SSB MODULATION AND QUADRATURE MULTIPLEXING

EEE3092F: Lab 2

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.

______________________________ 16th May 2019

Signature Date

AIM:

To investigate:
1) Single side band modulation (SSB-SC) and demodulation
2) Quadrature Multiplexing and demultiplexing

INTRODUCTION:

SSB-SC modulation and demodulation techniques will be implemented in this lab alongside quadrature multiplexing and demultiplexing and be compared to simulations made in Julia.

1. SSB-SC MODULATION AND DEMODULATION:

Method:

- 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.
- Another input signal was set up with a 90^o phase shift with the first set of input signals. This was achieved by connecting the two signals to the two scope channels and using the 'Phase measurement' function to set the two signals 90^o apart.
- Next, a summing circuit was constructed so that an SSB signal is generated by adding the two carrier and modulated waveforms.
- A potentiometer is added to enable to 'fine tune' the addition of the co-function signal component [f^(t) sin(ω_ct)] to obtain an SSB-SC signal.
- The expression of this SSB-SC waveform is: $\varphi_{SSB\mp}$ (t) = [f(t) cos(ω_ct)] ± [f^(t) sin(ω_ct)]

- To demodulate this signal, we first need to multiply the SSB-SC signal by the original carrier cos(ω_ct)
- This will result in the expression:
  φ_{SSB+}(t)*cos(ω_ct) = f (t)cos(ω_ct) – f^(t)*sin(ω_ct)*cos(ω_ct)
  = 0.5 f(t) + 0.5f(t)cos(2ω_ct) – 0.5f^(t)sin(2ω_ct)

- Passing this demodulated signal through a low pass filter will thus recover f(t)
- The cutoff frequency was chosen to be 1.6kHz (greater than 1kHz) and Resistor and capacitor were calculated using the equation: f_{3db}= 1/2πRC
- Values for the LPF used for the lab are recorded below:
  For LPF:
  Cut-off frequency = 1.6kHz
  R_{LP} = 1KΩ
  C_{LP} = 100nF

- These experimental outputs where then compared with the simulated waveforms from Julia
- Shown below is a summary of the above-mentioned method

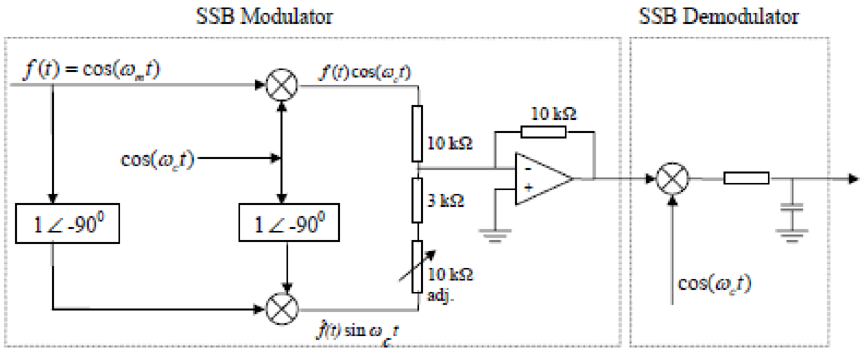

Figure 1: Schematic of SSB-SC modulator and demodulator

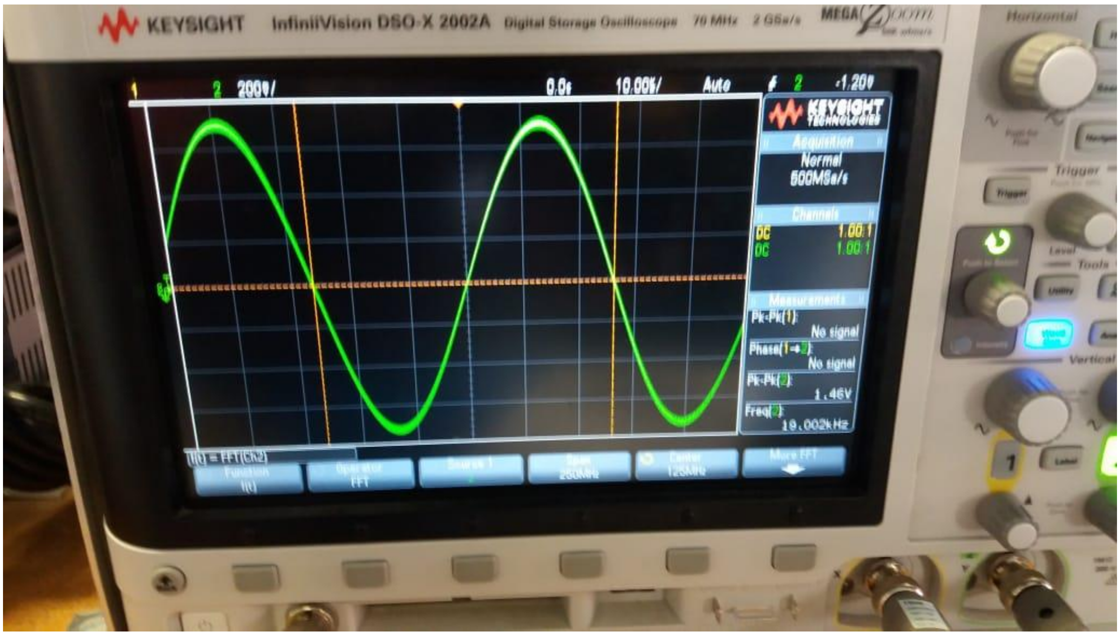

Figure 2: Oscilloscope output in time domain of SSB-SC waveform

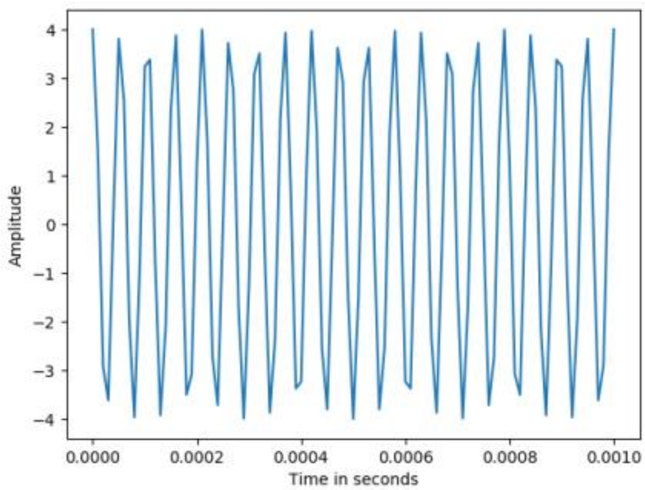

Figure 3: Simulated output in time domain of SSB-SC waveform

As expected we see a similar waveform similar sinusoidal waveform structure in lab as obtained in the simulation. This makes sense as the SSB-SC waveform in time domain should have been a combination of the two inputs summed up.

Figure 4: Oscilloscope output in frequency domain of SSB-SC waveform

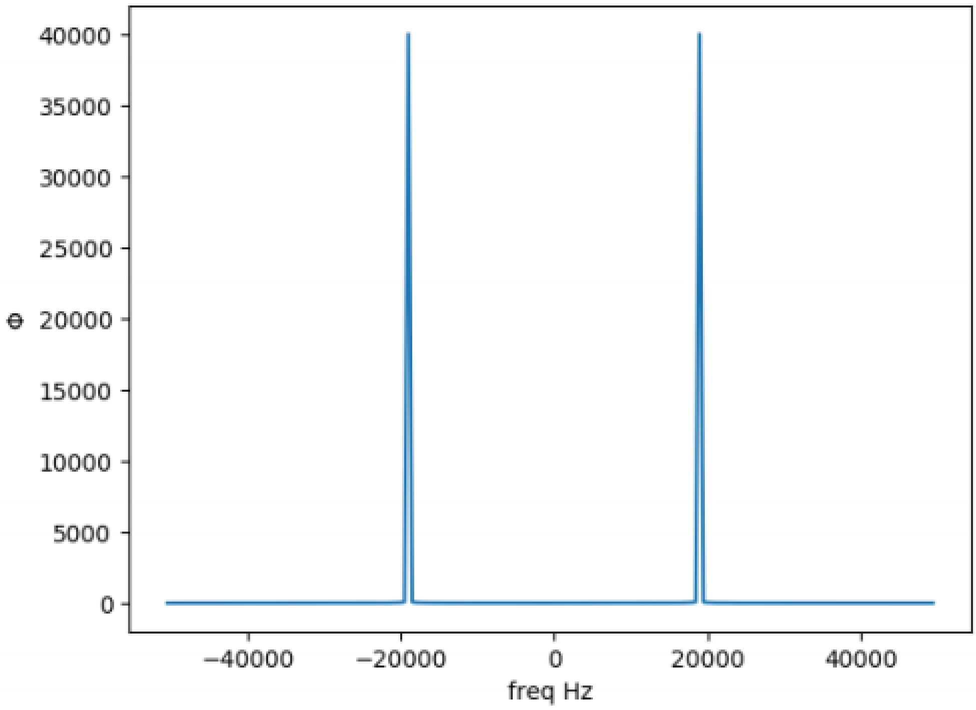

Figure 5: Simulated output in frequency domain of SSB-SC waveform

In the frequency domain of the SSB-SC waveform we again see similarities between the oscilloscope diagram and the simulation output. In the oscilloscope we only see the positive half of the output which is at 19kHz which matches exactly with the simulated output which also has a peak at 19kHz; although the simulation shows another peak at -19kHz which can't be visible in an oscilloscope as it only displays positive ffts.
The oscilloscope waveform has some disturbances which causes it to have a non-smooth output.

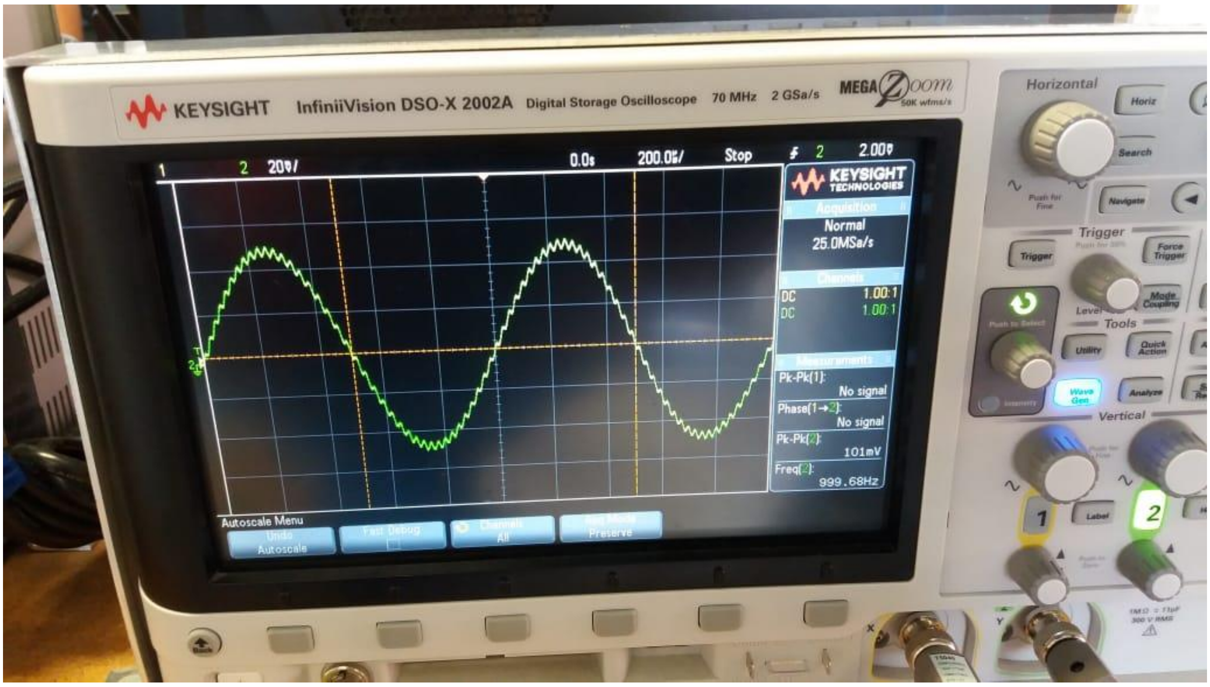

Figure 6: Oscilloscope output in time domain of demodulated output waveform

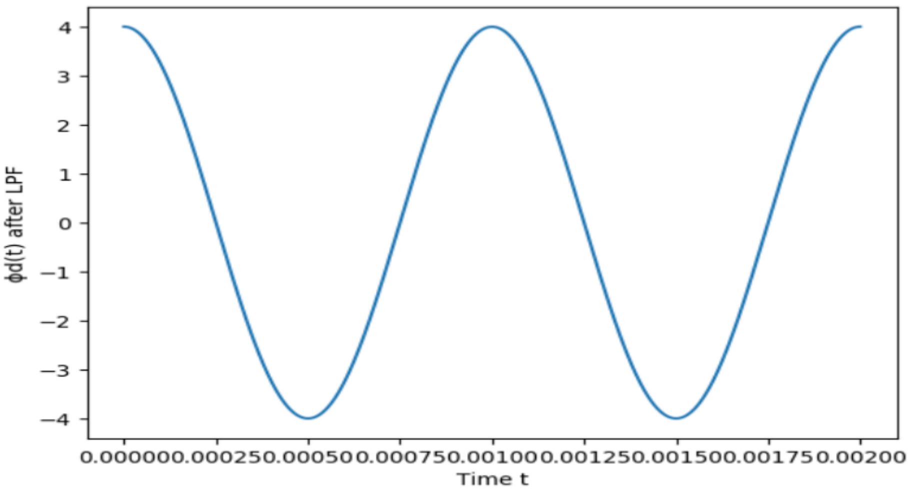

Figure 7: Simulated output in time domain of demodulated output waveform

The oscilloscope output of time domain for the demodulated signal after the addition of LPF is sort of similar to that when simulated in Julia apart from that in simulation the output is very smooth whereas in the oscilloscope, the output is not smooth due to presence of noise.

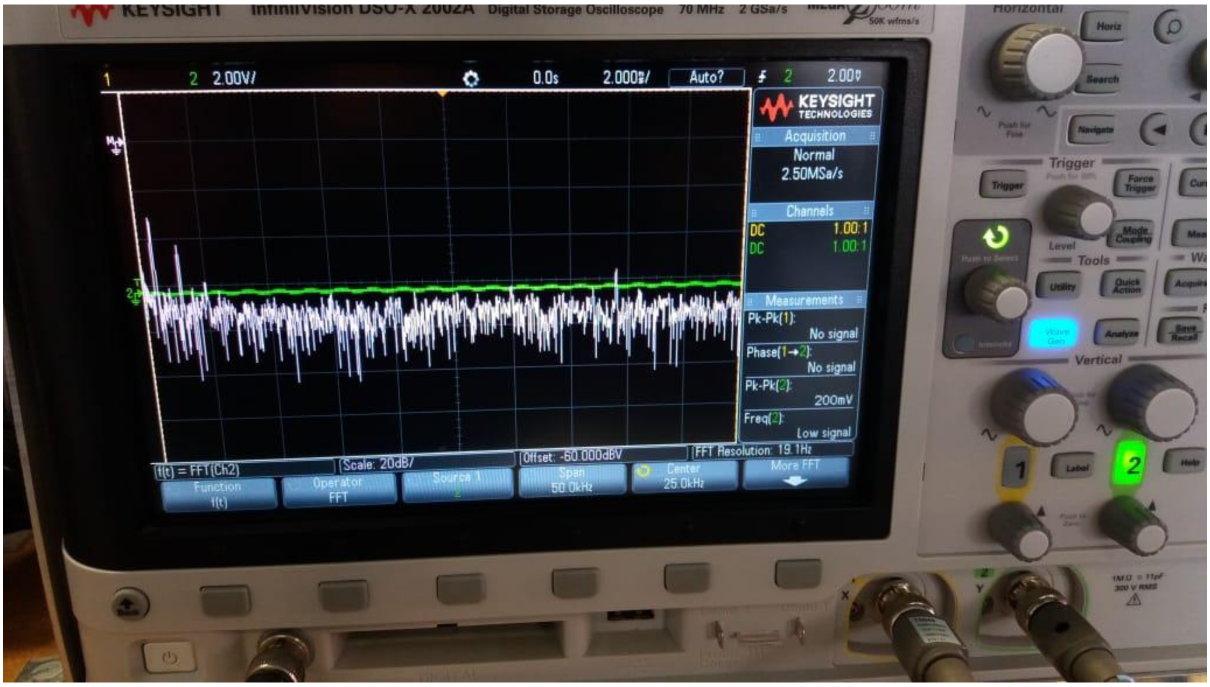

Figure 8: Oscilloscope output in frequency domain of demodulated output waveform

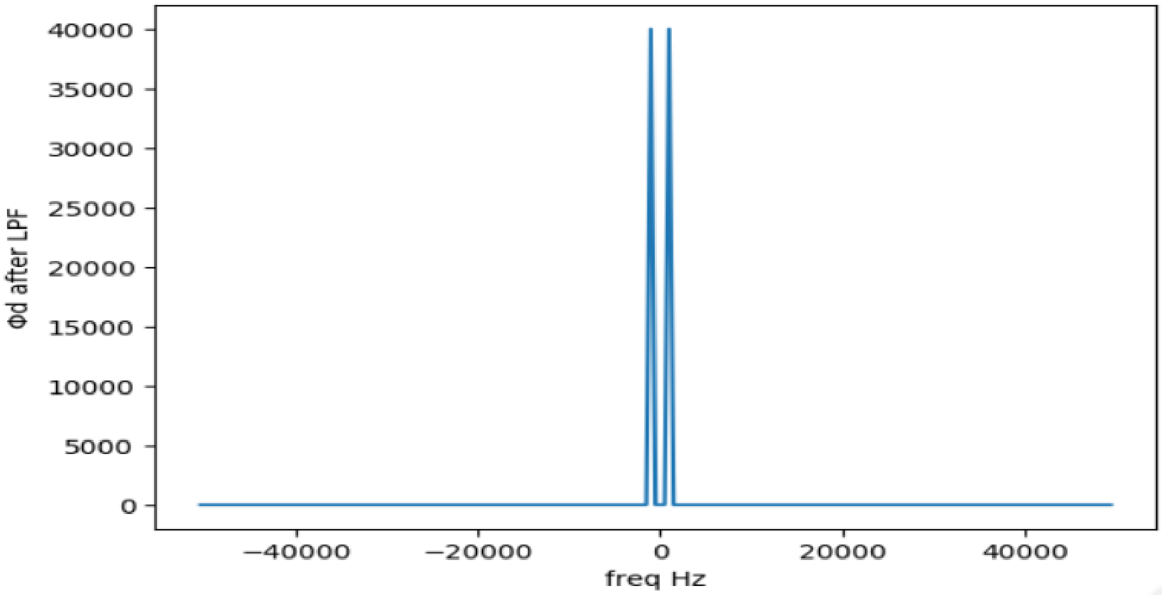

Figure 9: Simulated output in frequency domain of demodulated output waveform

Again, both these outputs match each other apart from the negative component in the simulated version which is not visible in the oscilloscope.
As expected, the frequency is also just above 1kHz.

2. QUADRATURE MULTIPLEXING AND DEMULTIPLEXING:

Method:

- This method demonstrates how two signals can be transmitted over the same medium using a single carrier frequency
- We will require three signal generators and four on-board modules as multipliers
- The first input signal f1(t) was a 50Hz modulating signal with peak-to-peak voltage of 4V
- The second input signal f2(t) was a 1kHz modulating signal 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
- To obtain a sin(ω_ct) we passed the cos(ω_ct) carrier waveform through a 90^o phase shifter
- f1(t) was multiplied with cos(ω_ct) and f2(t) was multiplied with sin(ω_ct). The two resulting waveforms were then passed through the same summing circuit from Experiment 1 (SSB-SC modulation)
- For demultiplexing, we passed the output of the previous waveform to two multipliers. One of the multipliers was similar to the carrier wave while the other multiplier was shifted 90^o out of phase to the first one
- These two waveforms were then each passed through a low-pass filter designed to only pass the modulating signal but suppress the carrier frequency

- Filter calculations using the equation: $f_{3db} = 1/2\pi RC$
  One LPF filter was centred above 50Hz to retrieve f1(t):
  Cut-off frequency = 72Hz
  R_{LP} = 100Ω
  C_{LP} = 22nF

  The other LPF filter was centred above 1kHz to retrieve f2(t):
  Cut-off frequency = 1.6kHz
  R_{LP} = 1kΩ
  C_{LP} = 0.1uF

- The two demultiplexed output signals obtained were half in magnitude of the respective original input signal. i.e. 0.5f1(t) and 0.5f2(t)
- Shown below is a summary of the above-mentioned method.

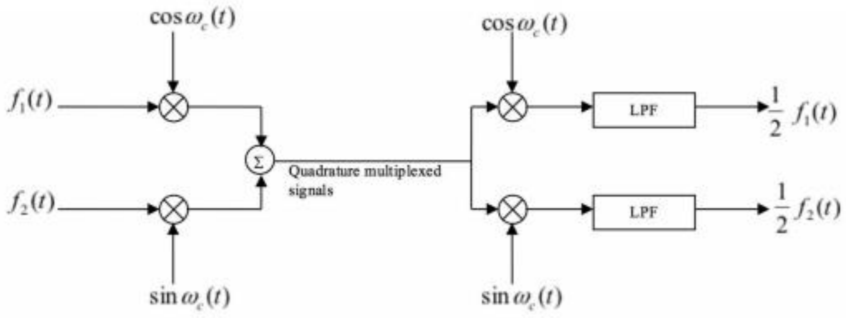

Figure 10: Block diagram of quadrature multiplexing and demultiplexing

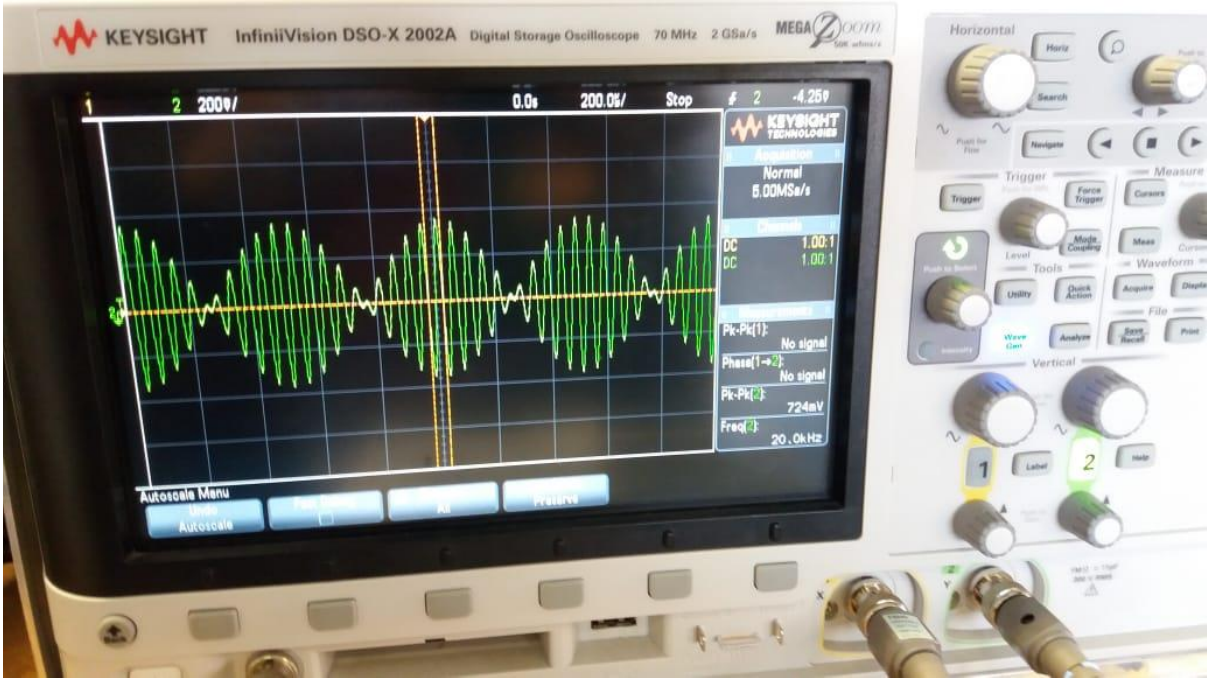

Figure 11: Oscilloscope output in time domain of summing signal

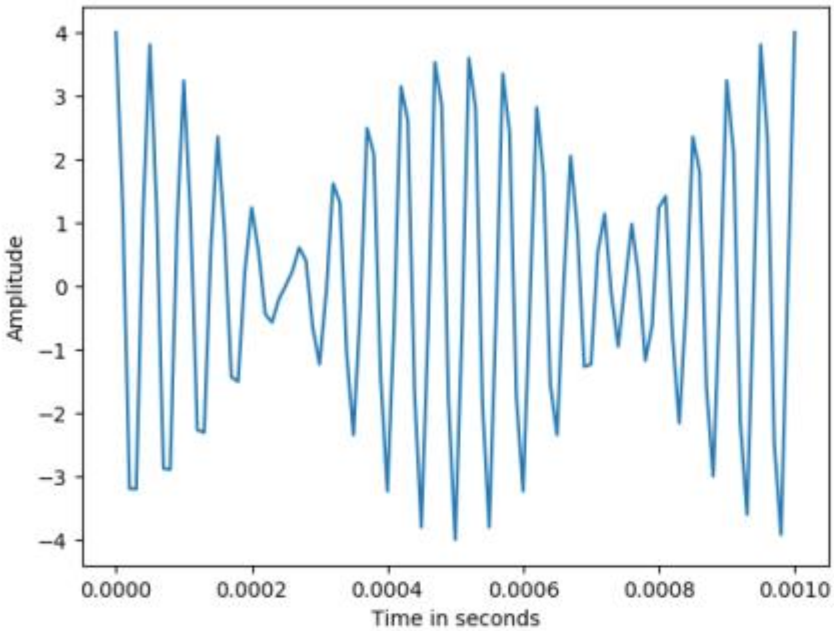

Figure 12: Simulated output in time domain of summing signal

In time domain, both the oscilloscope output and the simulated output look similar to each other as expected with a summing signal.

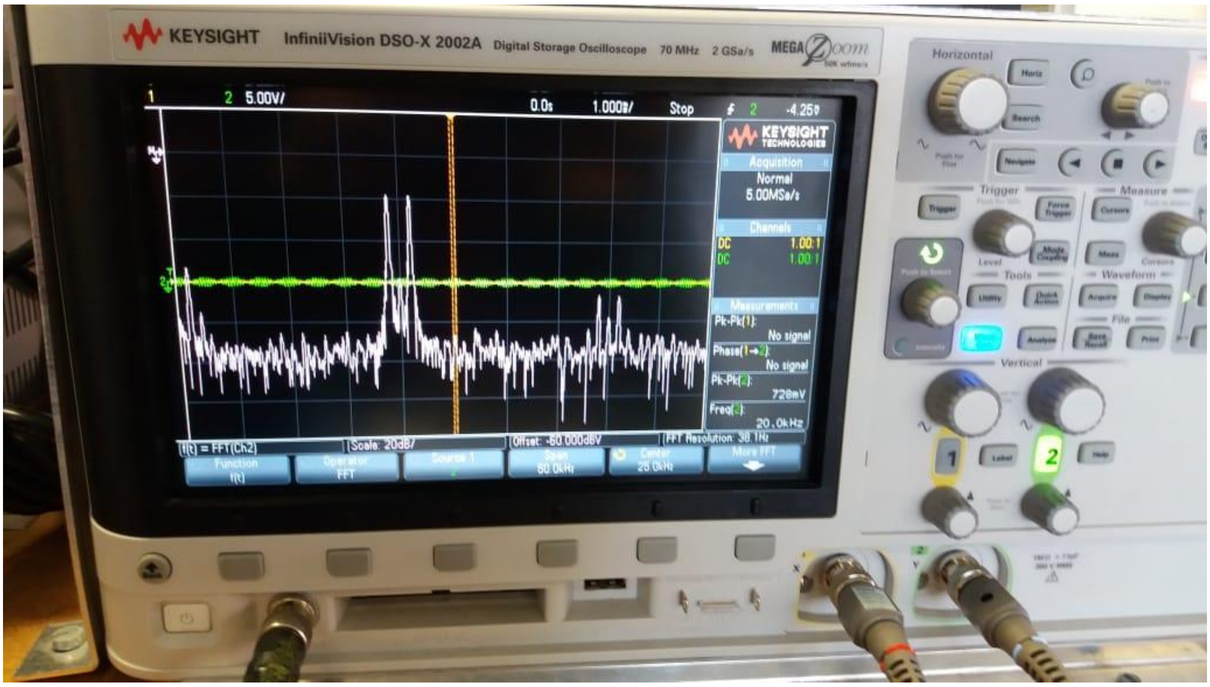

Figure 13: Oscilloscope output in frequency domain of summing signal

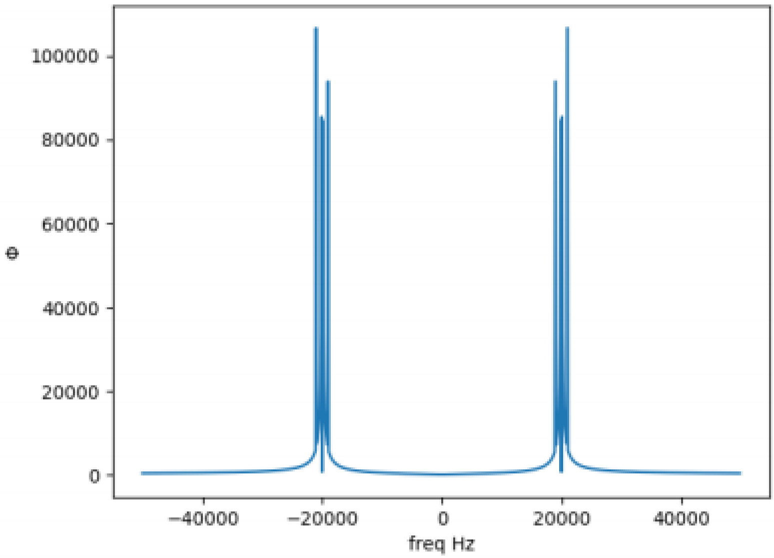

Figure 14: Simulated output in frequency domain of summing signal

In the frequency domain of the summing waveform we again see similarities between the oscilloscope diagram and the simulation output. In the oscilloscope we only see the positive half of the output which matches exactly with the simulated output which is also centred at 20kHz; although the simulation shows another peak centred at -20kHz which can't be visible in an oscilloscope as it only displays positive ffts.
The oscilloscope waveform has some disturbances which causes it to have a non-smooth output.

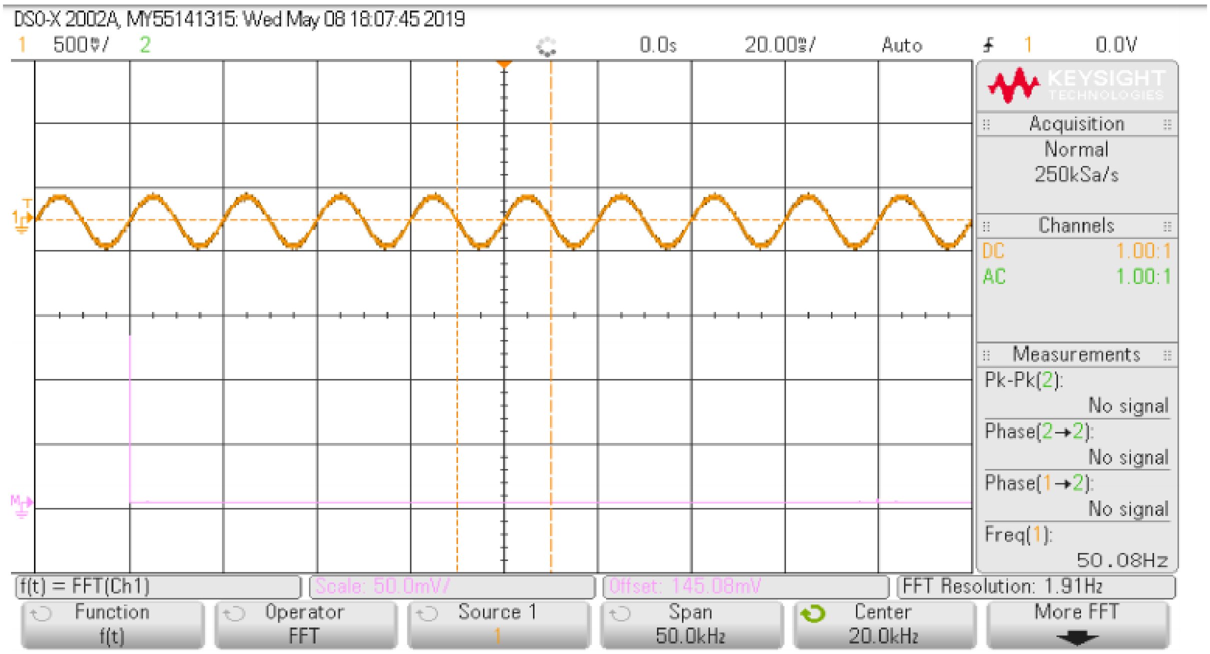

Figure 15: Oscilloscope output in time and frequency domain of output modulated signal of 0.5f1(t)

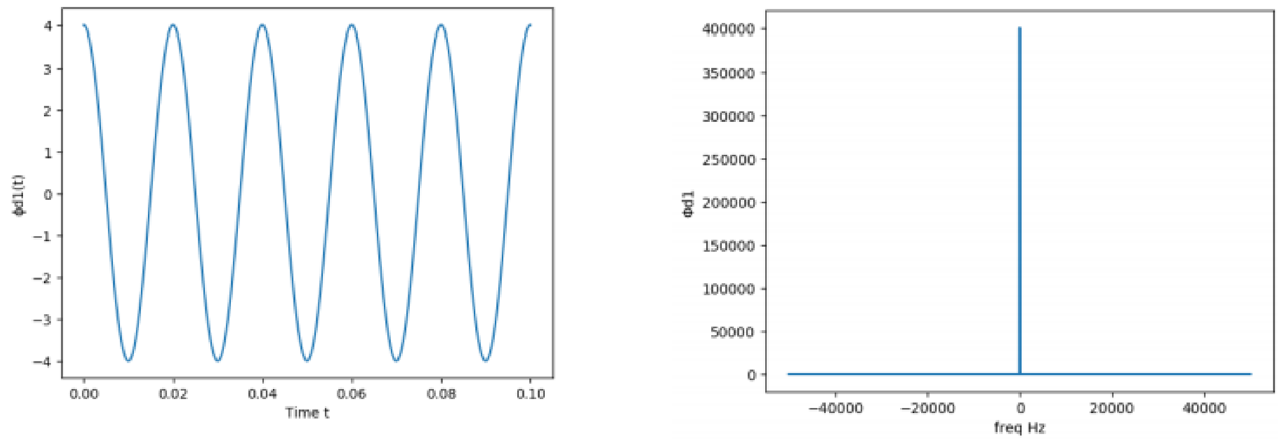

Figure 16: Simulated output in time and frequency domain of output modulated signal of 0.5f1(t)

In time domain, both simulated and oscilloscope outputs are similar and as expected should be a sinusoidal retrieving f1(t) but with having half of its amplitude as compared to the original

In frequency domain, we do see just a peak around 72Hz in the oscilloscope waveform and in the simulated signal. But again, we observe a second peak at -72Hz in the simulated signal, which should be self-explanatory now.
The 72Hz peak makes sense because it lets the 50Hz modulated signal pass through while blocking the high frequency carrier signal.

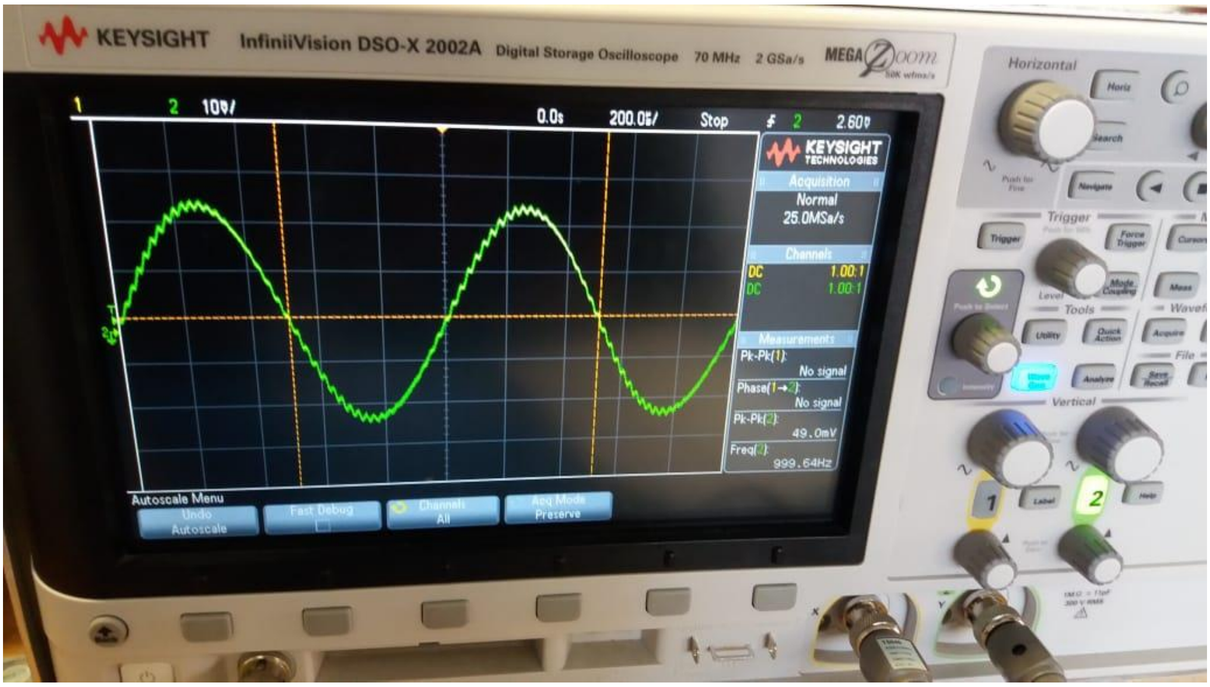

Figure 17: Oscilloscope output in time domain of output modulated signal of 0.5f2(t)

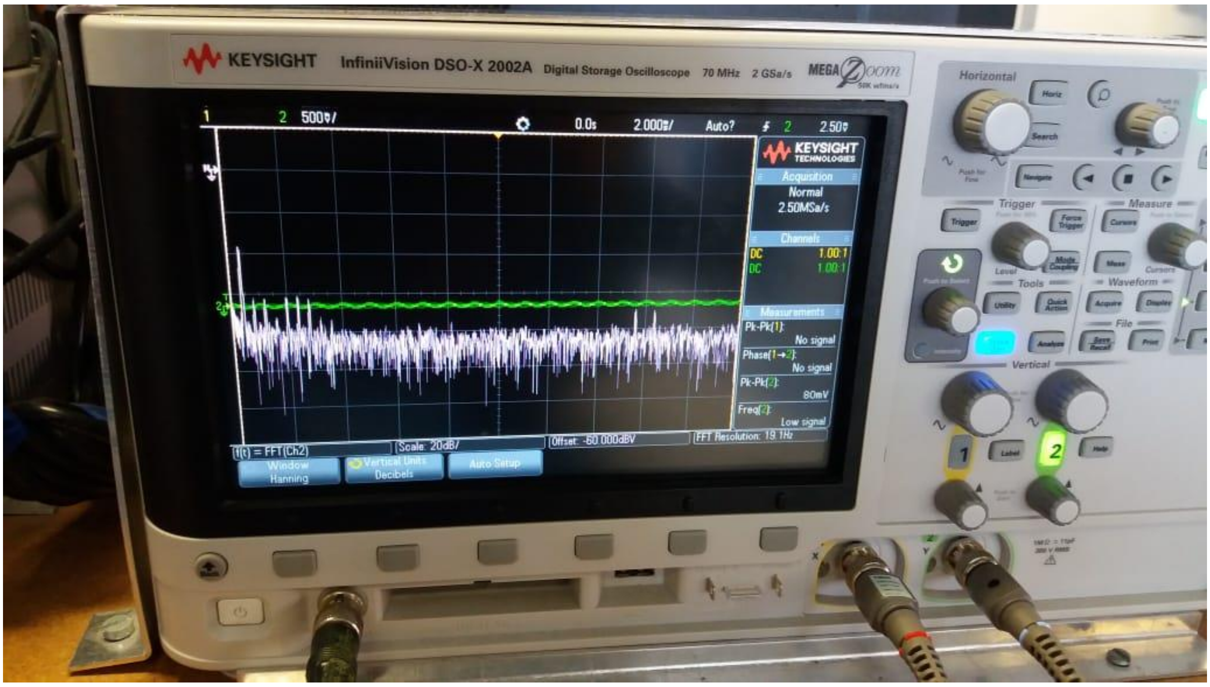

Figure 18: Oscilloscope output in frequency domain of output modulated signal of 0.5f2(t)

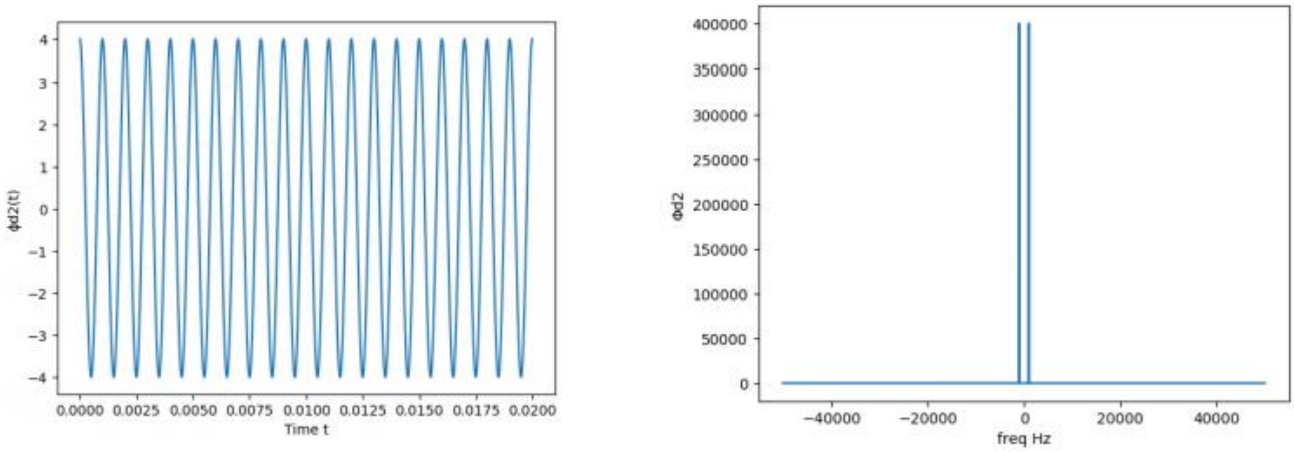

Figure 19: Simulated output in time and frequency domain of output modulated signal of 0.5f2(t)

In time domain, both simulated and oscilloscope outputs are similar and as expected should be a sinusoidal retrieving f2(t) but with having half of its amplitude as compared to the original

In frequency domain, we do see just a peak around 1.6kHz in the oscilloscope waveform and in the simulated signal. But again, we observe a second peak at -1.6kHz in the simulated signal, which should be self-explanatory now.
The 1.6kHz peak makes sense because it lets the 1.6kHz modulated signal pass through while blocking the high frequency carrier signal.

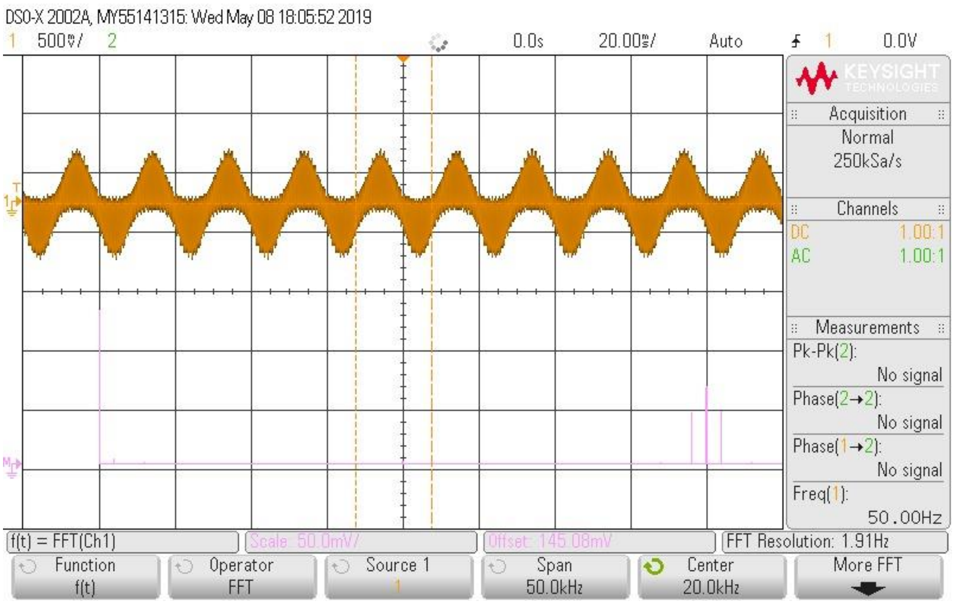

Figure 20: Oscilloscope output in time and frequency domain of output modulated signal but phase shifted with less than 90°

Effect of varying the phase shift of the carrier by varying the potentiometer on the phase-shifter module results in cross over between signals and causes the overall signal to be slightly distorted. Thus, to have a purely 50Hz or a purely 1kHz signal, there must be a phase shift of exactly 90° otherwise there will impure signal resulting in crossovers amongst them.

APPENDIX

Julia Code

```
#SSB-SC Modulation

using PyPlot;
using FFTW;
Δt = 0.0000001
t = 0:Δt:0.002

wm = 2*pi*1000; wc = 2*pi*20000;
fm = 2*cos.(wm*t);
fc_cos = 2*cos.(wc*t); fc_sin = 2*sin.(wc*t);
f_m = 2*sin.(wm*t);

φ1 = fm.*fc_cos; φ2 =f_m.*fc_sin;
φ = φ1+φ2;
Φ1 = fft(φ1); Φ2 = fft(φ2);
Φ = fft(φ);

figure();
plot(t, φ);
xlabel("Time t");  ylabel("φ(t)");

figure();
#plot(f_axis, fftshift(abs.(Φ)));
plot(f_axis[Int(floor(N/2)-a):Int(floor(N/2)+a)], fftshift((abs.(Φ)))[Int(floor(N/2)-a):Int(floor(N/2)+a)])
xlabel("freq Hz");
ylabel("Φ");

#LPF
function rect(t)
N = length(t)
x = zeros(N)
for n=1:N
abs_t = abs(t[n]);
  if abs_t > 0.5
  x[n]=0.0
  elseif abs_t < 0.5
  x[n]=1.0
  else
  x[n]=0.5
 end
end
return x
end
Δω = 2*pi/(N*Δt) # Sample spacing in freq domain in rad/s
ω = 0:Δω:(N-1)*Δω
f2 = ω/(2*π)
B = 1500 # filter bandwidth in Hz
B2 = 60 # filter bandwidth in Hz
```

```
H = rect(ω/(4*π*B)) + rect( (ω .- 2*π/Δt)/(4*π*B) );
H2 = rect(ω/(4*π*B2)) + rect( (ω .- 2*π/Δt)/(4*π*B2) );
Fdem_1=Fdem.*H;
Fdem_2=ifft(Fdem_1);
figure()
plot(t,Fdem_2) #Demodulated wave after filter
xlabel("Time in seconds");
ylabel("Amplitude");
figure();
xlabel("Frequency in Hz");


# Quadrature Multiplexing

F3 = 50; # 1 kHz
ω3 = 2*pi*f3; # rad/s
fm3 = A*sin.(ω3*t);
fmd2=(fm.*fc_cos) + (fm3.*fc_sin);

#multiplex signals
φ1 = f1.*fc_cos;
φ2 = f2.*fc_sin;
φ = φ1 + φ2;
Φ1 = fft(φ1);
Φ2 = fft(φ2);
Φ = fft(φ);

figure();
plot(t, φ);
xlabel("Time t");
ylabel("φ(t)");

figure();
#plot(f_axis, fftshift(abs.(Φ)));
plot(f_axis[Int(floor(N/2)-a):Int(floor(N/2)+a)],
fftshift((abs.(Φ)))[Int(floor(N/2)-a):Int(floor(N/2)+a)])
xlabel("freq Hz");
ylabel("Φ");

figure()
plot(t,fmd2) #Demodulated wave after filter
xlabel("Time in seconds");
ylabel("Amplitude");
Fmd22 = fft(fmd2);
figure();
  plot(f_axis, fftshift(abs.(Fmd22)),".-");
xlabel("Frequency in Hz");
fdem2=fmd2.*fc_cos ;
  Fdem2=fft(fdem2);
  fdem3=fmd2.*fc_sin ;
  Fdem3=fft(fdem3);
```