Jacobs University Bremen

Natural Science Laboratory
Measurement and Automation Lab

Spring 2022

Author of report: Roshan Nepal

Experiment conducted by : Roshan Nepal and Aditya Ojha

1 Introduction:

In the lab, the function generator, spectrum analyzer, vector network analyzer, and nano vector network analyzer are among the instruments studied. These devices were installed in a university lab, and we were required to control them remotely using Anydesk and MATLAB. For the first part of the lab, these devices were remotely controlled and made to perform efficiently with the help of Anydesk. The MATLAB begin files were provided and we were required to complete the script in order to remotely control the device.

In the second part of the lab assignment, LabView was used to model the device's blocks in a similar way to MATLAB and then used to remotely control the device. The starting files were also provided in lab view, and the missing blocks with values and parameters were inserted for proper implementation.

Finally, a Nano VNA was provided for the final section of the lab, and various properties of the cables and balun were investigated, including the s-parameters, characteristic impedance, and ABCD parameters. First, the nano VNA was needed to be calibrated to the perfect scale for precise measurement. Following the calibration, the open, short, and termination measurements for the balun were performed, and all other parameters were determined. In the case of cable, however, just open and short measurements were required. In addition, MATLAB was used to simulate the nano VNA, with the initial text file for open, short, and termination for cables and balun. The plots of Amplitude of characteristic impedance, Phase of characteristic impedance, Alpha (attenuation), and beta were determined using these files.

2 Execution

2.1 Spectrum Analyzer (SA):

The spectrum analyzer was setup with a function generator. The function generator was used to generate a wave shape within a range of frequency going from 1 µHz to 20 MHz. The input function was chosen as sinusoid as well as square wave with a load impedance of 50 Ω. The sweeping was done for the whole frequency range and the output was observed.

Now, in the spectrum analyzer, the output plots were not seen as discrete spectral lines for sinusoid function which is expected. This occurs due to the function principle of the spectrum analyzer. The analyzer sweeps the signal through the chosen frequency range and modulation and shift in frequency occurs through the filter carried out by resolution bandwidth. Increasing the resolution bandwidth leads the spectrum to be wider and broader. Narrowing down the resolution bandwidth can lead the spectrum to look more like the expected line. However, narrowing the resolution bandwidth means the sweep rate was slower. The resolution bandwidth of 100 kHz was chosen. Similarly, another parameter used was video bandwidth which when made narrower had a denoising effect acting as an averaging effect. The sweep time depended on these two parameters of resolution and video bandwidth.

The MATLAB script for SA is given below:

```
clc; close all; clear all;
h = visa('ni','TCPIP::10.70.13.175::INSTR');  % check IP address!!!
h.inputbuffersize = 1000000;
fopen(h);
fprintf(h, 'FREQ:START 9KHz')      % set start frequency
fprintf(h, 'FREQ:STOP 20MHz')      % set stop frequency

                                   %similar to the resoltion bandwidth,
fprintf(h, 'BAND:VID 1kHz')        % set video bandwidth to 1 kHz

%the narrower the resolution bandwidth, the more distinctive the frequency
%compoment but at the cost of larger sweep time.

fprintf(h, 'BAND:RES 100kHz')      % set resolution bandwidth to 100 kHz

fprintf(h, 'SWE:POIN 10001\n')     % set number of points to 10001
fprintf(h,'FORM ASCII');
pause(40)                          %waiting for some time so that complete data is plotted on
                                   the screen of instrument.
fprintf(h,'TRAC? TRACE1');
tr = fscanf(h);
fclose(h);
trace = str2num(tr);
size(trace);
freq = linspace(1,4, 601);         %produce frequency axis
plot(freq,trace);
xlabel('Frequency (Hz)');
ylabel('Magnitude (Hz)');
title('Plot of 9 KHz to 20 MHz for video Bandwidth of 1KHz');
```

Figure 1: MATLAB script for Spectrum Analyzer

The block diagram for LabView is given below:

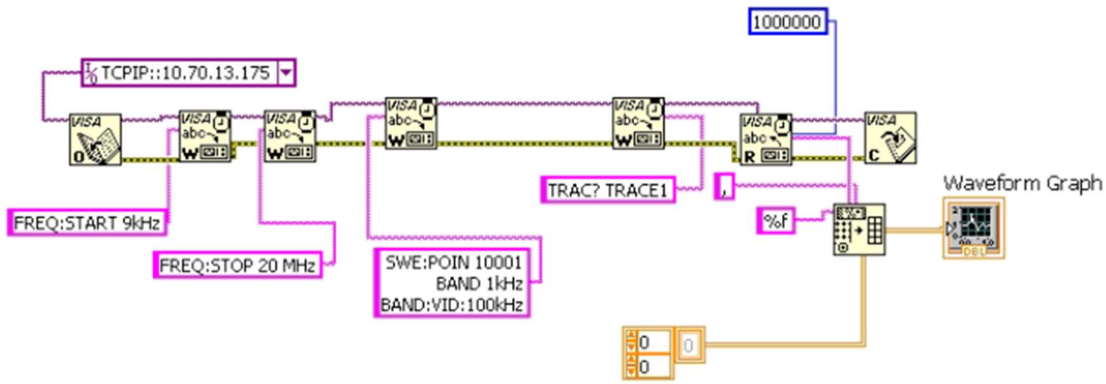

Figure 2: LabView Block diagram for Spectrum Analyzer

- MATLAB and LabView Plots:

  a. Frequency sweep range 9 kHz to 20 MHz for sine:

Figure 3: SA output plot for 9 kHz to 20 MHz range for sine

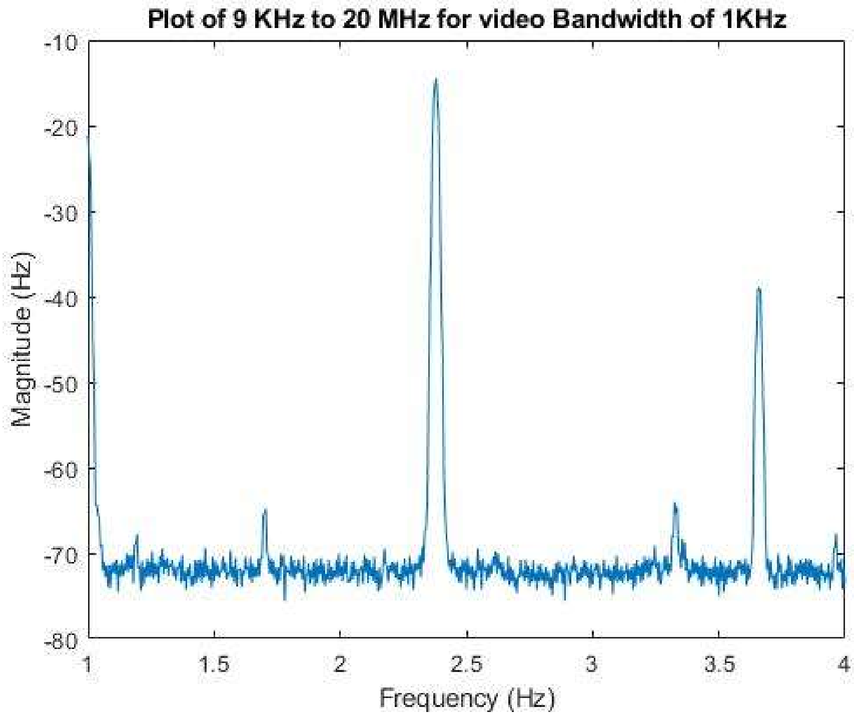

Figure 4: MATLAB plot for 9 kHz to 20 MHz for sine

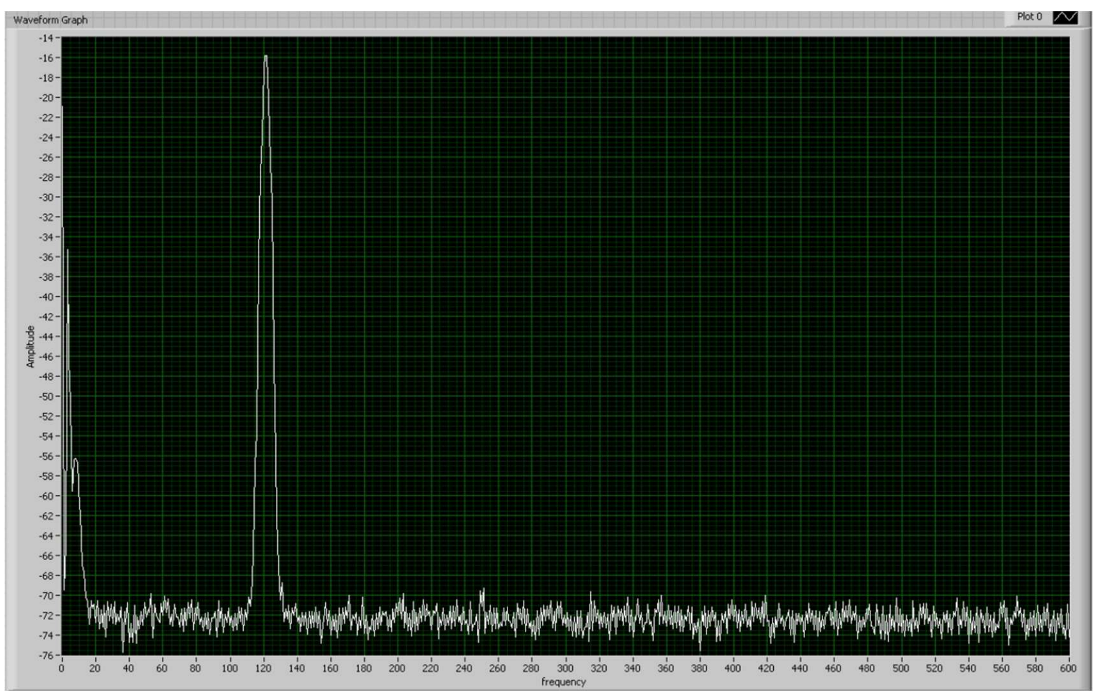

Figure 5: LabView plot for 9 kHz to 20 MHz for sine

b. Frequency sweep range 9 kHz to 30 MHz for sine:

Figure 6: SA output plot for 9 kHz to 30 MHz range for sine

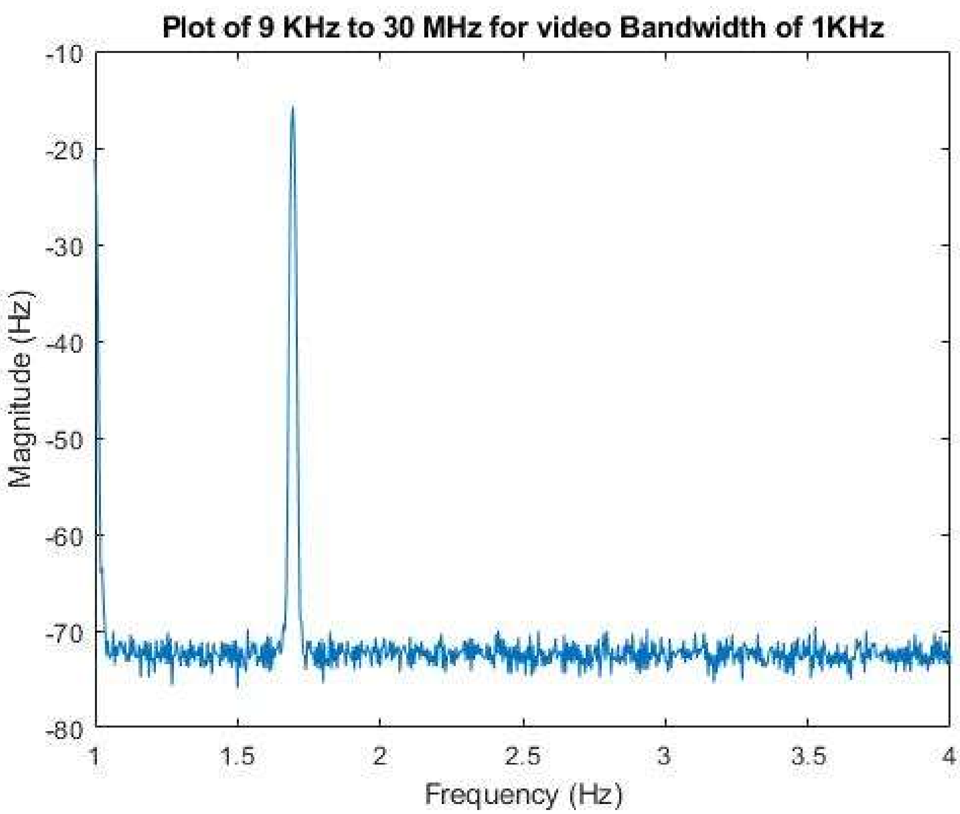

Figure 7: MATLAB plot for 9 kHz to 30 MHz for sine

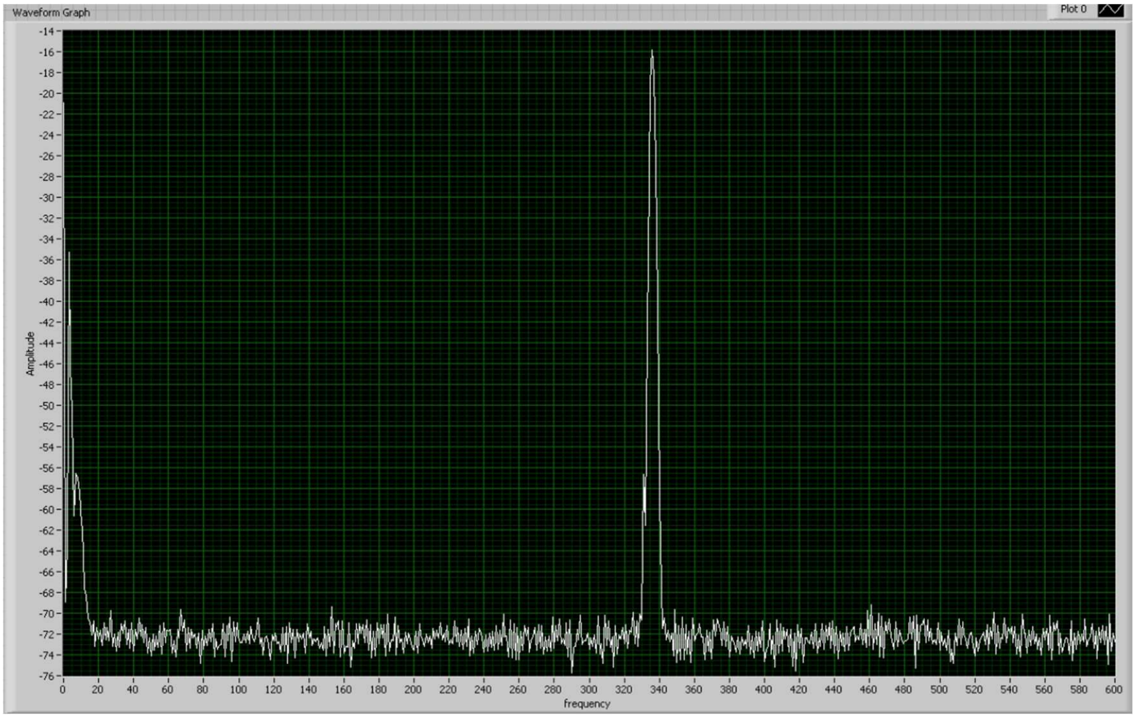

Figure 8: LabView plot for 9 kHz to 30 MHz for sine

c. Frequency sweep range 9 kHz to 20 MHz for square wave:

Figure 9: SA output plot for 9 kHz to 20 MHz range for square wave

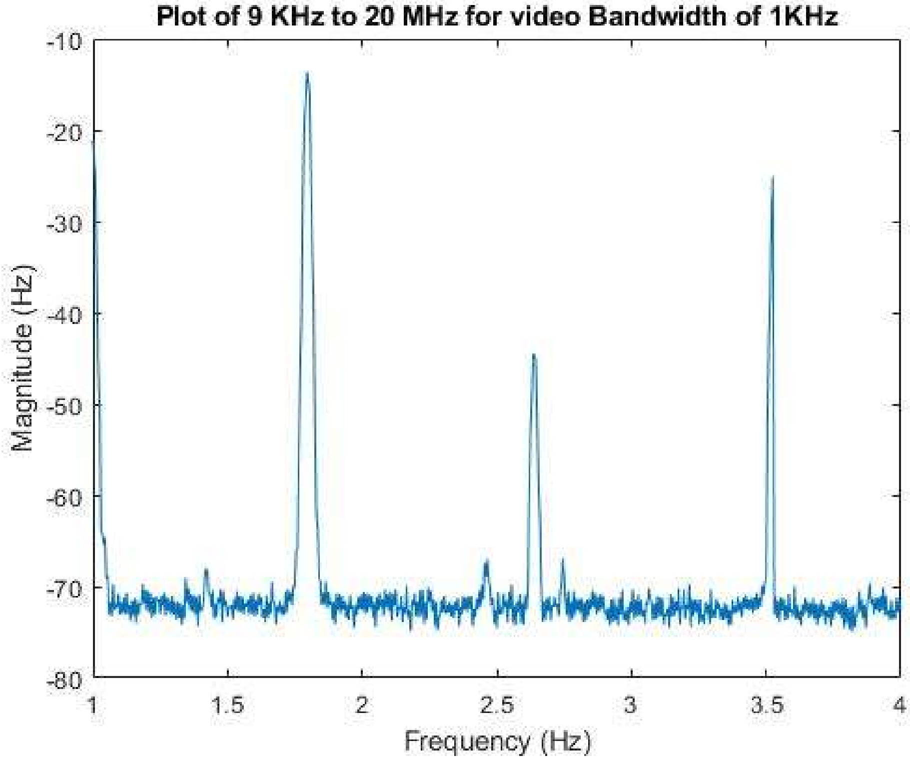

Figure 10: MATLAB plot for 9 kHz to 20 MHz for square wave

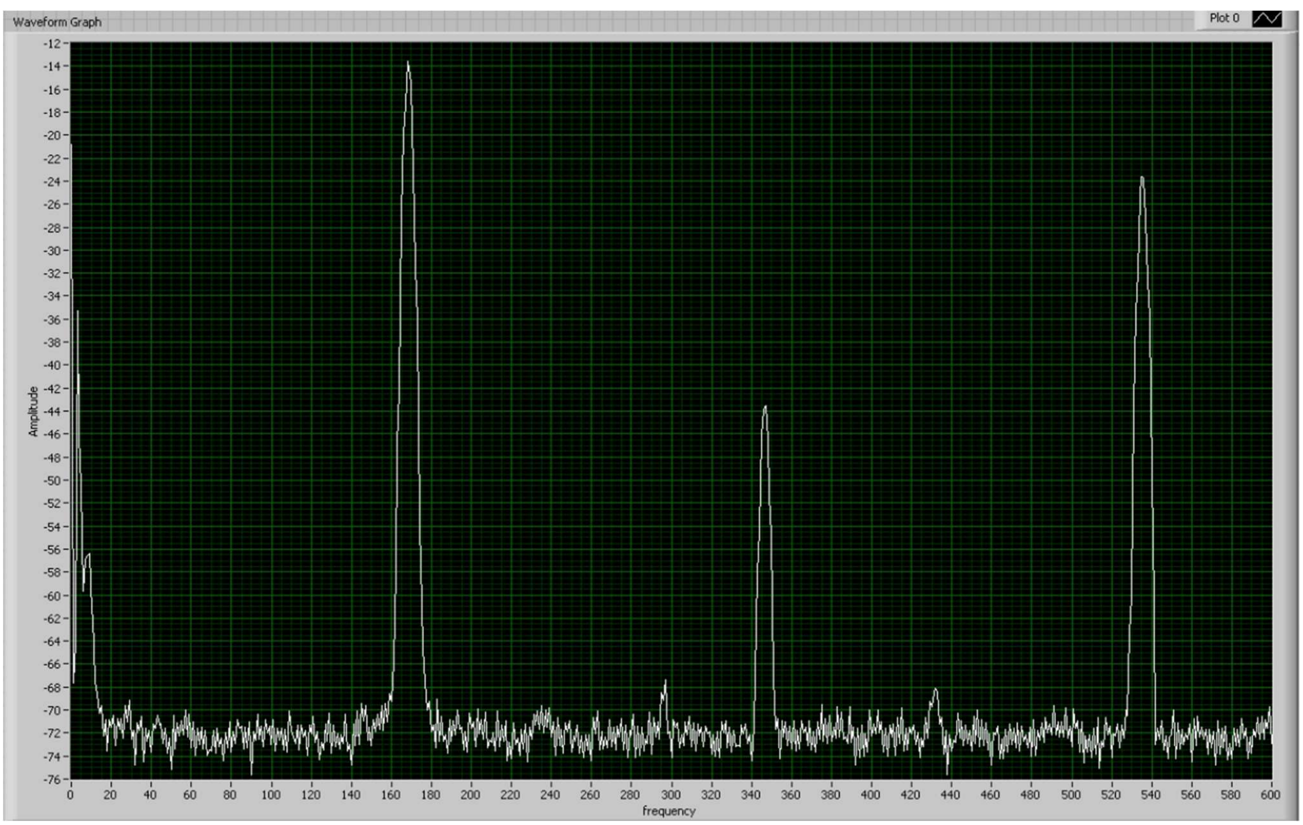

Figure 11: LabView plot for 9 kHz to 20 MHz for square wave

d. Frequency sweep range 9 kHz to 30 MHz for square wave:

Figure 12: SA output plot for 9 kHz to 30 MHz range for square wave

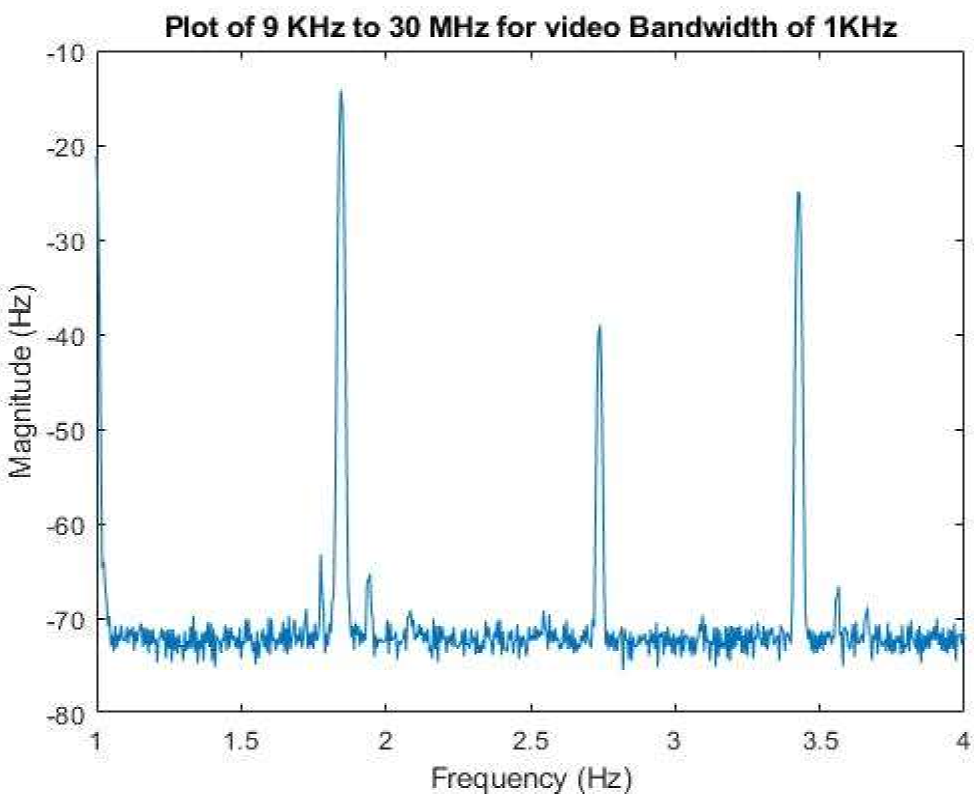

Figure 13: MATLAB plot for 9 kHz to 30 MHz for square wave

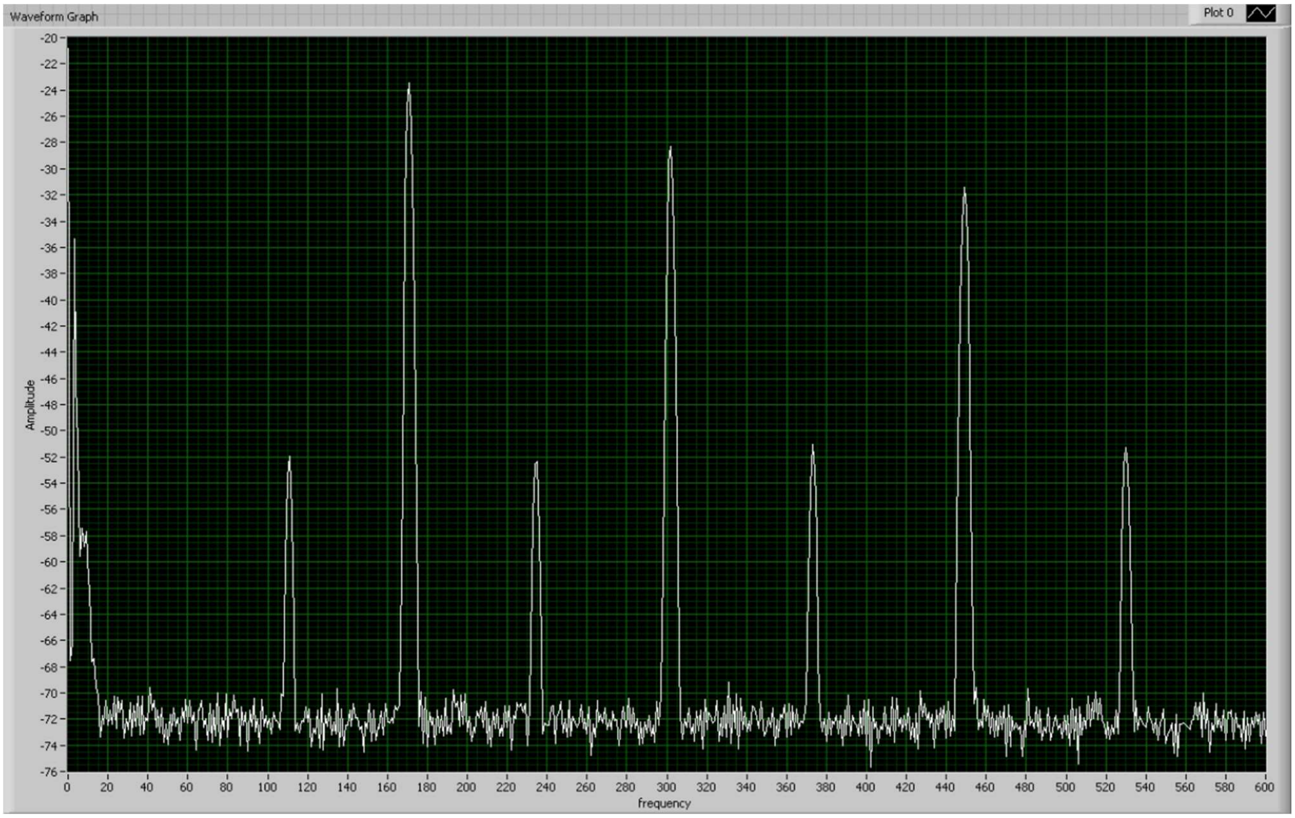

Figure 14: LabView plot for 9 kHz to 30 MHz for square wave

2.2 Vector Network Analyzer (VNA):

This lab task was completed using the Rhode and Schwarz VNA. The measuring parameters of coaxial cable over the balun (homemade or professional) were measured using this instrument. In my situation, the start and stop frequencies were set to 9KHz and 200 MHz, respectively. For this instrument, the sweep points were set at 801 with a maximum of 2001. The 'LOG SWEEP' with log division of the frequency scale was used. The smith chart format was used for the output plot. In order to gain a sense of the cable's parameters. It was necessary to calibrate the VNA. However, because it was already set up in a lab setting, it was assumed that the equipment was fully calibrated. Later, while discussing nano VNA, the calibration task will be explored in further depth.

The MATLAB script for VNA is:

```
clc; clear all; close all;
test_obj=visa('ni','TCPIP0::10.50.212.8::gpib0,2::INSTR'); % check IP address
%test_obj = visa('ni', 'GPIB2::10.50.212.8::0::INSTR');
set(test_obj,'InputBufferSize', 100000);
fopen(test_obj);
fprintf(test_obj, '*IDN?');
fscanf(test_obj)
fprintf(test_obj, 'INSTRument:SELect CHANNEL1');   % select channel 1
fprintf(test_obj, 'SENS:FREQ:START 9KHz');         % set start frequency
fprintf(test_obj, 'SENS:FREQ:STOP 200MHz');        % set stop frequency
fprintf(test_obj, 'SENS:SWE:POIN 801');    % set number of points <= 801 (NA   limit <=2001)
fprintf(test_obj, 'FREQ:MODE SWEEP');
fprintf(test_obj, 'SENS:SWE:SPAC LOG');    % set log sweep
```

```
fprintf(test_obj, 'CALC1:FORM COMP');                 % set format complex (Smith chart)
fprintf(test_obj, 'INIT:CONT OFF');                   % turn off continuous mode , we run the
                                                      code twice in order to get accurate plot
fprintf(test_obj, 'INIT:IMM; *WAI') ;                 % wait command
fprintf(test_obj, 'TRAC? CH1DATA');                   % read data of channel 1
s = fscanf(test_obj);                                 %data_points
data_points = str2num(s);
fclose(test_obj) ;
re = data_points(1:2:1602);
im = data_points(2:2:1602);
plot(re,im);
title('Smith Chart for Start Frequency = 10Hz and Stop Frequency = 40Mhz');
fid = fopen('cableopen.txt','w');
fprintf(fid, '%f', data_points);
fclose(fid);
```

Figure 15: MATLAB script for Vector Network Analyzer

The LabView block diagram for VNA is given below:

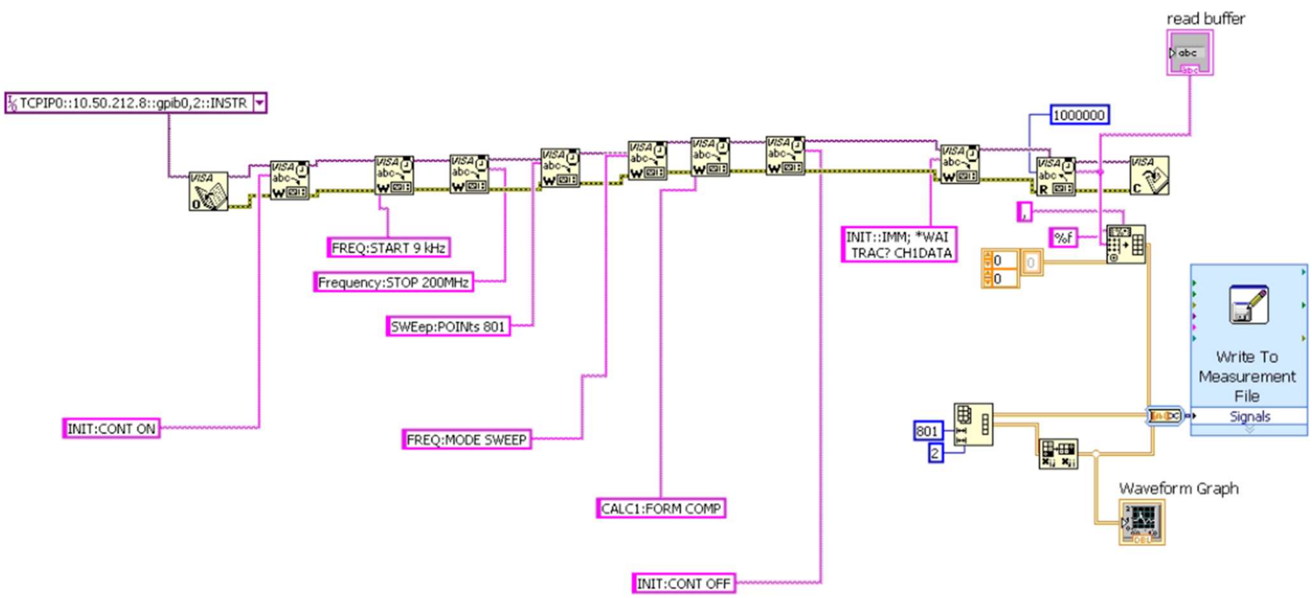

Figure 16: LabView block for VNA

MATLAB and LabView Plots:

a. Start frequency 9 kHz; Stop frequency 200 MHz:

Figure 17: VNA output plot for frequency range 9 kHz to 200 MHz

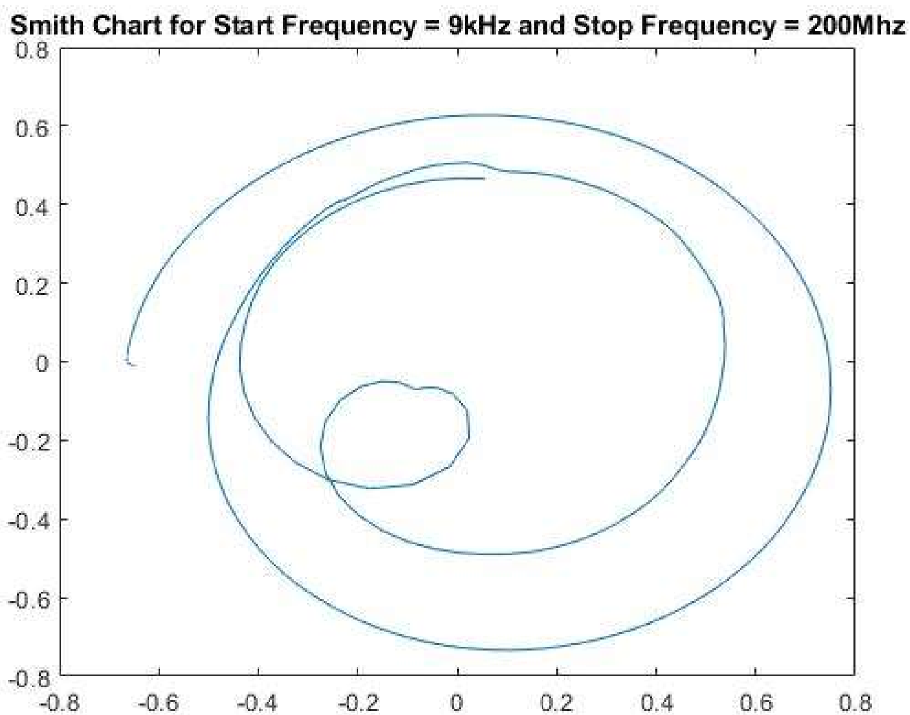

Figure 18: MATLAB plot for frequency range 9 kHz to 200 MHz

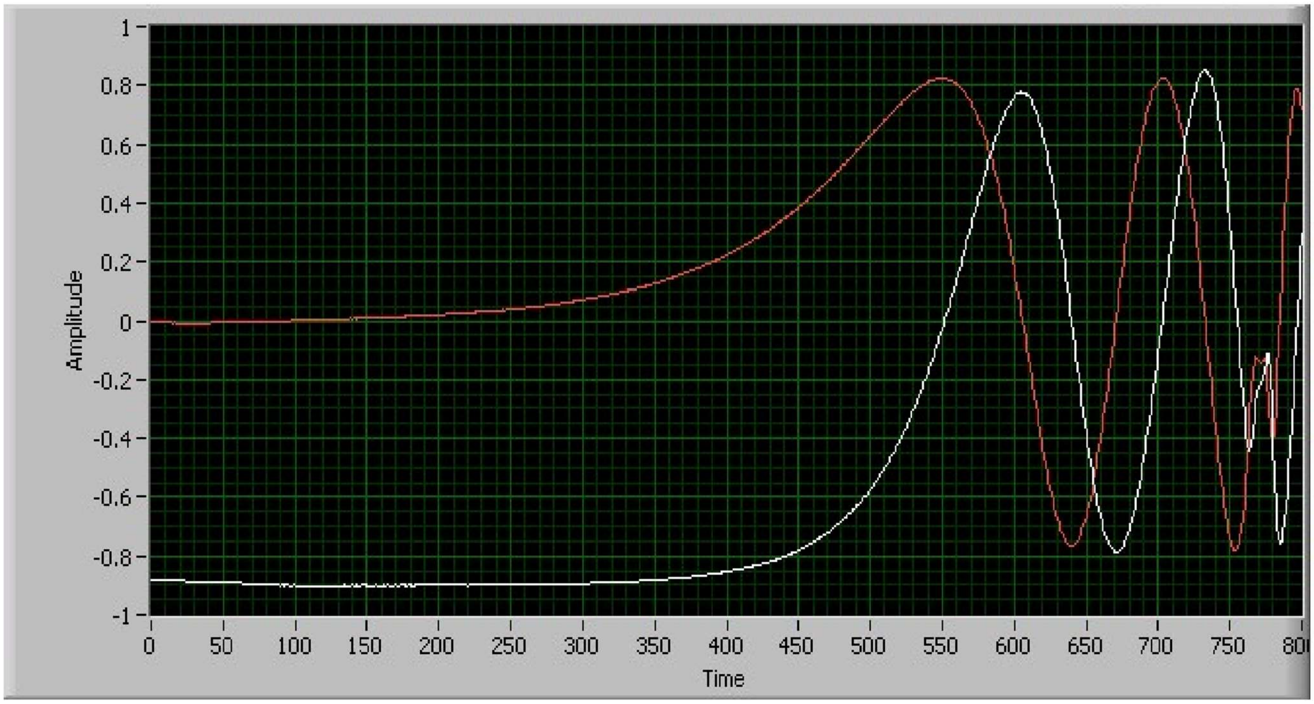

Figure 19: LabView output for VNA frequency range 9 kHz to 200 MHz

b. Start frequency 9 kHz; Stop frequency 100 MHz:

Figure 20: VNA output plot for frequency range 9 kHz to 100 MHz

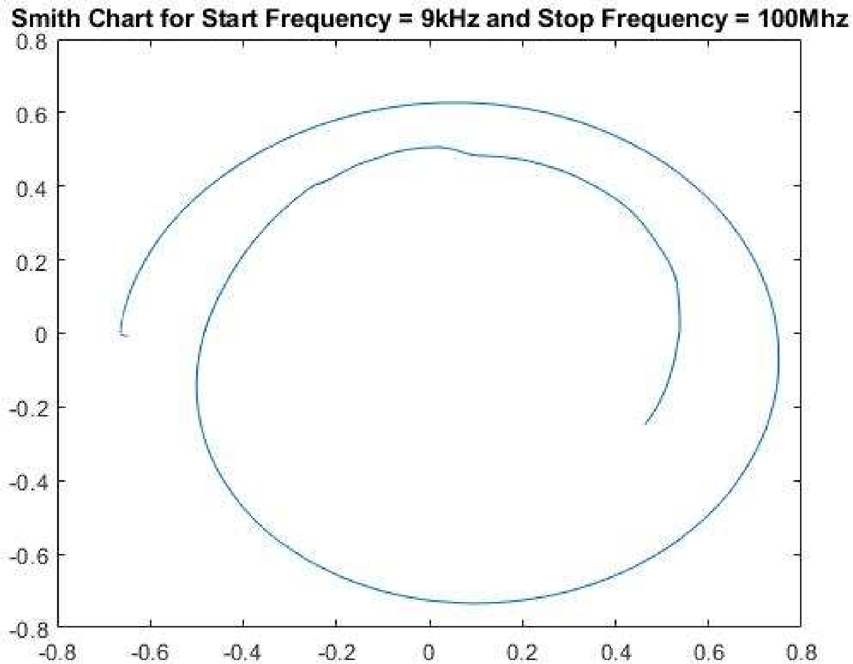

Figure 21: MATLAB plot for frequency range 9 kHz to 100 MHz

c. Start frequency 10 kHz; Stop frequency 40 MHz:

Figure 22: VNA output plot for frequency range 10 kHz to 40 MHz

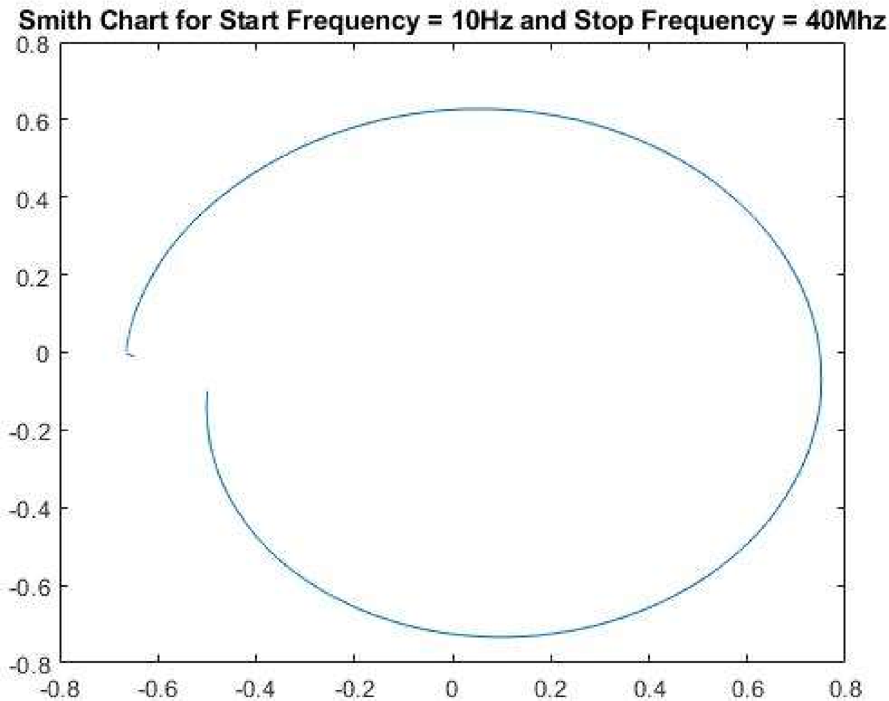

Figure 23: MATLAB plot for frequency range 10 kHz to 40 MHz

2.3 Digital Oscilloscope:

In this experiment, the non-stationary impulse noise at the two pairs of a quad telephone cable needed to be measured. The MATLAB code was used to turn on the two channels and set the vertical resolution and time base, along with the trigger level. It would then read the data from the channels, and plot it in the graph.

The MATLAB script for the digital oscilloscope is:

```
clc; clear all; close all;
test_obj=visa('ni','TCPIP::10.70.13.164::INSTR'); % check IP address
set(test_obj,'InputBufferSize', 100000);
fopen(test_obj);
fprintf(test_obj , '*IDN?');
fscanf(test_obj)
fprintf(test_obj, 'LOGGer:AUToset');        % AUTOSET (seems to reset)
fprintf(test_obj, 'CHANnel1:STATe ON');     % Turn channel 1 on
fprintf(test_obj, 'CHANnel2:STATe ON');     % Turn channel 2 on
fprintf(test_obj, 'CHANnel1:SCALe 50');     % Setting the vertical res. Ch1
fprintf(test_obj, 'CHANnel2:SCALe 50');     % Setting the vertical res. Ch2
fprintf(test_obj, 'LOGGer:SOURCe CHANnel1');
fprintf(test_obj, 'LOGGer:TIMebase:SCALe nS20');% Setting the time base - Ch1
fprintf(test_obj, 'TRIGger:EDGE:SLOPe POSitive'); % Set trigger edge slope to positive
fprintf(test_obj, 'TRIG:LEV1:VAL 0.005'); % Set trigger level to 0.005
fprintf(test_obj, 'TRIGger:SOURce CHANnel2'); % Set trigger source to be channel 2
fprintf(test_obj, 'TRIGger:MODE SING');     % Set trigger mode to SINGLE
fprintf(test_obj, 'CHAN1:DATA?');                   % read data of Channel 1
s1 = fscanf(test_obj);
data_points1 = str2num(s1);
fprintf(test_obj, 'CHAN2:DATA?');                   % read data of Channel 2
s2 = fscanf(test_obj);
data_points2 = str2num(s2);
```

```
%s = fscanf(test_obj);
fclose(test_obj);
plot(data_points1);
hold on;
plot(data_points2);
hold off;
```

Figure 24: MATLAB script for Digital Oscilloscope

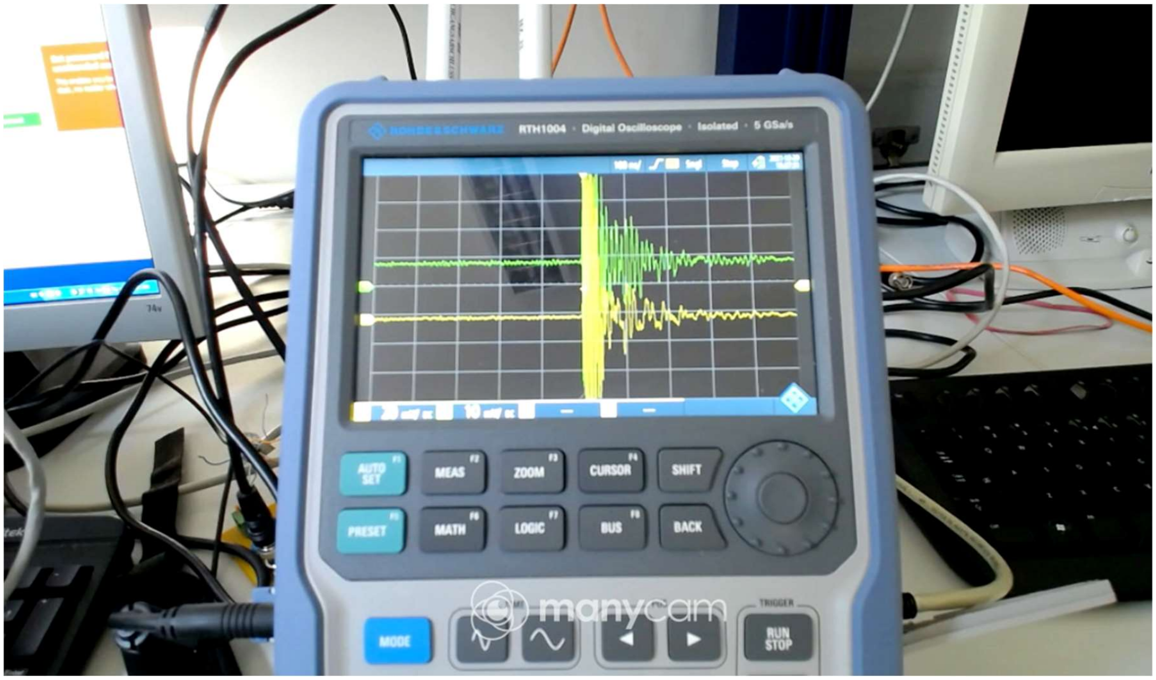

Figure 25: Output for Digital Oscilloscope

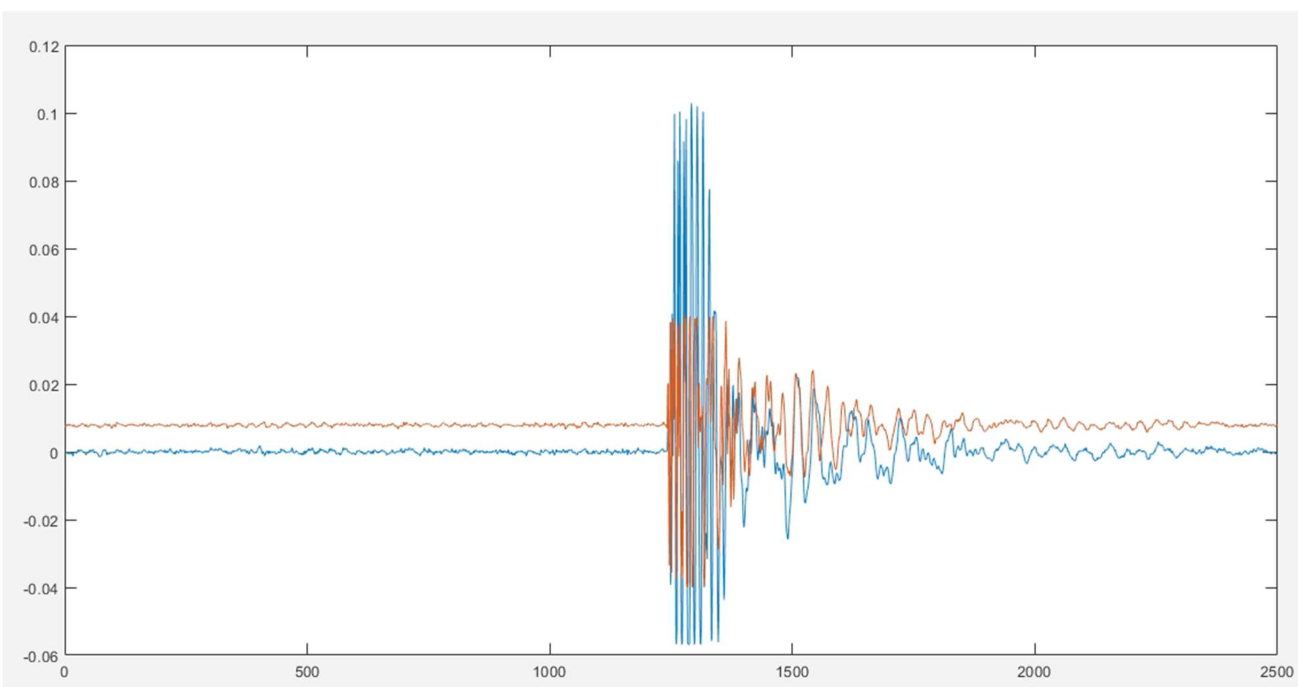

Figure 26: MATLAB Plot for Digital Oscilloscope

2.4 Function Generator:

It was used so as to generate a wave shape to be used for the device within certain range of frequency i.e. 1 μHz to 20MHz. In this particular case, the input function was chosen to be sinusoid with a load impedance of 50 Ω. The whole frequency range was swept and the output plot at the scope was observed.

The MATLAB script for function generator is:

```
h = visa('agilent','TCPIP::10.70.13.216::INSTR');
h.inputbuffersize = 1000000;
fopen(h);
fprintf(h, '*RST');                  %Reset the function generator
fprintf(h,'FUNC SINUSOID');          %Select waveshape
fprintf(h,'OUTPut:LOAD 50');         %Set the load impedance to 50 Ohms (default)
fprintf(h,'VOLTage 0.1');            %Set the amplitude to 100 mV-pp
fprintf(h,'SWEep:SPACing LIN'); %Set linear spacing;
fprintf(h,'SWEep:TIME 10');          %Set Sweep time 5s
fprintf(h,'FREQuency:STARt 1e(-6)'); %Set Start frequency 1 microHz
fprintf(h,'FREQuency:STOP 20e6'); %Set Stop frequency 20 MHz
fprintf(h,'');                       %Set type of internal triggering
fprintf(h,'OUTPut ON');   %Turn on the instrument output
fprintf(h,'SWEep:STATe ON');         %Turn sweep on
fclose(h);
```

Figure 27: MATLAB script for Function Generator

The LabView block diagram for function generator is:

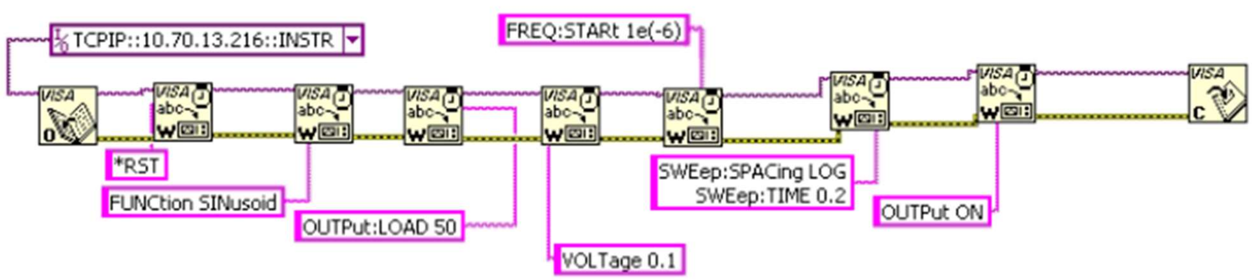

Figure 28: LabView Block diagram for Function Generator

2.5 Nano VNA:

The Nano Vector Network Analyzer along with coaxial cable, balun, ethernet cable, open, short and termination tools were provided. The measurements were carried out in the Nano VNA. Firstly, the VNA was calibrated with open, short and termination measurements as per the guideline from the manual. Then, the open, short and termination measurements were done with the calibration of balun and coaxial cable. Finally, the open and short measurements were done on the cable pair. The impedances were recorded and used for the calculation of the parameters.

- Physical Measurements:

Two measurements were done for the cable. One for open and another for short. The plots for the measurements in the Nano VNA are shown below:

- Cable Open

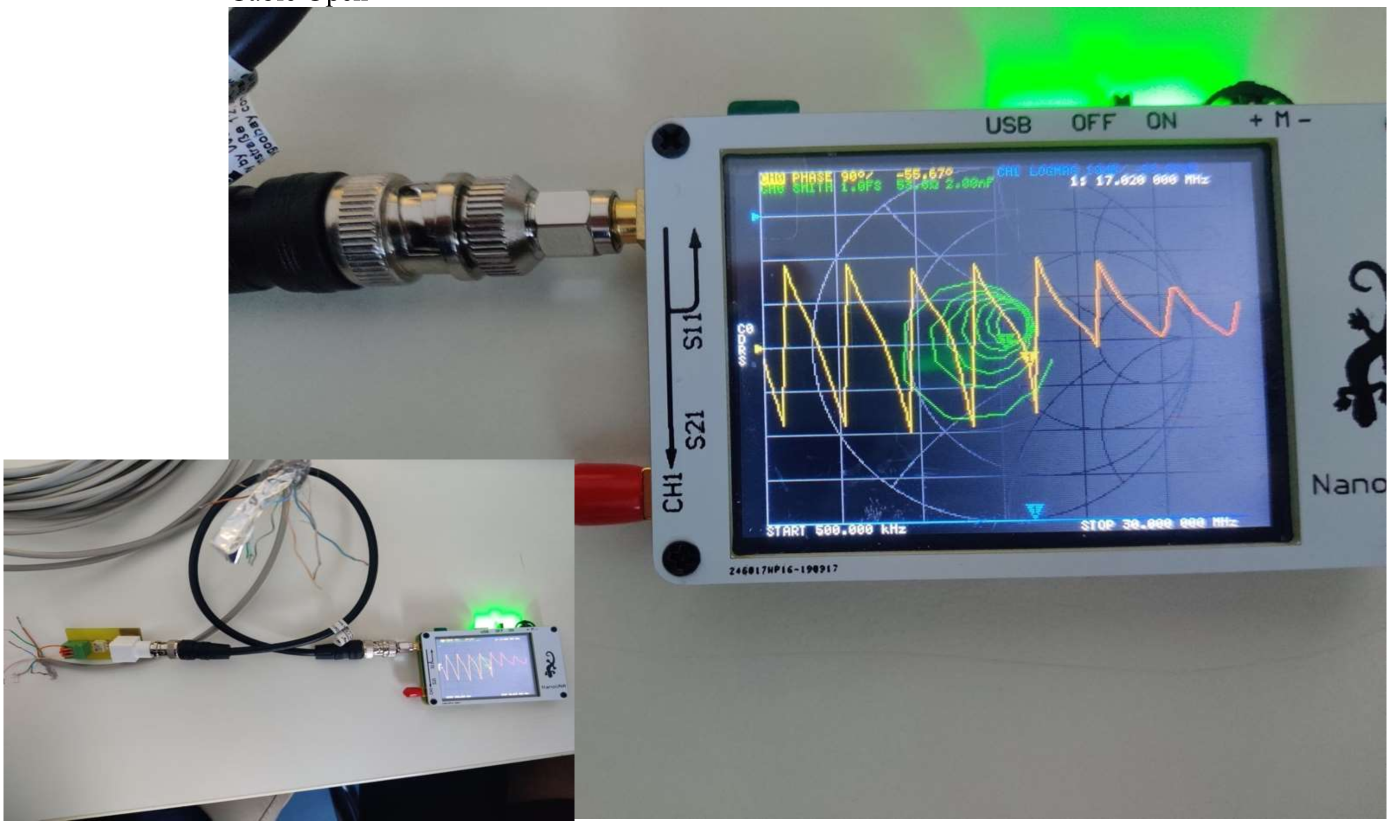

Figure 29: Nano VNA Measurement for Cable Open

- Cable Short

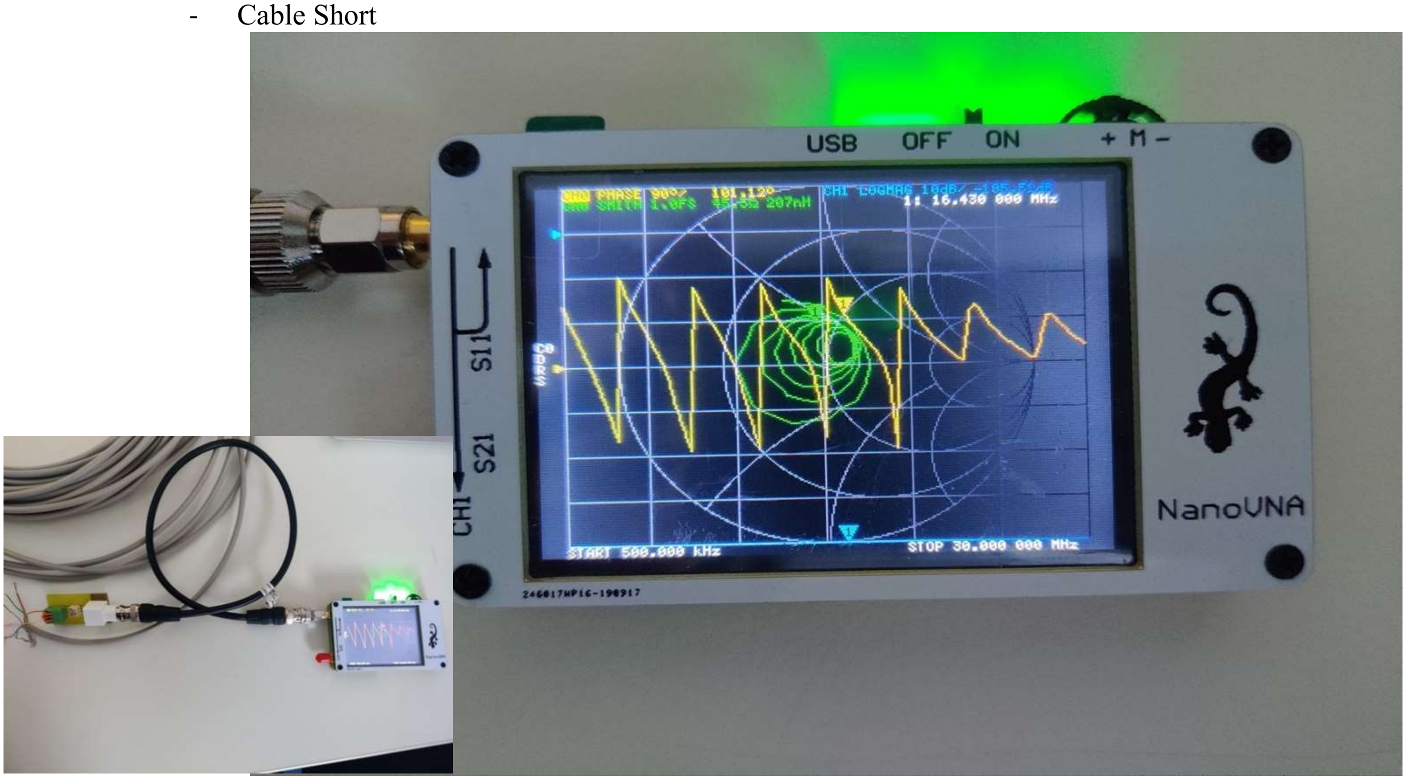

Figure 30: Nano VNA Measurement for Cable Short

Similarly, the measurements were done with balun connected with the VNA through the coaxial cable. Three measurements were carried out. Open, short and termination with 75Ω.

The plots for the measurements in the Nano VNA are shown below.

- Balun Open

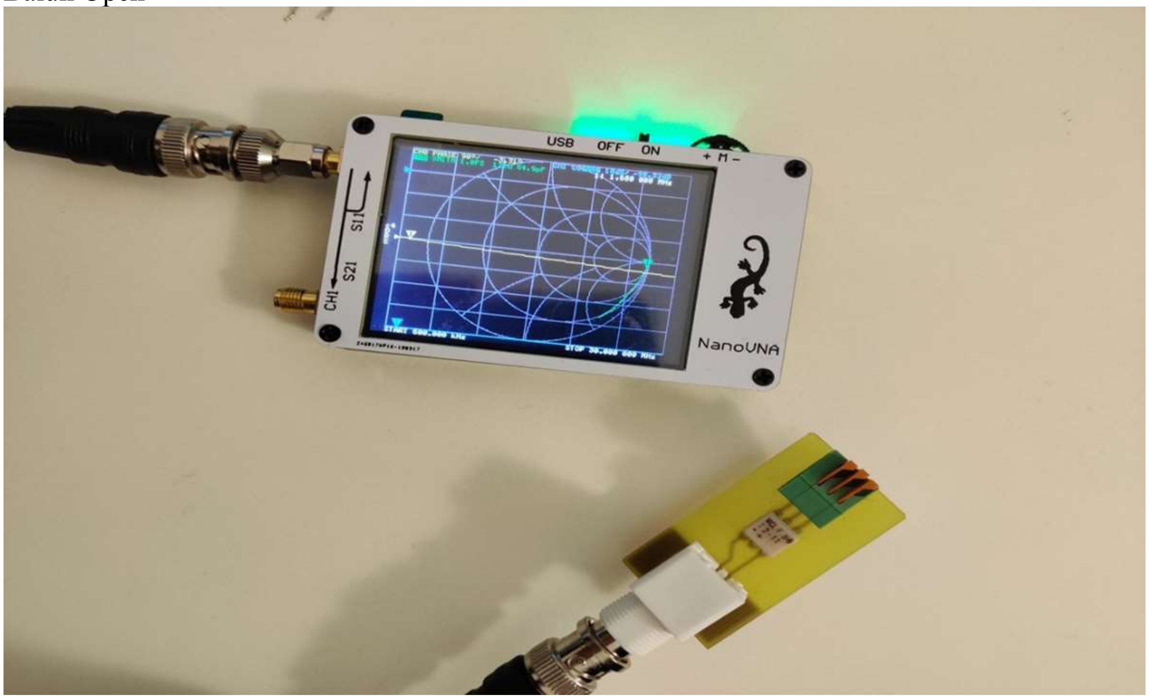

Figure 31: Nano VNA Measurement for Balun Open

- Balun Short

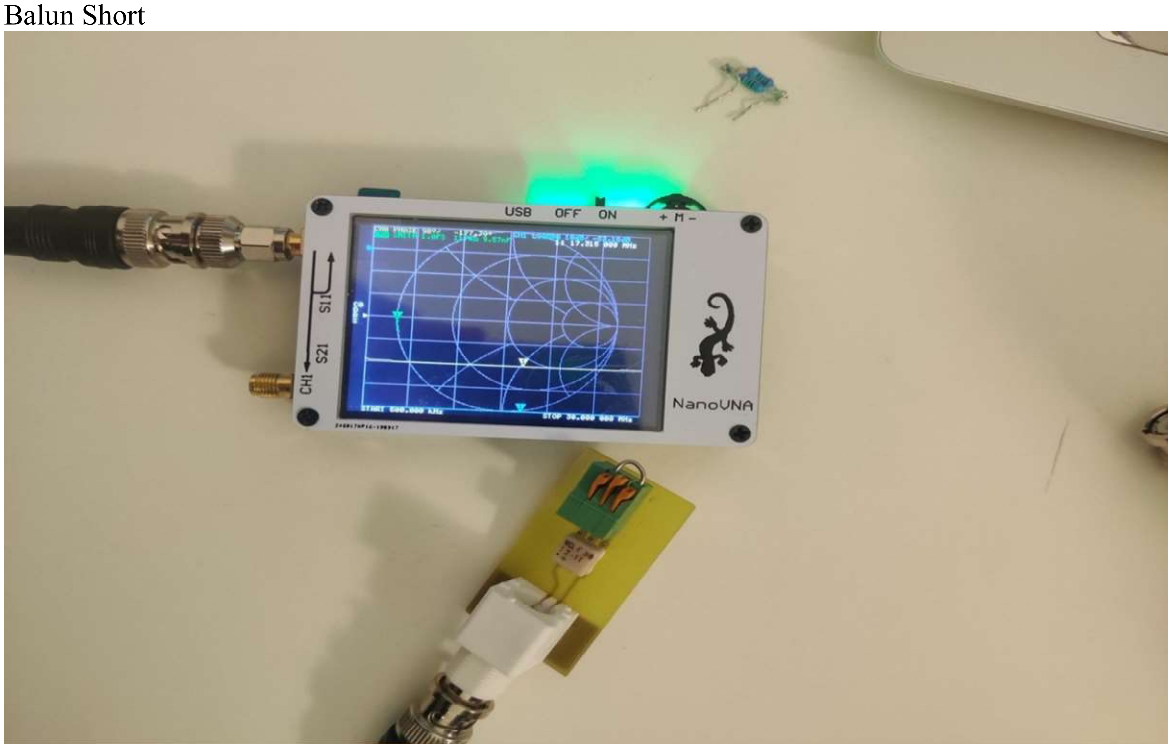

Figure 32: Nano VNA Measurement for Balun Short

- Balun Termination

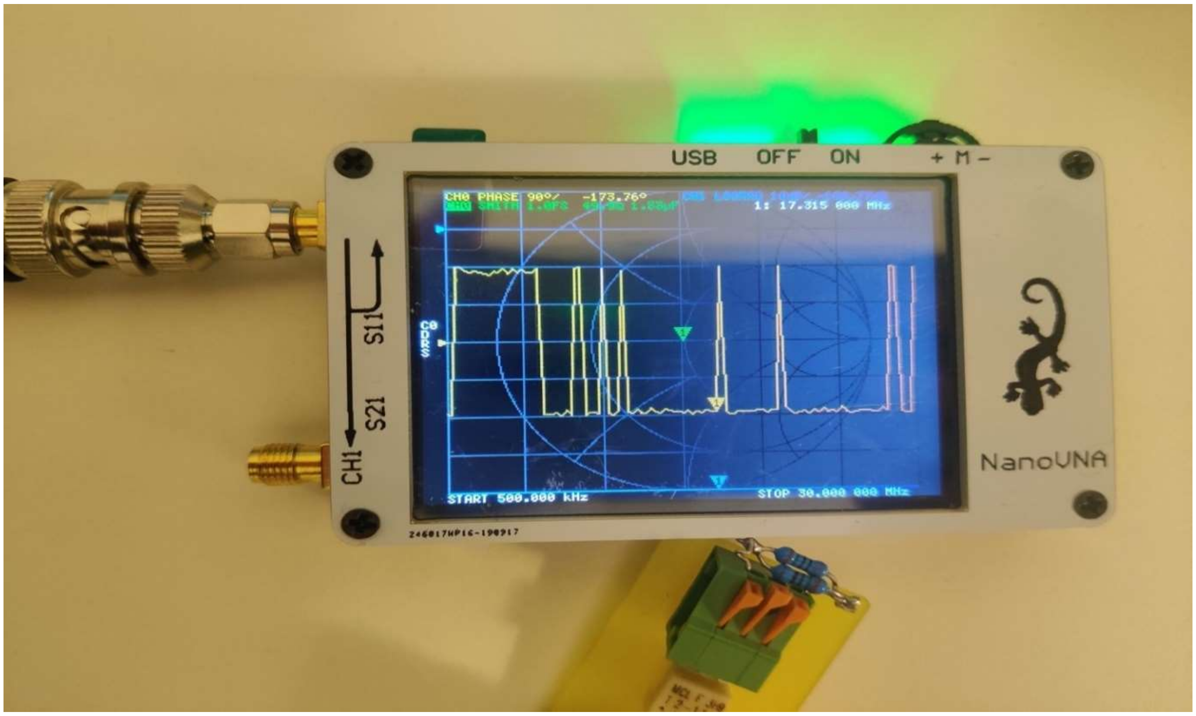

Figure 33: Nano VNA Measurement for Balun Termination

It can be seen from the green index in the figures that the points in the smith chart are in the expected positions that is to the right for open, left for short and in the middle for termination.

- Remote control of the Nano VNA via MATLAB and Nano VNA Saver (Python)
  As, with other instruments, the Nano VNA also can be used and controlled by a computer through a USB connection. Here, MATLAB as well as a python script for Nano VNA Saver were used to get familiar with a remote connection and use of the device. The measurements were done as similar to the physical Nano VNA task but the results were verified from the plots generated by each program used instead of the Nano VNA screen.

  - MATLAB
    The MATLAB script used was provided which can be accessed from the following link:
    https://github.com/qrp73/NanoVNA-MATLAB
    The results obtained as follows:

- Cable Open

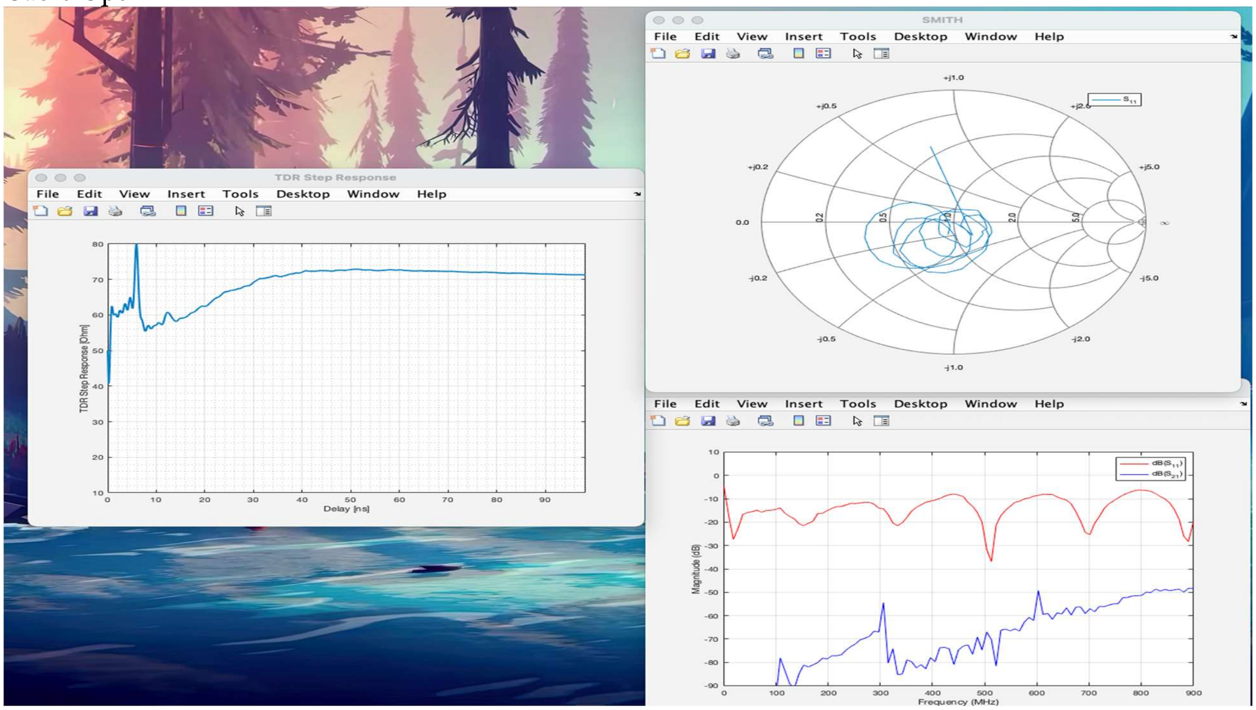

Figure 34: MATLAB Plots for Cable Open

- Cable Short

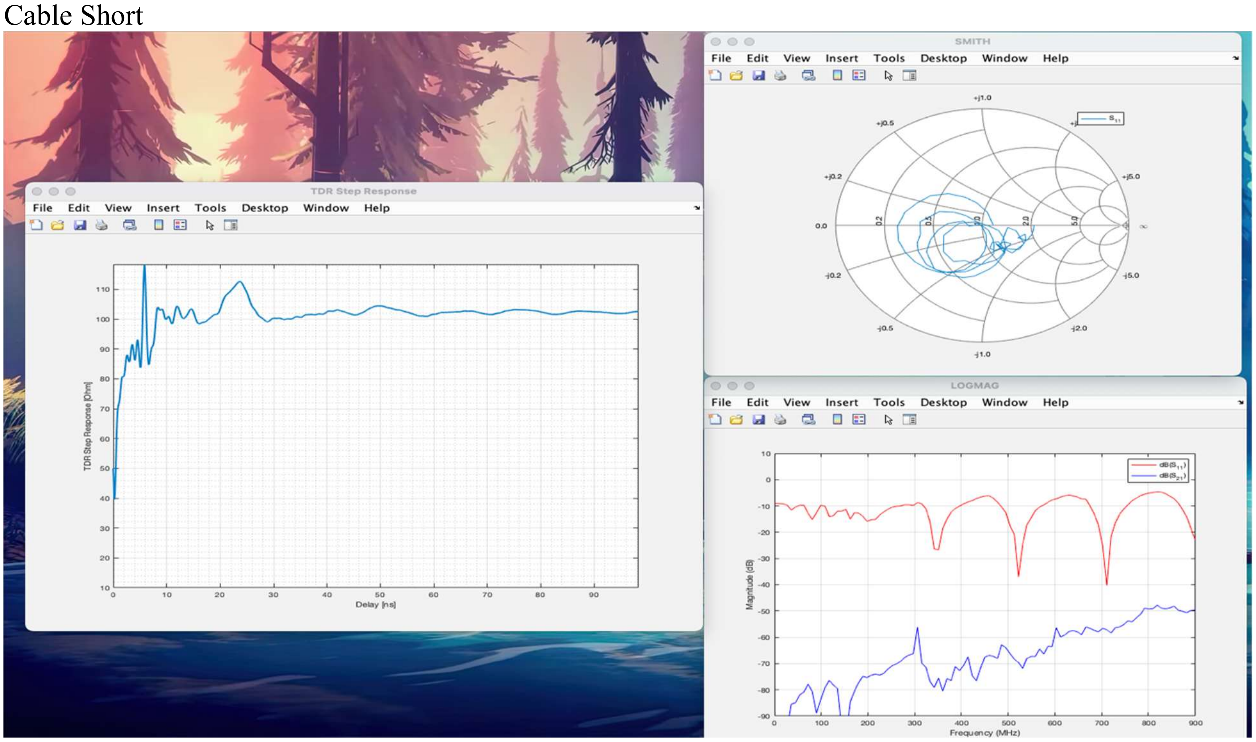

Figure 35: MATLAB Plots for Cable Short

- Nano VNA Saver
  Similarly, the drivers and the python code for Nano VNA saver were imported from the information provided and the plots generated from Nano VNA are:

- Cable Open

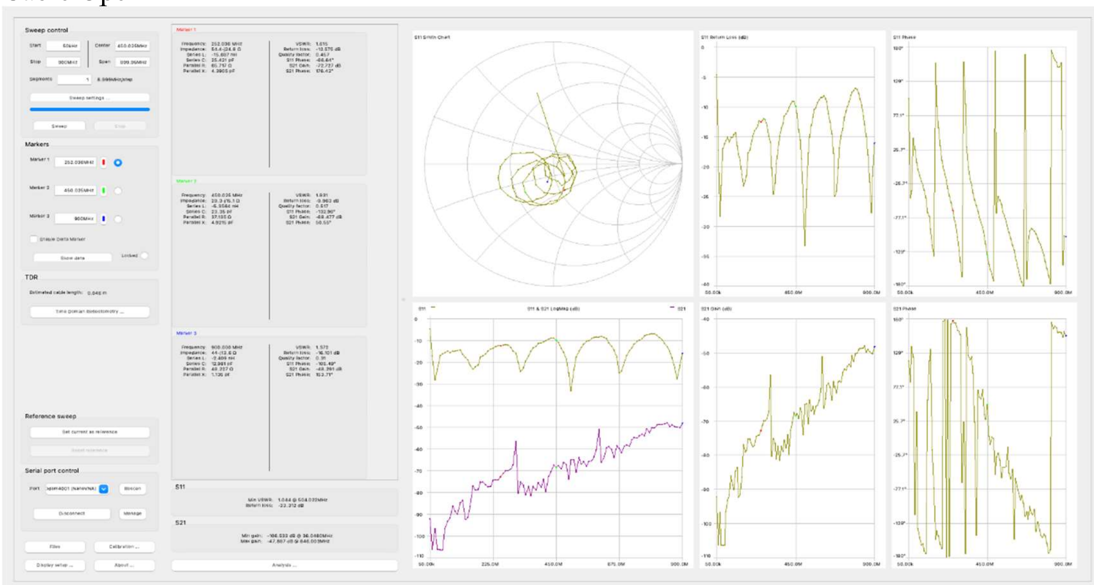

Figure 36: NanoVNA Saver Plots for Cable Open

- Cable Short

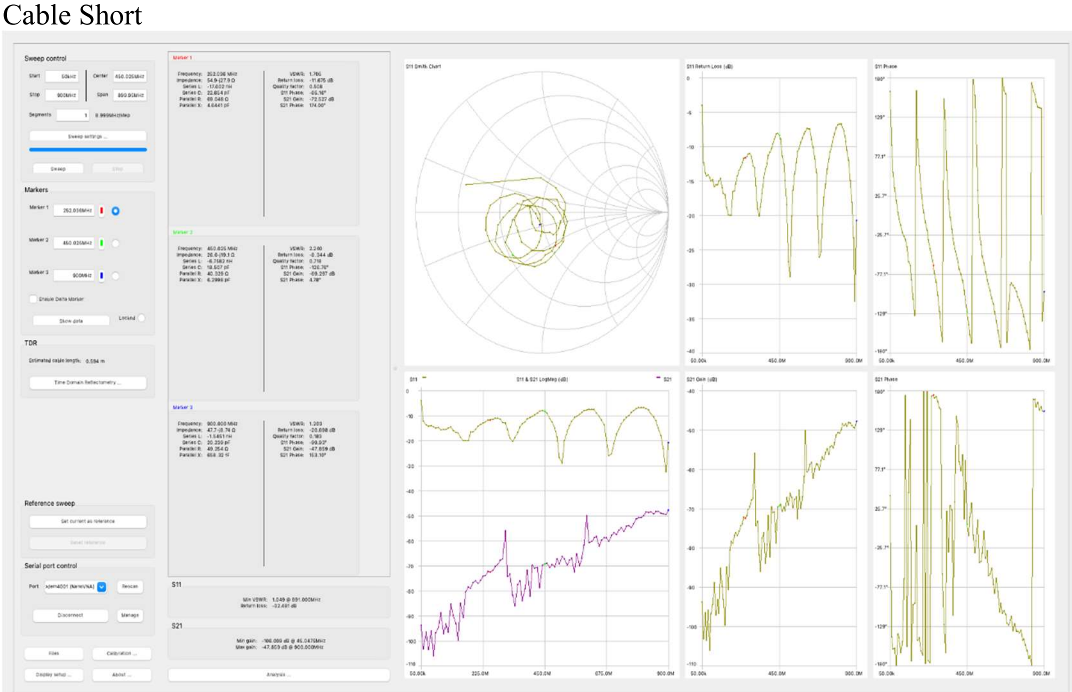

Figure 37: NanoVNA Saver Plots for Cable Short

- Balun Open

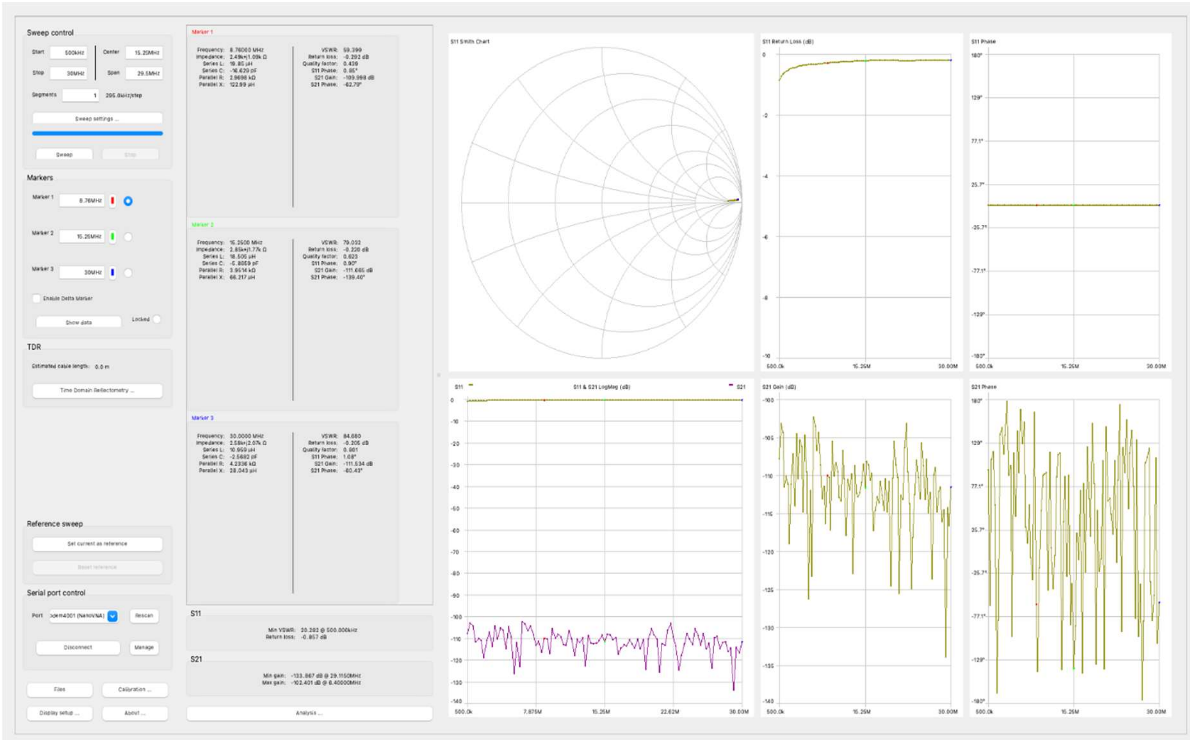

Figure 38: NanoVNA Saver Plots for Balun Open

- Balun Short

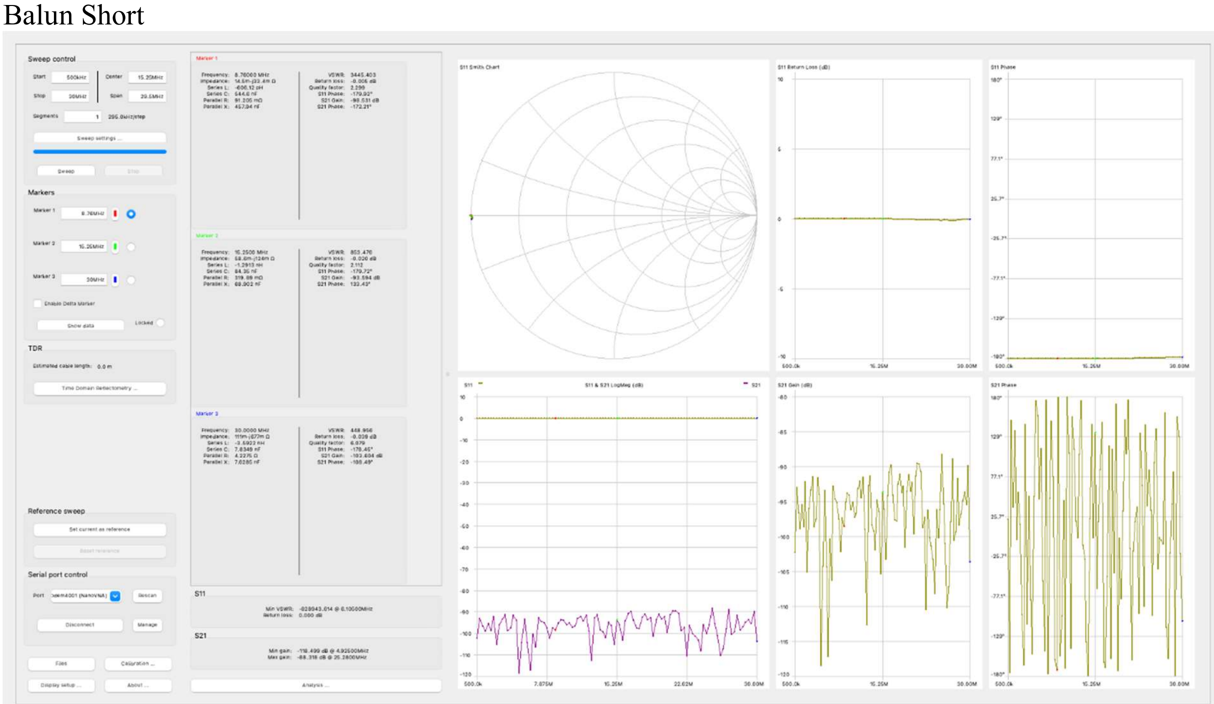

Figure 39: NanoVNA Saver Plots for Balun Short

- Balun Termination

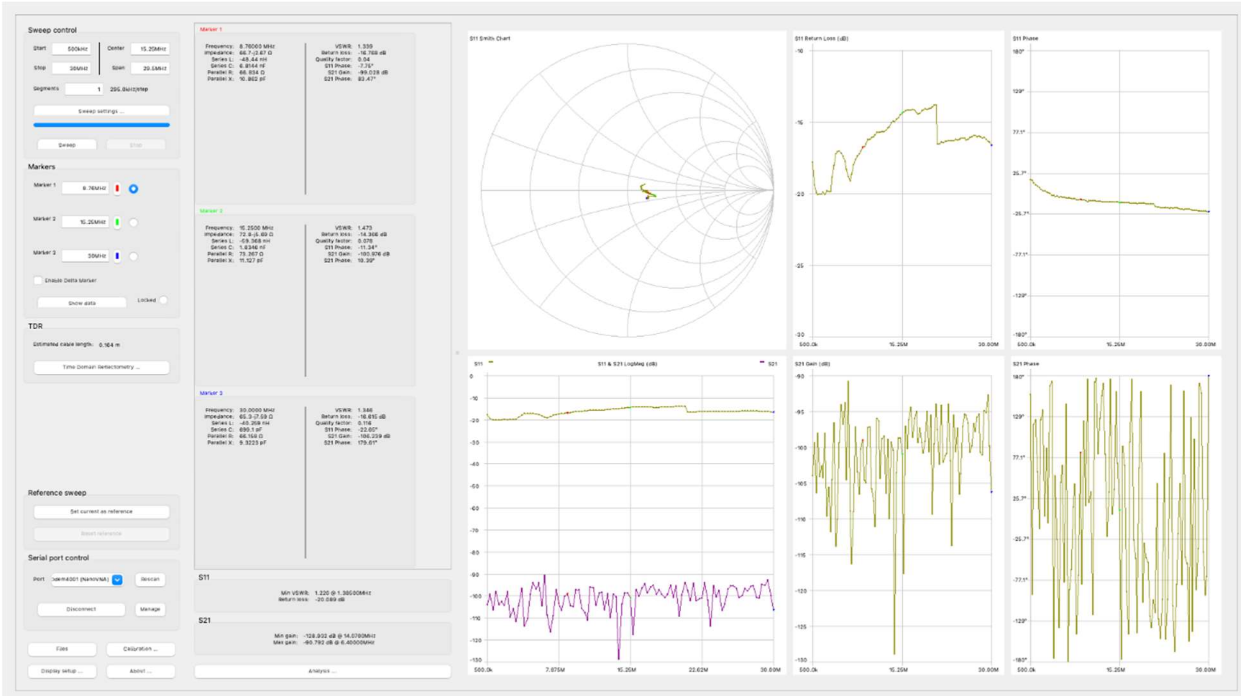

Figure 40: NanoVNA Saver Plots for Balun Termination

Note: A different resistor of 100Ω was used for termination when using NanoVNA Saver as the soldering of the resistor provided was broken.

It can be seen from all the plots involved that all of them seem fairly similar for their corresponding measurements.

The measurements can be summarized in the following table:

Parameters	Impedance Value [Ω]
Cable Open (C_{open})	72
Cable Short (C_{short})	101.5
Balun Open (B_{open})	171
Balun Short (B_{short})	$117 \cdot 10^{-9}$
Balun Termination ($B_{termination}$)	49.9

Table 1: Impedance measurements

- **Characteristic Impedance (Z_o)**
  From the impedance measurements of the cable one can calculate the characteristic impedance by:

$$Z_o = \sqrt{C_{open} \cdot C_{short}} \cdot e^{jn\pi} = \sqrt{\frac{R+j}{G+j\omega C}} = \sqrt{\frac{L}{C}}\sqrt{\frac{1-j\frac{R}{\omega L}}{1-j\frac{G}{\omega C}}}$$

$$|Z_o| = 86\ \Omega$$

$$\measuredangle Z_o = 0$$

- **ABCD Parameters for Balun**

  For the balun three measurements were performed and the ABCD parameters can be determined from these measurements:

$$A = B_{open} \cdot \sqrt{\frac{(B_{short} - B_{termination})}{75 \cdot (B_{termination} - B_{open}) \cdot (B_{open} - B_{short})}}$$

$$A = 171 \cdot \sqrt{\frac{(117 \cdot 10^{-9} - 4\ \ .9)}{75 \cdot (49.9 - 171) \cdot (171 - 117 \cdot 10^{-9})}}$$

$$A = 0.97$$

$$B = B_{short} \cdot \sqrt{\frac{75 \cdot (B_{termination} - B_{open})}{(B_{short} - B_{termination}) \cdot (B_{open} - B_{short})}}$$

$$B = 117\ \cdot 10^{-9} \cdot \sqrt{\frac{75 \cdot (49.9 - 171)}{(117 \cdot 10^{-9} - 49.9) \cdot (171 - 117 \cdot 10^{-9})}}$$

$$B = 1.21 \cdot 10^{-7}\ \Omega$$

$$C = \sqrt{\frac{(B_{short} - B_{termination})}{75 \cdot (B_{termination} - B_{open}) \cdot (B_{open} - B_{short})}}$$

$$C = \sqrt{\frac{(117 \cdot 10^{-9} - 49.9)}{75 \cdot (49.9 - 171) \cdot (171 - 117 \cdot 10^{-9})}}$$

$$C = 5.67 \cdot 10^{-3}\ \Omega^{-1}$$

$$D = \sqrt{\frac{75 \cdot (B_{termination} - B_{open})}{(B_{short} - B_{termination}) \cdot (B_{open} - B_{short})}}$$

$$D = \sqrt{\frac{75 \cdot (49.9 - 171)}{(117 \cdot 10^{-9} - 49.9) \cdot (171 - 117 \cdot 10^{-9})}}$$

$$D = 1.03$$

  Finally, the ABCD matrix is given by:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 0.97 & 1.21 \cdot 10^{-7}\ \Omega \\ 5.67 \cdot 10^{-3}\ \Omega^{-1} & 1.03 \end{bmatrix}$$

- **Secondary Parameters:**

  For the secondary parameters, firstly Gamma γ is calculated which is then used to determine alpha and beta.

$$\gamma l = atanh\left(\sqrt{\frac{Z_s}{Z_o}} \cdot e^{jn\pi}\right) + jm\pi$$

  Assumption: $l = 1$, $n = 0$ and $m = 1$. Then, the equation reduces to:

$$\gamma = atanh\left(\sqrt{\frac{C_{short}}{C_{open}}} \cdot e^{j0\pi}\right) + j\pi = atanh\left(\sqrt{\frac{46.6}{53.8}} \cdot e^{j0\pi}\right) + j\pi$$

$$\gamma = 1.66 + 3.14\, j$$

  Giving the values:

$$\alpha\ (alpha) = Re\{\gamma\} = 1.66$$

$$\beta\ (beta) = Im\{\gamma\} = 3.14$$

- **MATLAB calculation**
  All the results were calculated from MATLAB with the following code:

```
% Constants and Measurement Readings
n = 0;
m = 1;
Z_open = 171 ;
Z_short = 117 * 10^(-9);
Z_termination = 49.9;
Z_C_short = 101.5 ;
Z_C_open = 72;

% ABCD Paramter Calculation
A = Z_open.*sqrt((Z_short-Z_termination)./(75*(Z_termination-Z_open).*(Z_open-Z_short)));
B = Z_short.*sqrt(75*(Z_termination-Z_open)./((Z_short-Z_termination).*(Z_open-Z_short)));
C = sqrt((Z_short-Z_termination)./(75*(Z_termination-Z_open).*(Z_open-Z_short)));
D = sqrt(75*(Z_termination-Z_open)./((Z_short-Z_termination).*(Z_open-Z_short)));

% Characterisitc Impedance Calculation
Z_w = sqrt((Z_C_short.*Z_C_open))*exp(1i*n*pi);
Magnitude = abs(Z_w); % Magnitude of Characterisitc Impedance
Phase = angle(Z_w); % Phase of Characterisitc Impedance

% Secondary line parameters
gamma = atanh(sqrt(Z_C_short/Z_C_open).*exp(1i*n*pi))+1i*m*pi;
alpha = real(gamma);
beta = imag(gamma);

fprintf('A = %f\n', A);
fprintf('B = %d\n', B);
fprintf('C = %d\n', C);
fprintf('D = %f\n', D);
fprintf('Magnitude Characteristic Impedance |Z_w| = %f\n', Magnitude);
fprintf('Phase of Z_w = %d\n', Phase);
fprintf('Alpha = %f\n', alpha);
fprintf('Beta = %f\n', beta);
```

- OUTPUT

```
A = 0.969272

B = 1.207092e-07

C = 5.668257e-03

D = 1.031702

Magnitude Characteristic Impedance |Z_w| = 85.486841

Phase of Z_w = 0

Alpha = 1.663470

Beta = 3.141593
```

- **Professional VNA Measurements**
  The data for cable open, cable short, balun open, balun short and balun termination were provided as a ".asc" file data. One measurement was Keysight Ethernet using own balun and another was ZVR 0.4mm TP using North Hills balun. The first column of every file had the frequency division, the second column had the real part of the impedance while the third column accounted for the imaginary component. The data were read in the MATLAB accordingly. These data were used to calculate the parameters of both the balun

and cables. The formulas were then followed from the lab manual mediums to determine the different responses including: amplitude and phase response of the characteristic impedance.
For the Keysight 10001 data were used and for ZVR 0.4 mm TP 2001 data were used.
The MATLAB script used was:

```
% reading 10001 data points from their respective files and storing them.
data_cab_op = dlmread('CABLE_O.ASC','','B1..B10001')...
+ 1i*dlmread('CABLE_O.ASC','','C1..C10001');      % reading the cable_open file given in real
and imaginary form
data_cab_sh = dlmread('CABLE_S.ASC','','B1..B10001')...
  + 1i*dlmread('CABLE_S.ASC','','C1..C10001');    % reading the cable_short file given in real
and imaginary form

f = dlmread('BALUN_O.ASC','','A1..A10001'); % reading the frequency from the given data ASC
files
data_op = dlmread('BALUN_O.ASC','','B1..B10001')...
+ 1i*dlmread('BALUN_O.ASC','','C1..C10001');   % reading the balun_open file given in real and
imaginary form
data_sh = dlmread('BALUN_S.ASC','','B1..B10001')...
+ 1i*dlmread('BALUN_S.ASC','','C1..C10001');    % reading the balun_short file given in real
and imaginary form
data_term = dlmread('BALUN_T.ASC','','B1..B10001')...
+ 1i*dlmread('BALUN_T.ASC','','C1..C10001');     % reading the balun_termination file given in
real and imaginary form

% Balun Impedance calculations with 75 ohm termination
Z_open = 75*(1+data_op)./(1-data_op); % calculating open impedance for balun
Z_short = 75*(1+data_sh)./(1-data_sh); % calculating short impedance for balun
Z_termination = 75*(1+data_term)./(1-data_term); % calculating termination impedance for balun

% ABCD parameters calculations
A = Z_open.*sqrt((Z_short-Z_termination)./(75*(Z_termination-Z_open).*(Z_open-Z_short)));
B = Z_short.*sqrt(75*(Z_termination-Z_open)./((Z_short-Z_termination).*(Z_open-Z_short)));
C = sqrt((Z_short-Z_termination)./(75*(Z_termination-Z_open).*(Z_open-Z_short)));
D = sqrt(75*(Z_termination-Z_open)./((Z_short-Z_termination).*(Z_open-Z_short)));

% Impedances for Cable
Z_C1_open = 75*(1+data_cab_op)./(1-data_cab_op);
Z_C1_short = 75*(1+data_cab_sh)./(1-data_cab_sh);
Z_C2_open = (B-D.*Z_C1_open)./(C.*Z_C1_open-A);
Z_C2_short = (B-(D.*Z_C1_short))./((C.*Z_C1_short)-A);
n = 0;
m = 1;

% Other Parameters
Z_w = sqrt((Z_C2_short.*Z_C2_open))*exp(1i*n*pi); % characteristic impedance calculation
gamma = atanh(sqrt(Z_C2_short./Z_C2_open).*exp(1i*n*pi))+1i*m*pi; % gamma calculation
alpha = real(gamma);
beta = imag(gamma);

% Plots for Amplitude and Phase of Characteristic Impedance, Attenuation and Beta
figure (1);
sgtitle('Plots for Keysight Ethernet');
subplot(2,2,1);
semilogx(f,abs(Z_w)); % magnitude plot for characterisitc impedance
title('Amplitude Response of Characteristic Impedance');
xlabel('Frequency [Hz])');
ylabel('Magnitude |Z_w|');
xlim([10e3 10e6]);
ylim([0 1000]);
```

```
subplot(2,2,2);
semilogx(f,angle(Z_w)*(180/pi)); % phase plot for characteristic impedance in radians
title('Phase Response of Characteristic Impedance');
xlabel('Frequency[Hz]');
ylabel('Phase Z_w');
xlim([10e3 10e6]);
ylim([0 1000]);


subplot(2,2,3);
alpha_Np = alpha*(20/log(10)); % Neper Conversion
semilogx(f,alpha); % plot for alpha
title('Attenuation in the cable');
xlabel('Frequency [Hz])');
ylabel('Magnitude of Alpha [Np]');
xlim([10e3 10e6]);

subplot(2,2,4);
semilogx (f,beta); % plot for beta
title('Beta');
xlabel('Frequency [Hz])');
xlim([10e3 10e6]);
ylabel('Phase Constant [rad/m]');
```

- Plot for Keysight Ethernet Data

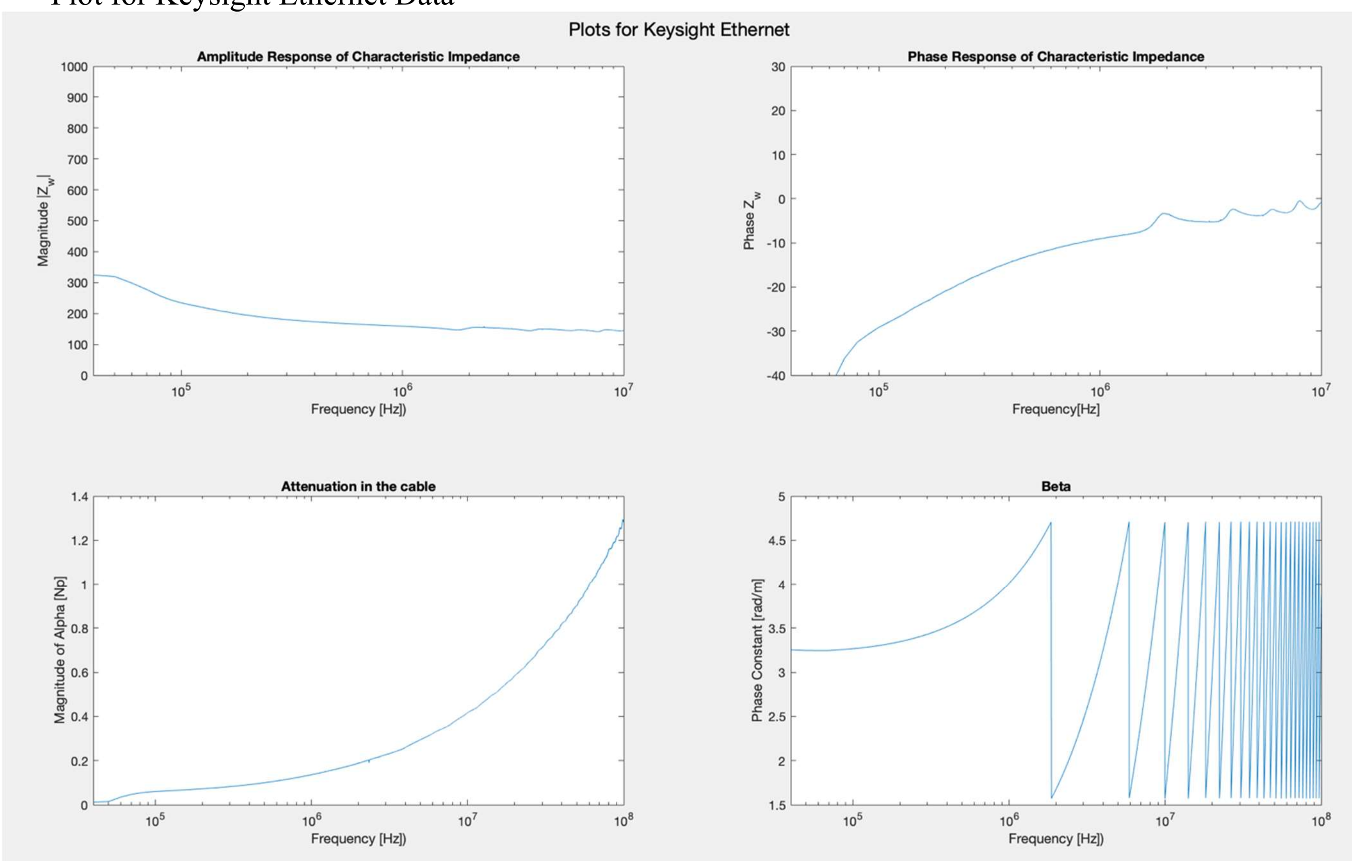

Figure 41: Plots for Amplitude, Phase Response and Alpha, Beta Constants for Keysight Ethernet

- Plots for ZVR 0.4 mm TP

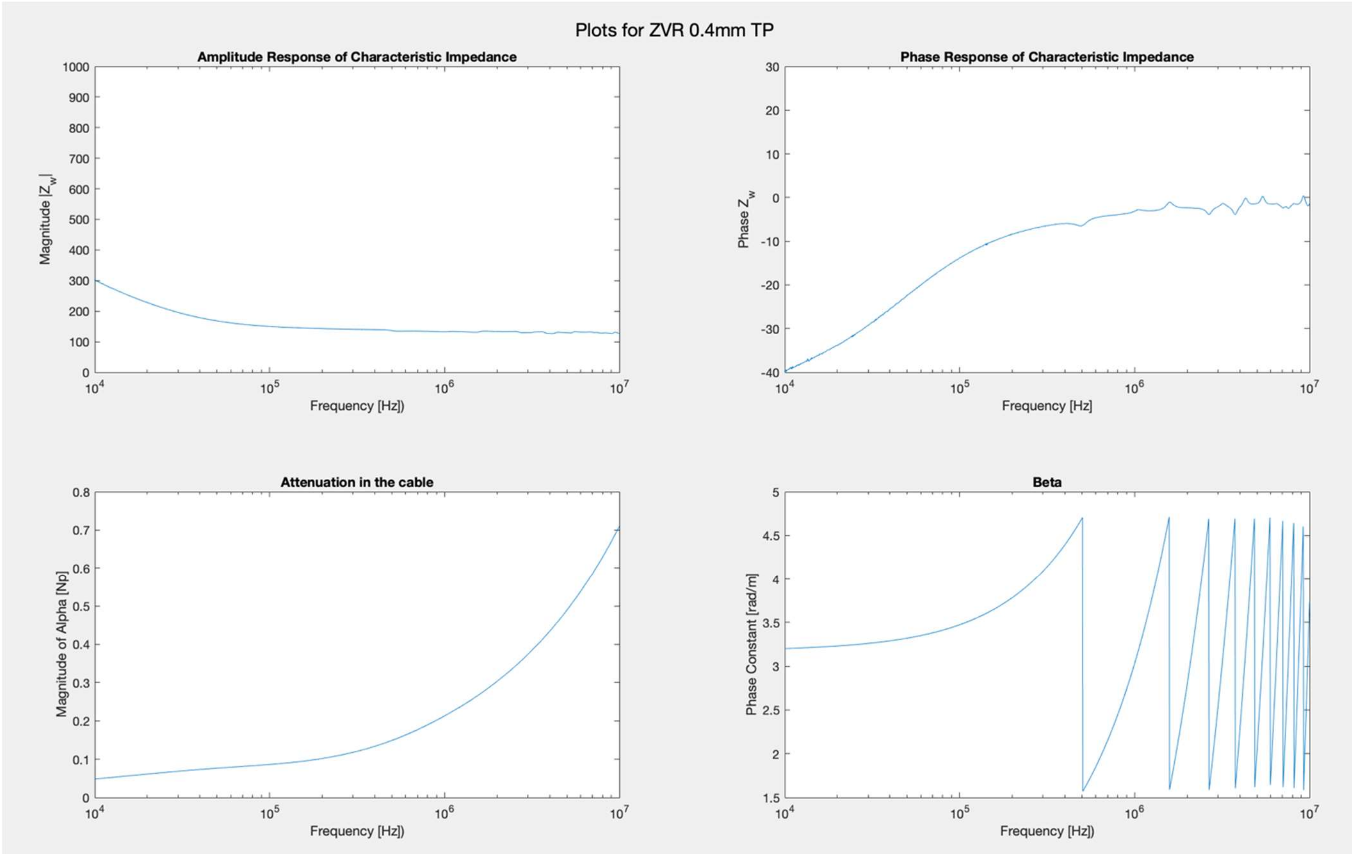

Figure 42: Plots for Amplitude, Phase Response and Alpha, Beta Constants for ZVR 0.4mm TP

- It can be seen from the plots that the calculated data from the Nano VNA are similar with the parameters calculated from the data provided. The reason for slight deflection in characteristic impedance could be due to the reflection coefficient not being equal to 1.

3 Conclusion:

In this lab module, we learned how to use MATLAB to change and manipulate parameters and data to measure different properties of cables, as well as how to remotely control different measuring devices. In addition, LabView was used with MATLAB algorithms in the form of blocks to examine the devices, giving a broader understanding of their functions. Moreover, the functionalities of NanoVNA were also studied in detail. The NanoVNA was used to achieve the open, short, and termination measurements for cables and baluns, with the output smith chart obtained. The ABCD matrix, Characteristic Impedance, and Secondary Line Parameters were all calculated after that.

Finally, the 10001 data for professional VNA was provided, which was used to generate the ABCD matrix, the Amplitude and Phase of Characteristic Impedance, and Secondary Parameters Alpha and Beta. However, during the lab session, we came across some issues, the most serious of which was an equipment malfunction. Because we had to control the device remotely, the devices occasionally did not respond well due to a faulty wire connection, and the remote PCs were occasionally slow to execute the MATLAB files. With task repetitions, and support from the TA and the professor, we were successfully able to complete the tasks.

4 References:

- http://trsys.jacobs-university.de/login/spec_lab.htm
- https://groups.io/g/nanovna-users/topic/38757935#5894
- https://www.youtube.com/watch?v=zw7Dp1nwvD8
- https://www.youtube.com/watch?v=G66_iqOu-Bs
- https://www.youtube.com/watch?v=zE_24-b2d_Q