (Give reasons with equations if possible)

2(d) Provide the snapshot of the block diagram and the front panel with the above mathscripts for the following parameters.

Carrier signal

- Frequency: 2000 Hz
- Amplitude: 2
- low cutoff freq: 2500

[The **sampling info** to this sine waveform generator is obtained from the outputs of the **Mathscript node 1**.]

What is the value of the modulation index obtained?

2.3.2 Design of the low-pass filter

Try varying the # taps and the cut-off frequency, so that the reconstructed speech sounds like the original. If you hear a lot of distortion, you need to tune these two parameters. Fix the sampling frequency of this filter also to be fs.

2(e) Vary the low cut off frequency. At what low cut-off frequency do you observe distorted filtered signal ?

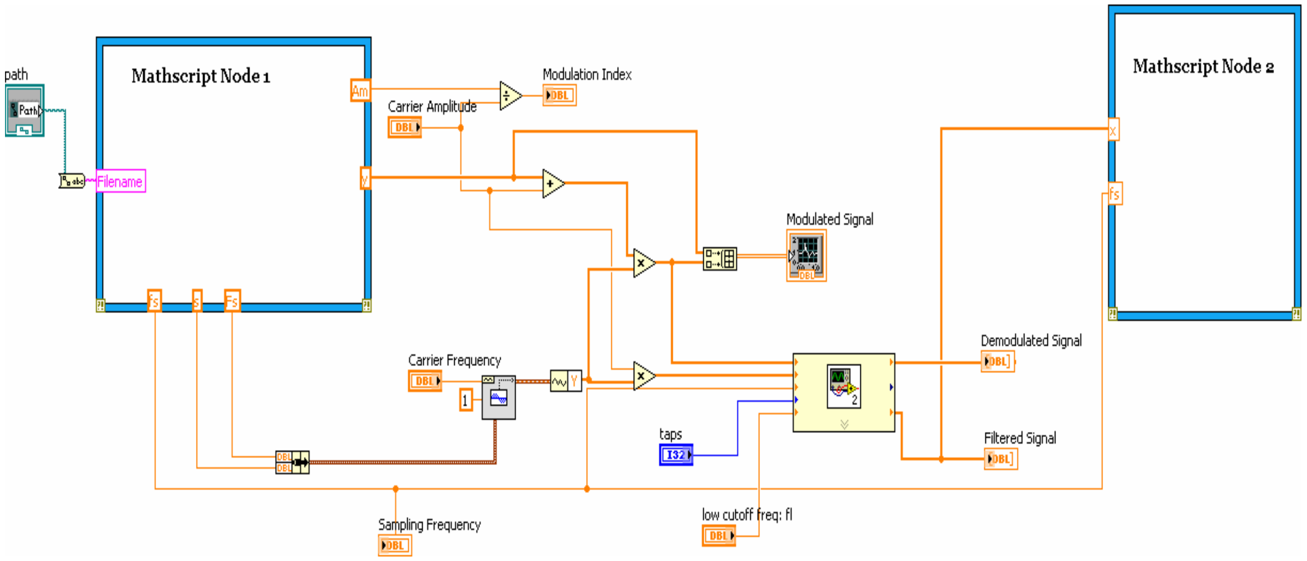


Figure 6. Block diagram with use of Mathscript.

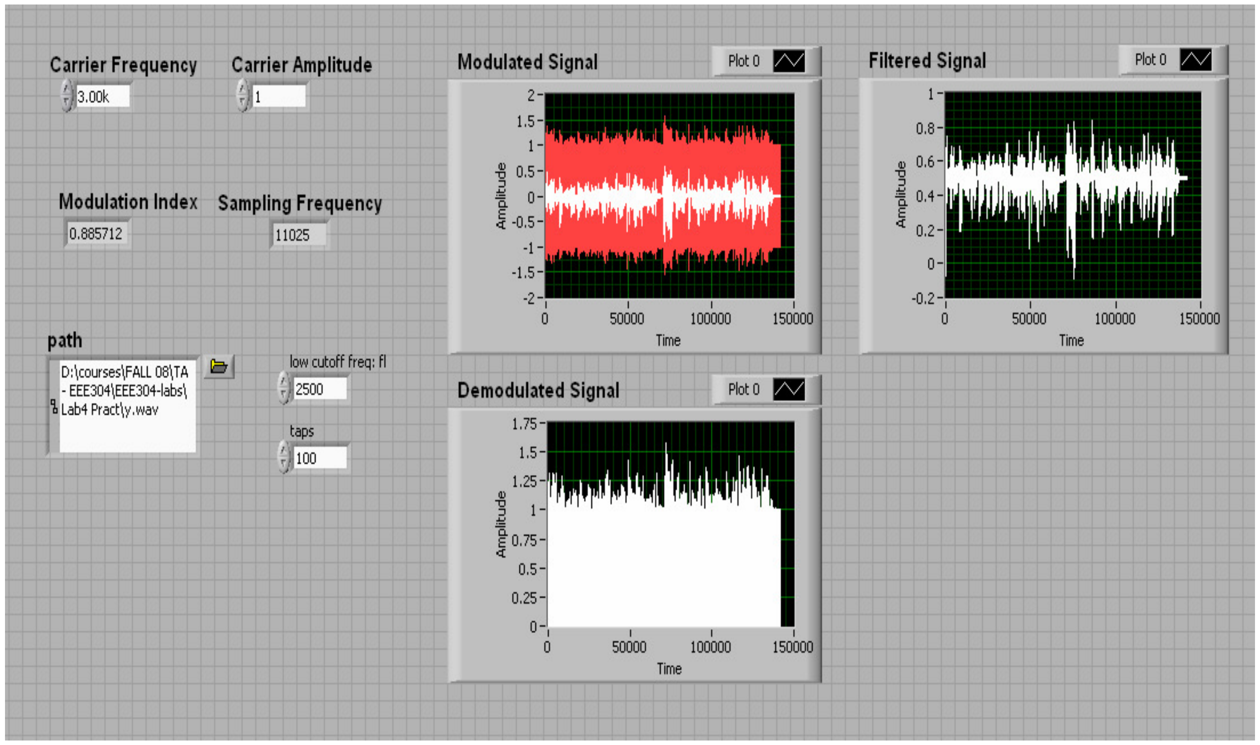


Figure 7. Front Panel for Exercise 2