AM MODULATION & DEMODULATION

EEE3092F: Lab 1

Prepared By: **Ronak Mehta - MHTRON001**

Lab Partner: **Maneno Mgwami – MGWMAN006**

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and to pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each significant contribution to, and quotation in, this essay from the work, or works, of other people has been acknowledged through citation and reference.
3. This work-piece is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

______________________________ 27th April 2019

Signature Date

AIM:

To investigate amplitude modulation and demodulation techniques, including frequency translation, the nature of sidebands and synchronous versus asynchronous demodulation.

INTRODUCTION:

Amplitude modulation and demodulation techniques will be implemented in this lab and be compared to simulations in Julia.

1. MODULATION:

1.1 Double Sideband Suppressed Carrier (DSB-SC)

- A modulating waveform was set up using a signal generator and it was a sine wave of frequency 1kHz with peak-to-peak voltage of 4V.
- The carrier wave was generated using the oscilloscope's generator and it was a sine wave of frequency 20kHz with peak-to-peak voltage of 4V. The following result was achieved:

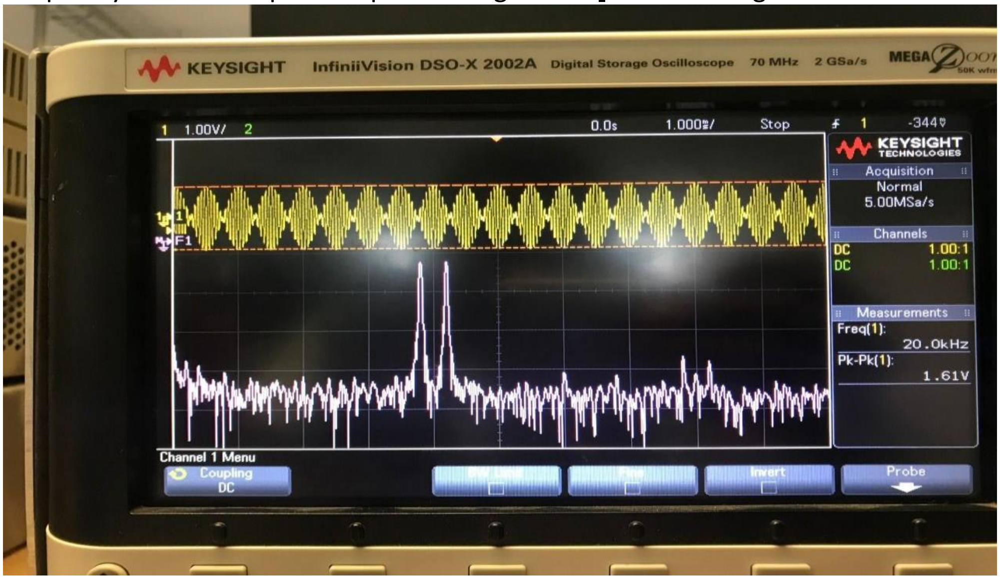

Figure 1: Oscilloscope output of the modulating waveform. Purple graph shows the fourier transform of the output

- Note that the multiplier introduced factor of 0.4, hence output goes 1.6V peak-to-peak.
- The following result was achieved from simulation:

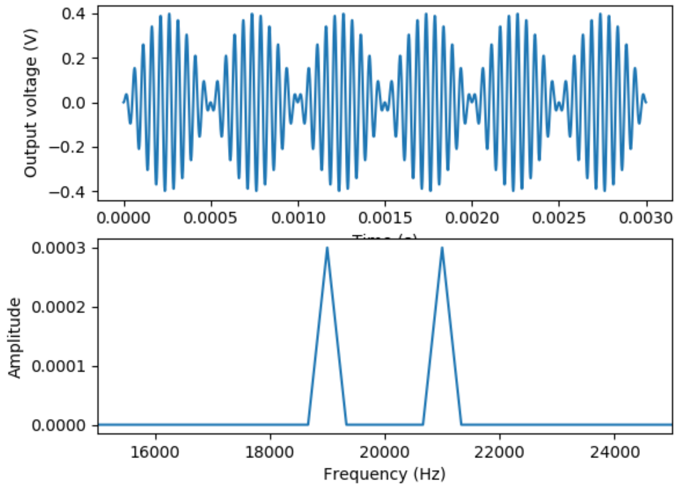

Figure 2: Simulated output of the modulated signal

- Notice how the waveforms are identical to the simulations.

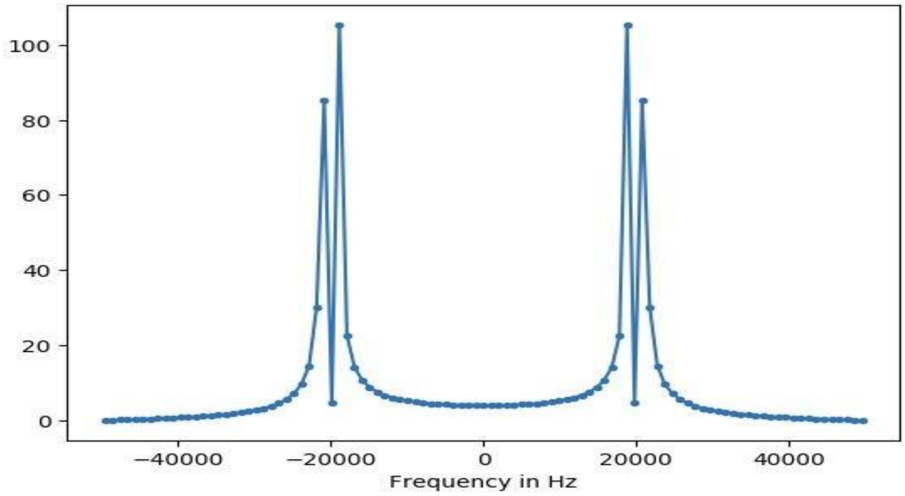

Figure 3: Frequency spectrum of DSB-SC signal

- The frequency of the modulating signal was then varied (changed to 5kHz). An increase in frequency causes an increase in the separation of the peak distance in the frequency domain of the modulating signal. The following outputs observed:

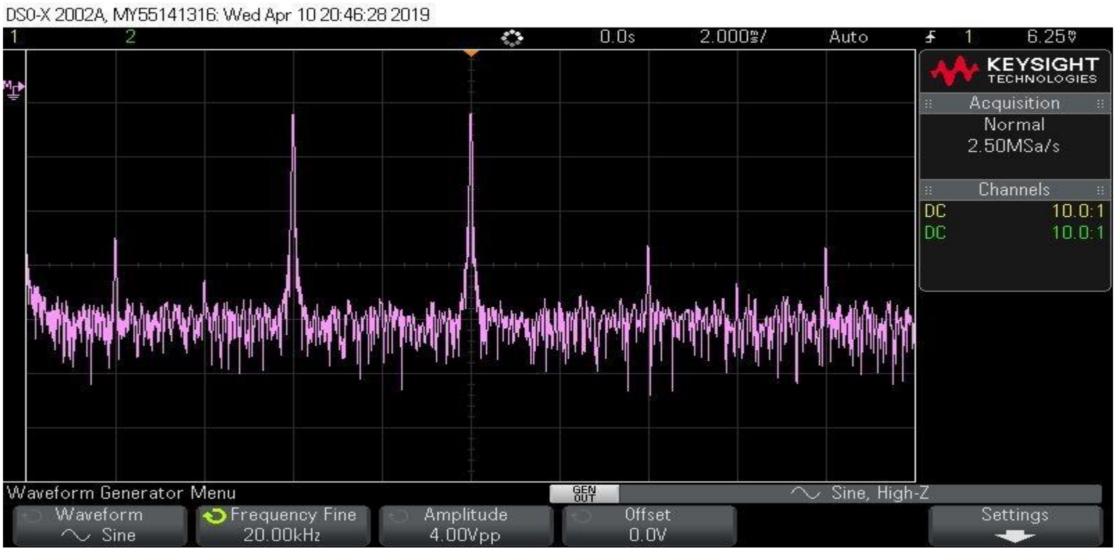

Figure 4: Oscilloscope output of the modulator output with modulating wave frequency increased

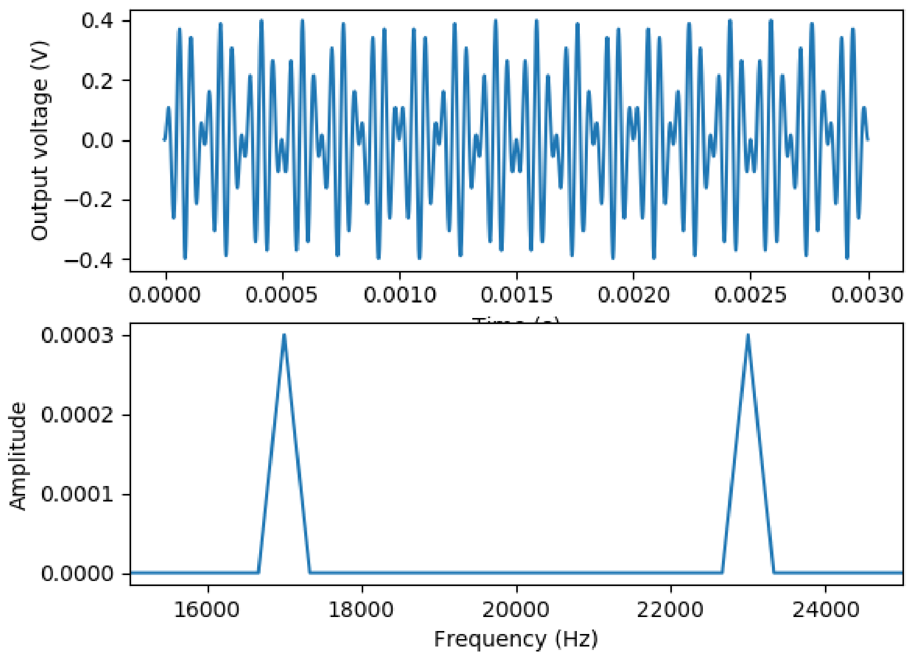

Figure 5: Simulated output of the modulator output with modulating wave frequency increased

- A square wave of 4V peak-to-peak with 20kHz frequency was then used as the modulating signal and the following modulator outputs observed:

Figure 6: Oscilloscope output of the modulating waveform when square wave used as modulating signal

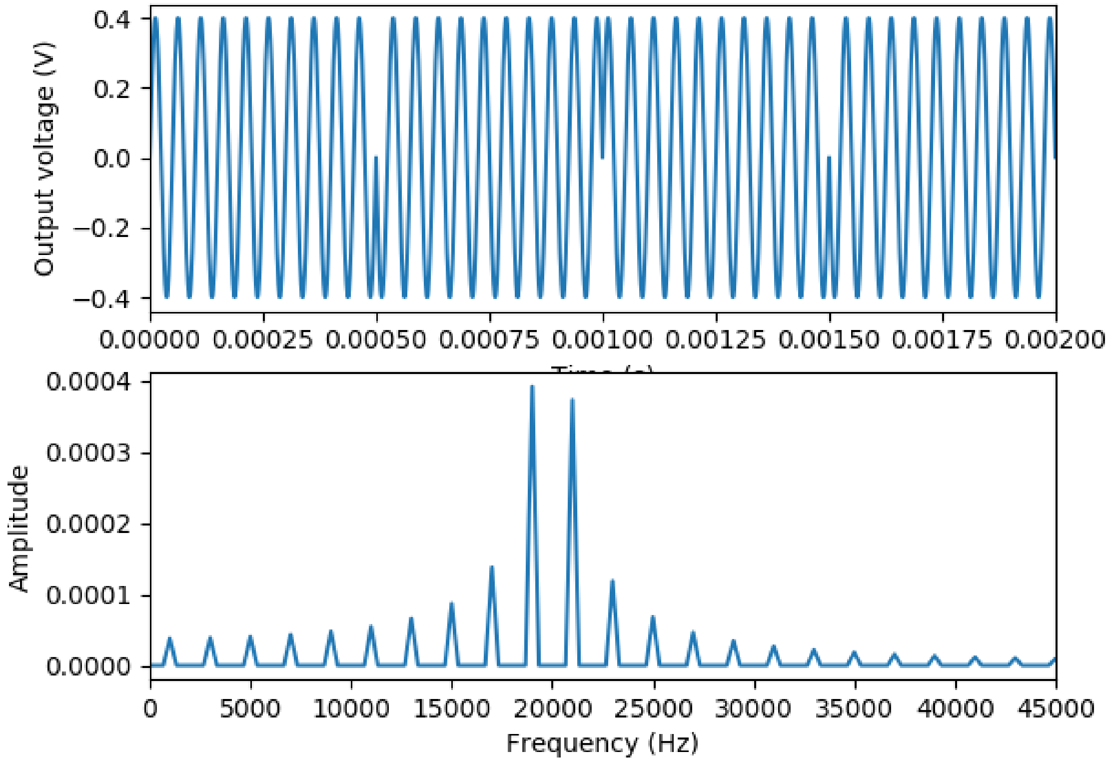

Figure 7: Simulated output of the modulating waveform when square wave used as modulating signal

- As seen, the spectrums and waveforms match. Also, there appears to be many spectral lines of finite width. This is expected since a square wave has fourier coefficients which bring about impulses(lines) at odd harmonics in the spectrum.

- Windowing causes spectral lines to have finite width because it reduces side lobe

1.2 Double sideband large carrier (DSB-LC)

The original modulating and carrier waves were used for this section. However, an offset was added to the modulating signal, and the resultant outputs analysed.

Note that modulation index (m) = $\frac{Envelope\ peak - Carrier\ amplitude}{Carrier\ amplitude}$

Case 1:

- Since our carrier and modulating signals each had an amplitude of 2V, an offset of greater than 2V was applied to get the modulation index to be less than 1.
- The following outputs are obtained for this case:

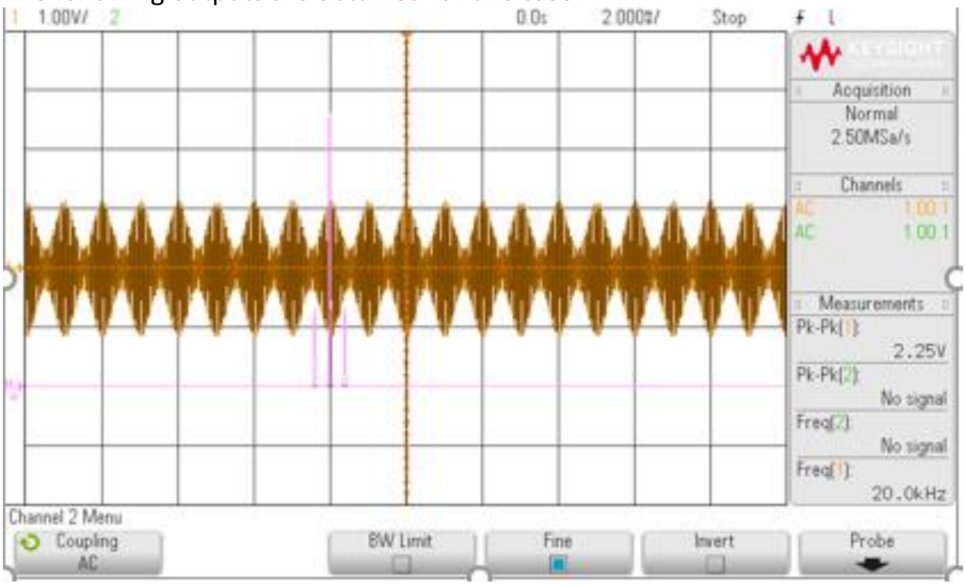

Figure 8: Oscilloscope output for the case m<1

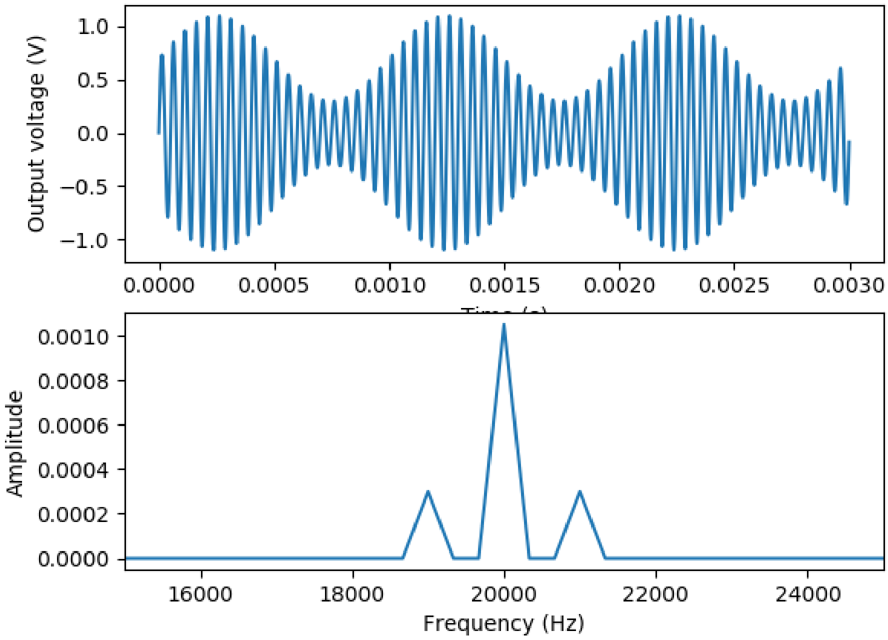

Figure 9: Simulation output for the case m<1

- The waveforms appear to match. Also, notice how the frequency spectrum has a line of finite width at the carrier frequency. This is because of the offset applied.

Case 2:

- An offset of 2V is applied to give the modulation index of exactly 1.
- The following results are then obtained:

Figure 10: Oscillator output for the case m=1

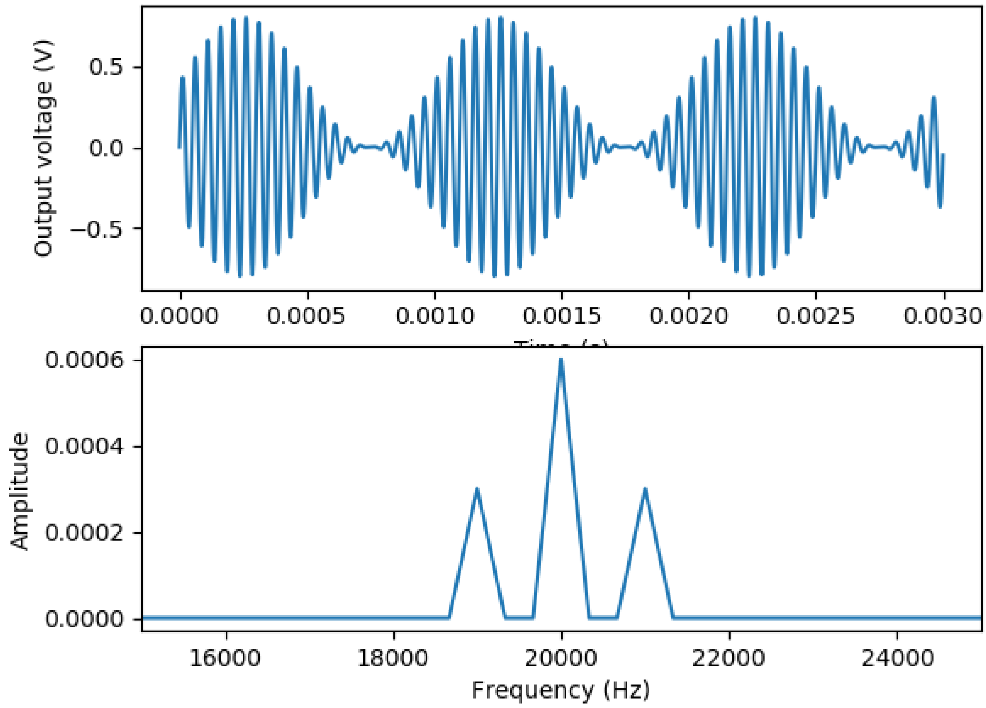

Figure 11: Simulation output for the case m=1

- The waveforms match one another. The modulator signal now appears to be zero for a very short duration of time.

Case 3:

- The offset is now brought to less than 2V to make the modulation index greater than 1.
- The following outputs are then observed:

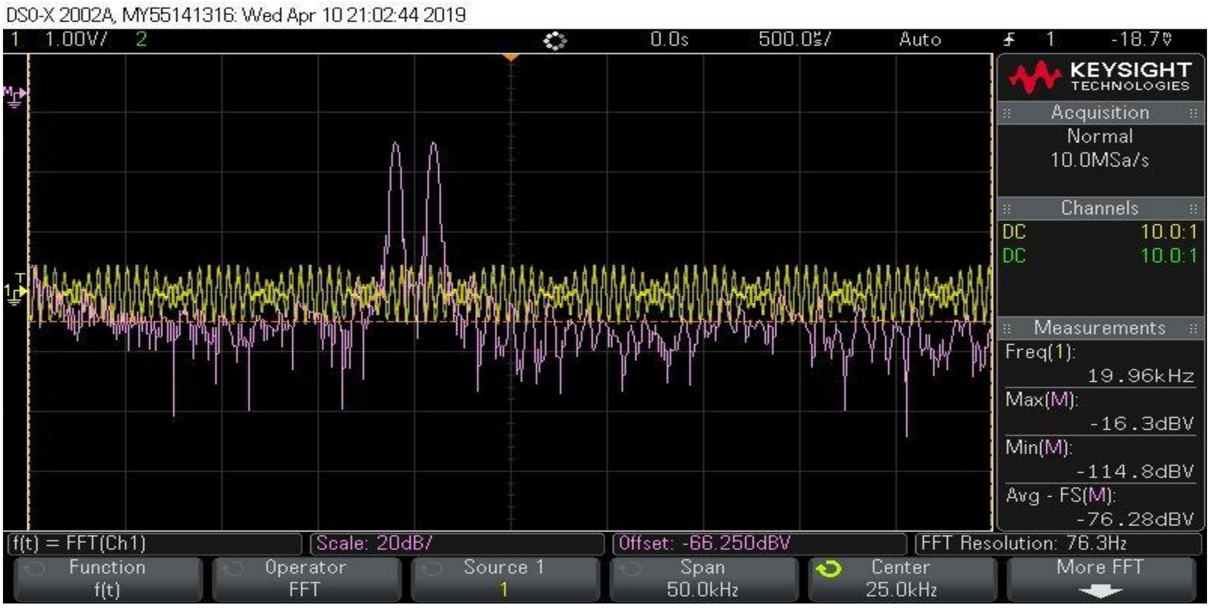

Figure 12: Oscilloscope output for the case m>1

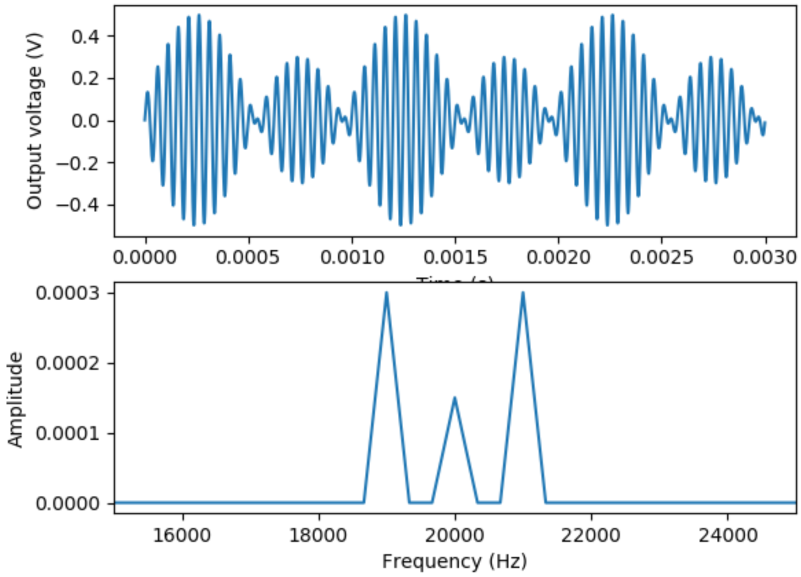

Figure 13: Simulated output for the case m>1

- The line at the carrier frequency appears to be less in amplitude then the two previous cases. This is because the offset is about 0.5V and hence, as we decrease it, we approach our case where the line at carrier frequency goes to zero in amplitude (Suppressed carrier case).

2. DEMODULATION

2.1 Synchronous Demodulation of DSB-SC

- This type of demodulation is done by multiplying the modulated signal by the original carrier and passing the product through a low-pass filter.
- A first order low pass passive filter was used with resistor value 1kΩ and capacitor value of 145nF, giving a cut-off frequency of about 1.kHz.
- Since it's a first order filter, the roll-off rate is approximately -20dB/decade. Hence, a carrier (at f=20kHz) would have an attenuation of approximately 26dB (i.e. 20log $(\frac{20k}{1k})$).
- The expected output is 28mV (56mV-pp).
- The following results are obtained:

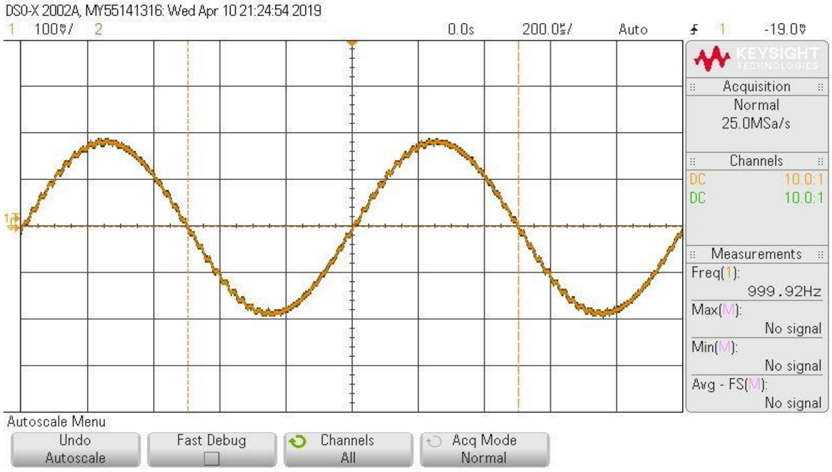

Figure 14: Demodulated signal output

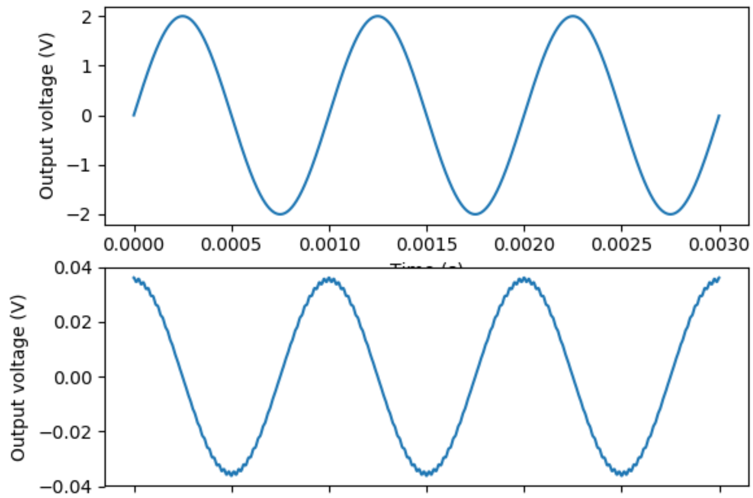

Figure 15: Simulated output of the original signal (top) and the final demodulated signal (bottom)

Investigating effects of phase error in the demodulation process

- Analysis shows that output of a low pass filter is 0.5*f(t)*cosθ
- The phase was varied in the lab using a phase shifter. The following results were obtained:

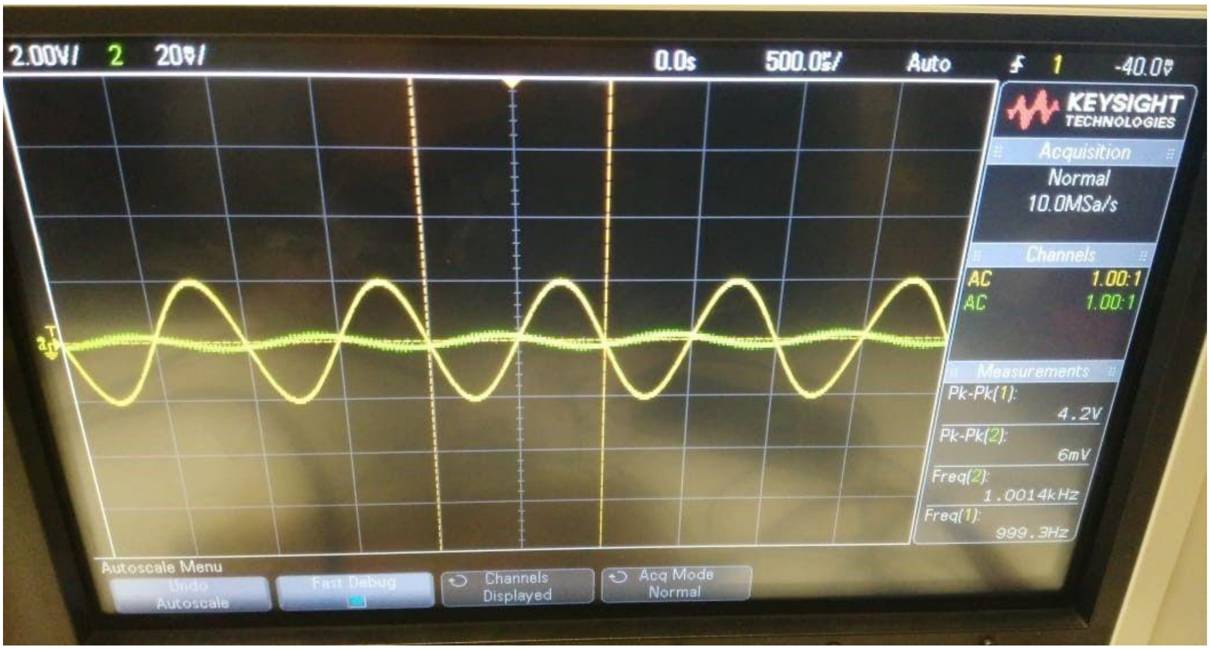

Figure 16: Oscillator output for original signal (yellow) and demodulated signal (green) when a carrier had a phase shift of 90° at demodulation

- We see that the amplitude of the output decreases significantly when a phase error has been introduced. It goes from 56mV-pp to 6mV-pp

2.2 Asynchronous Detection (Envelope Detection)

- This is normally used for DSB-LC.
- The envelope detector is implemented by adding a parallel RC combination to the output of a half-wave diode rectifier. RC values of 2.2kΩ and 100nF were used in this lab.
- The output from the envelope detector is then fed to a low pass filter.
- The following are the outputs achieved:

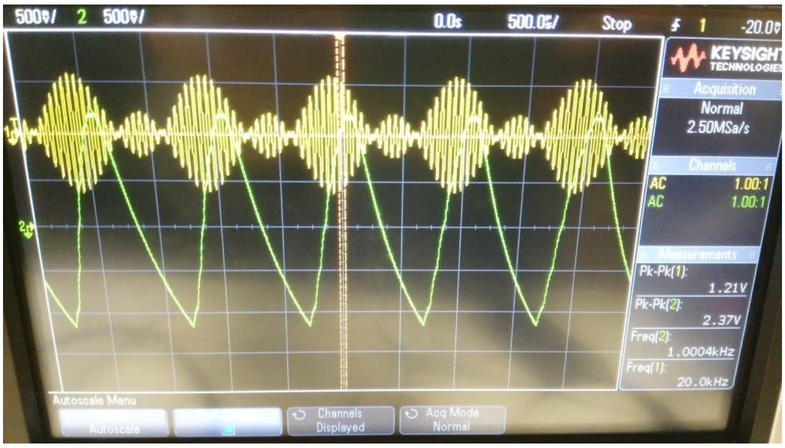

Figure 17: Oscilloscope output for the signal after the low pass filter (green)

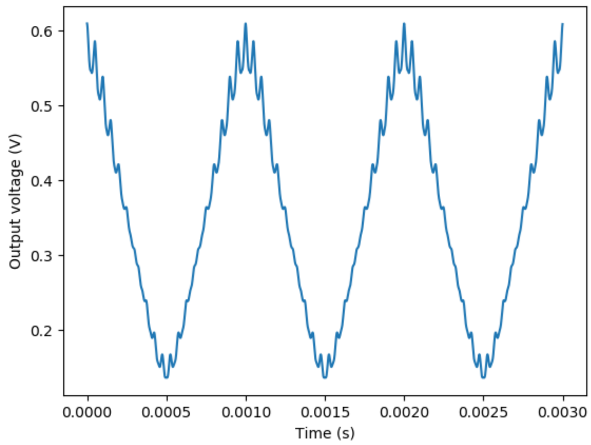

Figure 18: Simulated output for the signal after the low pass filter

- Notice how there exists a DC offset i.e. the average value is no longer 0. This is because we are dealing with large carrier signals that contain offset.
- To get rid of this dc offset, we pass it through a high pass filter, so that the low frequency component (in this case 0Hz) gets eliminated.
- The following is then the final output:

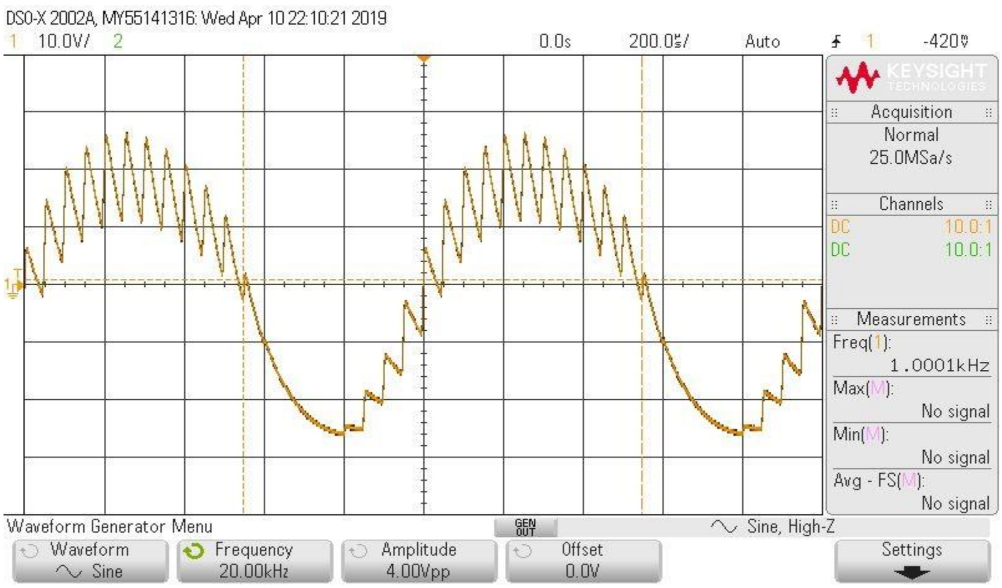

Figure 19: Oscillator output for the final signal

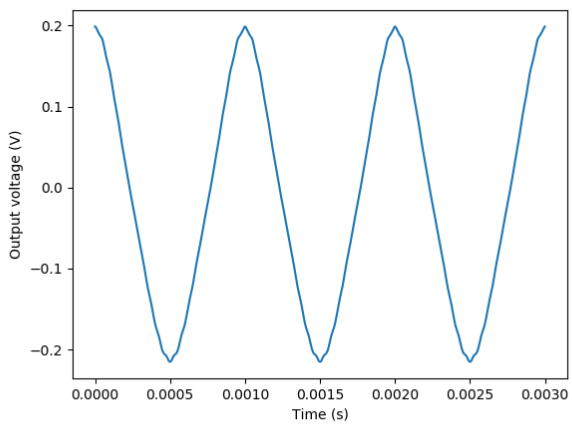

Figure 20: Simulated output for the final signal

- Values for the LPF and HPF used for the lab are recorded below:
  <u>For LPF:</u>
  Cut-off frequency = 2kHz
  R_{LP} = 10KΩ
  C_{LP} = 8.1nF

  <u>For HPF:</u>
  Cut-off frequency = 500Hz
  R_{HP} = 100KΩ
  C_{HP} = 3.3nF

APPENDIX

Julia Code

```
using PyPlot
using Pkg

Pkg.add("FFTW) using FFTW

#Double Sideband Suppressed Carrier
#Defining the time range
t = np.arange(0,0.003, 0.000001)

#Defining the modulating signals
f = 2*np.sin(2*np.pi*1000*t)            #modulating sine wave signal with 1KHz freq
f2 = 2*signal.square(2*np.pi*1000*t)        #modulating square wave signal with KHz freq

#Defining the carrier signal
carrier = 2*np.sin(2*np.pi*20000*t)

#Defining the output of the modulator together with its fourier transform
modulating_output = 0.1*f*carrier      #Factor of 0.1 because of the multiplier used in lab
MO = np.fft.fft(modulating_output)
MOshift = np.fft.fftshift(abs(MO))

#Defining the frequency axis
#Since there are 3001 samples and we're using fftshift, centre point becomes 0
freq_left = -(1500/(3000*0.000001))
freq_right = (1500/(3000*0.000001))
increment = (1000/3)
freq = np.arange(freq_left, freq_right, increment)

#Plot the carrier and modulating signals
#plt.subplot(2,1,1)
#plt.plot(t, f, t, carrier)
#plt.xlabel('Time (s)')
#plt.ylabel('Output voltage (V)')

#Plot the modulator output signal
plt.subplot(2, 1, 1)
plt.xlim(0, 0.002)
plt.plot(t, modulating_output)
plt.xlabel('Time (s)')
plt.ylabel('Output voltage (V)')

#Plot the fft of modulator output signal and zoom in on the positive side of freq
plt.subplot(2,1,2)
plt.xlim(0, 45000)
plt.plot(freq, 0.000001*MOshift)    #0.000001 to account for scaling in the domain
plt.xlabel('Frequency (Hz)')
plt.ylabel('Amplitude')
#Double Sideband Large Carrier
```

```
#Setting the offset
k = 1     #Adjust to get various m values

#Defining the output of the modulator and its fft
modulating_output2 = 0.1*(f+k)*carrier
MO2 = np.fft.fft(modulating_output2)
MOshift2 = np.fft.fftshift(abs(MO2))

#Plot the modulator output signal with its fourier transform
plt.subplot(2, 1, 1)
plt.plot(t, modulating_output2)
plt.xlabel('Time (s)')
plt.ylabel('Output voltage (V)')
plt.subplot(2,1,2)
plt.xlim(15000, 25000)
plt.plot(freq, 0.000001*MOshift2)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Amplitude')

#Synchronous Demodulation of DSB-SC
%matplotlib notebook

#Defining the demodulator outut and finding its fft
demodulating_output = 0.707*0.1*0.5*modulating_output*carrier     #Factors of approximately 0.5
and 0.707 introduced by filter
DO = np.fft.fft(demodulating_output)

#Defining the LPF using a rect function and finding its fft
y = np.zeros(len(t))
y[0:455] = 1
Y = np.fft.fft(y)

#Defining the output of the LPF in frequency and time domain
OUTPUT = DO*Y
output = np.fft.ifft(abs(OUTPUT))
output_scaled = output/(40*np.pi)

#Plotting outputs
plt.subplot(1,1,1)
plt.plot(t, demodulating_output)
plt.xlabel('Time (s)')
plt.ylabel('Output voltage (V)')
plt.subplot(2,1,2)
plt.plot(t, output_scaled)
plt.xlabel('Time (s)')
plt.ylabel('Output voltage (V)')
plt.show()

#Synchronous Demodulation of DSB-SC with phase error
%matplotlib notebook
```

```
#Defining the carrier with a certain phase
carrier2 = 2*np.sin((2*np.pi*20000*t))

#Defining the modulator output and its fft
demodulating_output2 = 0.1*modulating_output*carrier2
DO2 = np.fft.fft(demodulating_output2)

#Defining the output from LPF in time and frequency
OUTPUT2 = DO2*Y
output2 = 0.5*np.fft.ifft(abs(OUTPUT2))
output2_scaled = output2/(40*np.pi)

#plotting outputs
plt.subplot(2,1,1)
plt.plot(t, f)
plt.xlabel('Time (s)')
plt.ylabel('Output voltage (V)')
plt.subplot(2,1,2)
plt.plot(t, output2_scaled)
plt.xlabel('Time (s)')
plt.ylabel('Output voltage (V)')
plt.show()

#Asynchronous Detection (Envelope detection) of DSB-LC
%matplotlib notebook

#Reectifier output in time and frequency
rectified_modulating_output2 = [0 if i < 0 else i for i in modulating_output2]
RECTIFIED_MODULATING_OUTPUT2 = np.fft.fft(rectified_modulating_output2)

#Defining a smooth function to work as a capacitor
#def smooth(y, box_pts):
   #box = np.ones(box_pts)/box_pts
   #y_smooth = np.convolve(y, box, mode='same')
   #return y_smooth

#Envelope output in time and frequency
#envelope_output = smooth(rectified_modulating_output2, 500)
#ENVELOPE_OUTPUT = np.fft.fft(envelope_output)

#Defining the LPF using a rect function and finding its fft
y2 = np.zeros(len(t))
y2[0:490] = 1
Y2 = np.fft.fft(y2)

#Finding output from the LPF
#OUTPUT3 = ENVELOPE_OUTPUT * Y2
#output3 = np.fft.ifft(abs(OUTPUT3))

LPF_OUT = RECTIFIED_MODULATING_OUTPUT2 * Y2
lpf_out = np.fft.ifft(abs(LPF_OUT))
lpf_out_scaled = lpf_out/(40*np.pi)
```

```
#Defining the HPF using a rect function and finding its fft
y3 = np.ones(len(t))
y3[0:1800] = 0
Y3 = np.fft.fft(y3)

#Defining the final output and finding its fft
OUTPUT3 = LPF_OUT * Y3
output3 = np.fft.ifft(abs(OUTPUT4))
output3_scaled = (output3/(1960*np.pi) - 8)/2

#Plotting outputs
#plt.subplot(1,1,1)
#plt.plot(t, lpf_out_scaled)
#plt.xlabel('Time (s)')
#plt.ylabel('Output voltage (V)')
plt.subplot(1,1,1)
plt.plot(t, rectified_modulating_output2)
plt.xlabel('Time (s)')
plt.ylabel('Output voltage (V)')
```