Lab Report

Spring 2021
CA11-300303
ECE Specialization Areas Lab
Instructor: Prof. Dr.-Ing. Werner Henkel

Authors: Tallah A. Chowhan

Done by: Haseeb Ahmad/ Tallah Chowhan
Matriculation number: 30002360

Introduction

In this lab we were introduced to transmission line theory; and experiment with remotely controlling a digital scope, network and spectrum analyser, and a function generator by using MATLAB and LabVIEW codes.

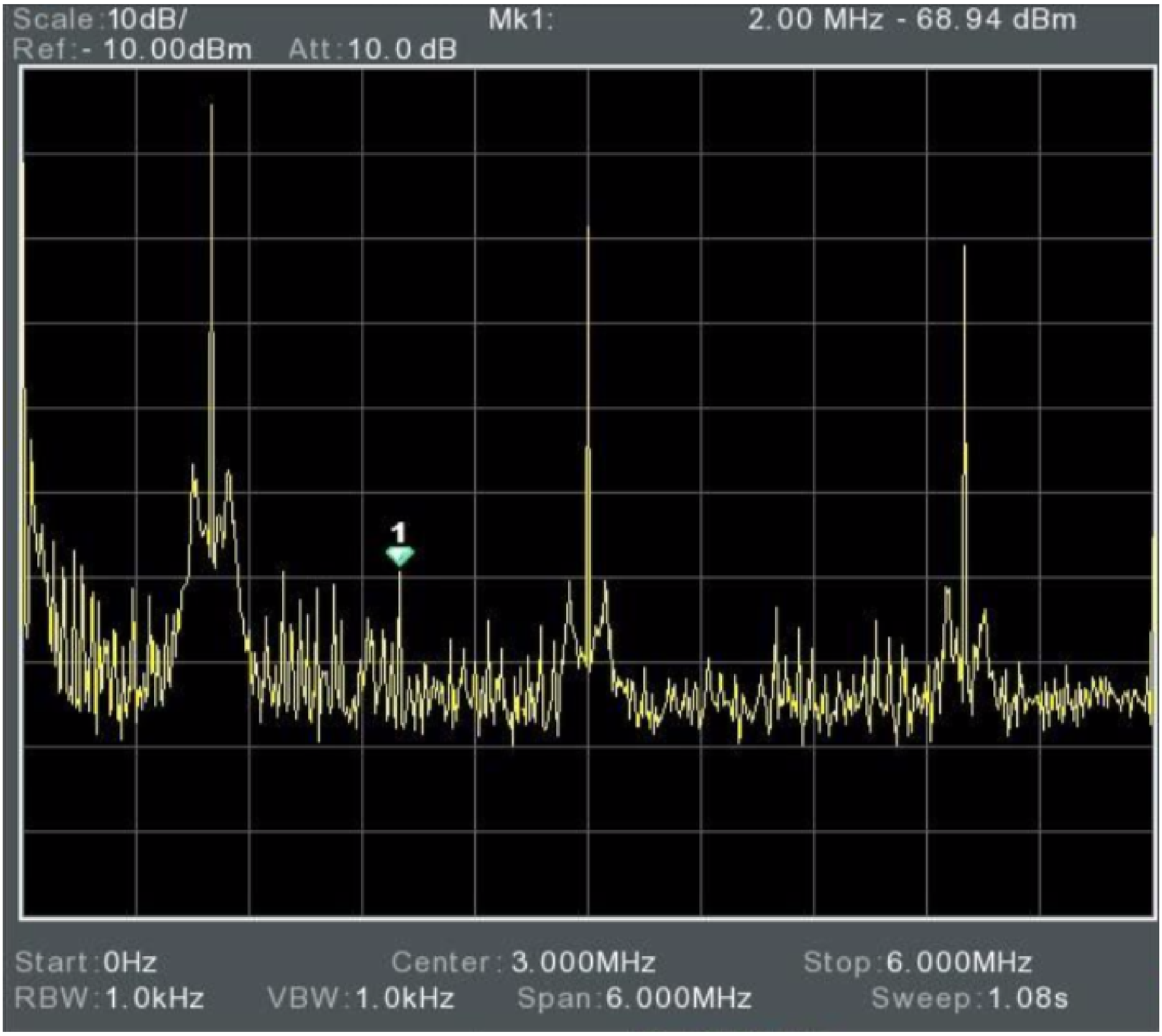
Theory

In this lab we used the below stated instruments to measure the capacitance, inductance and length of a transmission line. These measurements will demonstrate transmission line theory and can also be used to characterize or test transmission lines

Instruments

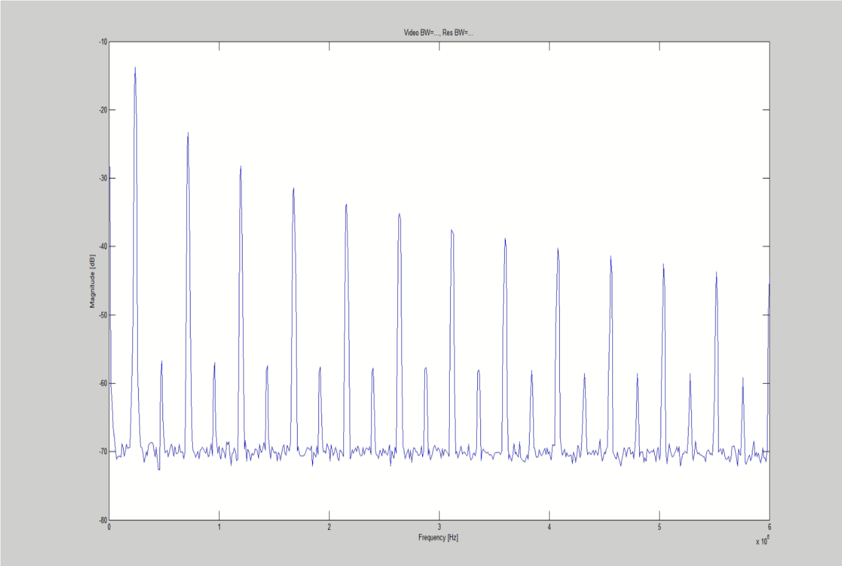
1. Spectrum Analyzer

We used different settings while interacting with the spectrum analyser such as resolution and video bandwidth, and the fast sweep setting which all had different effects on our results. We finally used the video bandwidth results as they were less noisy and more accurate.

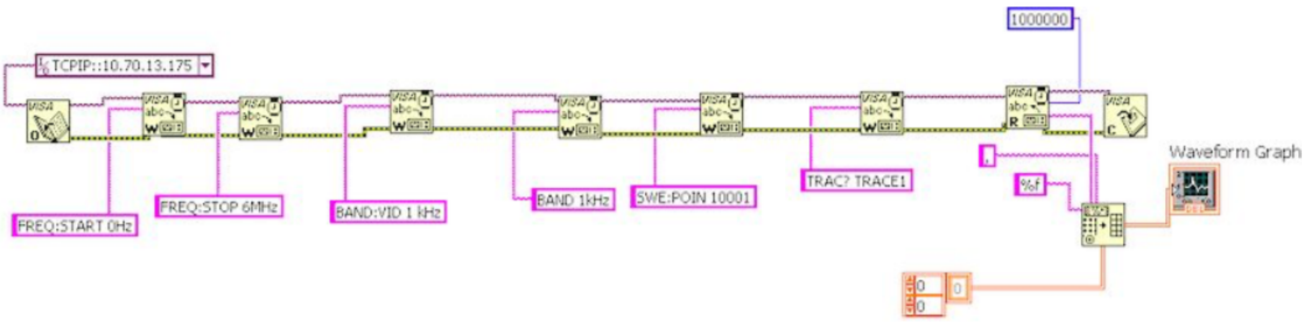

Scale:10dB/
Mk1:
2.00 MHz - 68.94 dBm
Ref:- 10.00dBm
Att:10.0 dB
1
Start:0Hz
Center:3.000MHz
Stop:6.000MHz
RBW:1.0kHz
VBW:1.0kHz
Span:6.000MHz
Sweep:1.08s

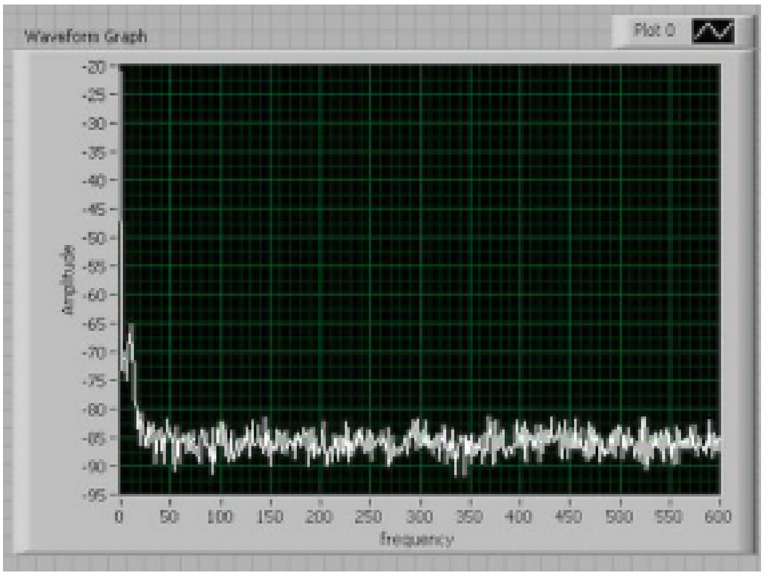
In MATLAB

```
clc; close all; clear all;

h = visa('agilent','TCP IP::10.70.13.175::INSTR'); % check IP

address!!! h.inputbuffersize = 1000000;

fopen(h);

        fprintf(h,'FREQ:STAR 0Hz'); % set start frequency

        fprintf(h,'FREQ:STOP 6MHz'); % set stop frequency

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

        fprintf(h,'BAND 1kHz'); % set resolution bandwidth to 100 kHz

        fprintf(h, 'SWE:POIN 10001\n'); % set number of points to 10001

%fprintf(h,'FORM ASCII');

pause(10)

fprintf(h,'TRAC? TRACE1');

tr = fscanf(h);

fclose(h);

trace = str2num(tr);

%freq = ; % produce frequency axis

freq1= linspace(0,6E6,601) ;

plot(freq1, trace);

xlabel('Frequency [Hz]');

ylabel('Magnitude [dB]');

title('Video BW=..., Res BW=...');
```


Video BW=..., Res BW=...
Magnitude [dB]
Frequency [Hz]
-10
-20
-30
-40
-50
-60
-70
-80
0
1
2
3
4
5
6

In Labview


1000000
TCPIP::10.70.13.175
FREQ:START 0Hz
FREQ:STOP 6MHz
BAND:VID 1 kHz
BAND 1kHz
SWE:POIN 10001
TRAC? TRACE1
%f
Waveform Graph

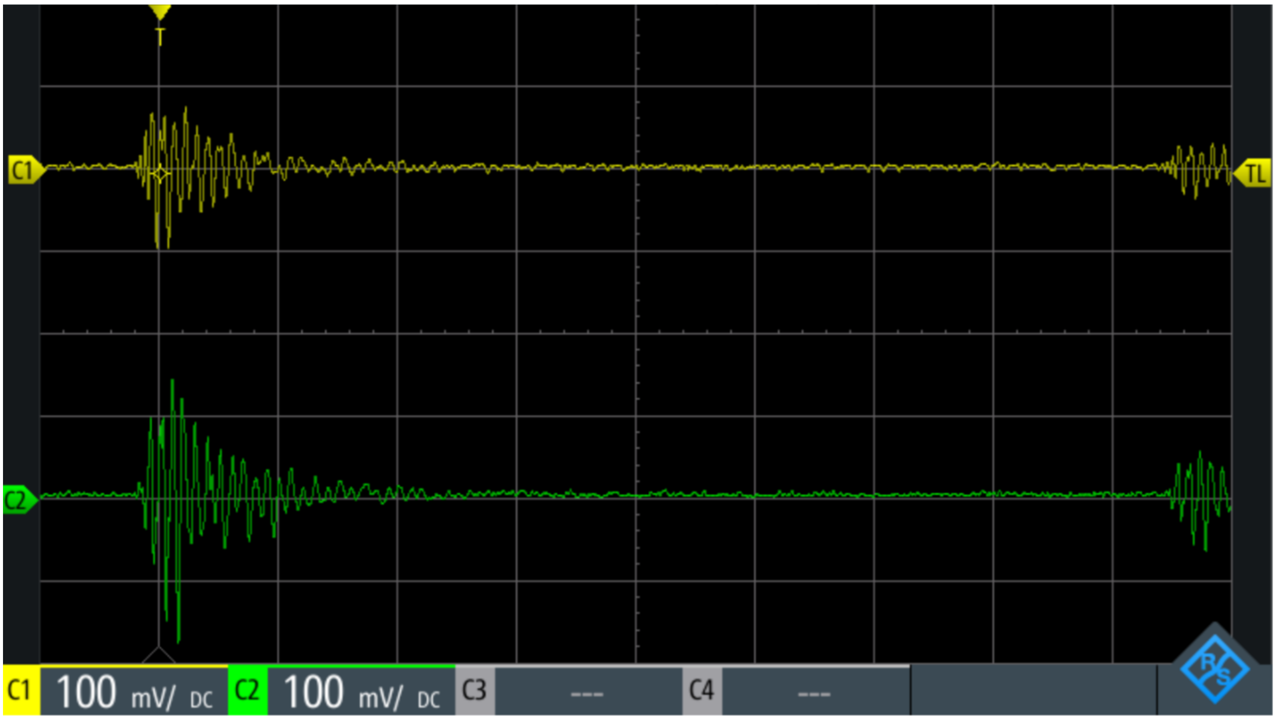

Waveform Graph
Plot 0
Amplitude
-20
-25
-30
-35
-40
-45
-50
-55
-60
-65
-70
-75
-80
-85
-90
-95
0
50
100
150
200
250
300
350
400
450
500
550
600
frequency

2. Oscilloscope

We worked with both R&S Oscilloscope and the LeCroy Scope.

- **Rohde & Schwarz**

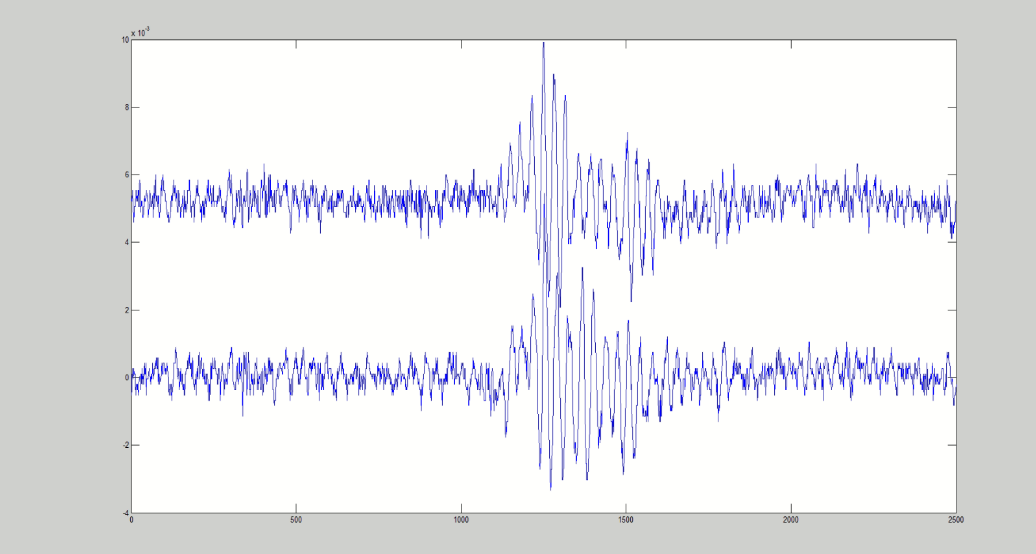
For Rohde & Schwarz Oscilloscope we only took the measurements and then completed and ran the given MATLAB code.



In MATLAB (R&S)

```
clc; clear all; close all;

test_obj=visa('agilent','TCPIP0::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<0.1>'); % Setting the vertical res.

Ch1 fprintf(test_obj, 'CHANnel2:SCALe<0.1>'); % Setting the vertical

res. Ch2

fprintf(test_obj, 'TIMebase:SCALe<5E-9>'); % Setting the time base - Ch1

fprintf(test_obj, 'TRIGger:EDGE:SLOPe<POSitive>'); % Trigger on positive edge

fprintf(test_obj, 'TRIGger:LEVel<2>:VALue <0.01>');% Trigger level fprintf(test_obj,

'TRIGger:SOURce <C2>'); % 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;
```



The averaging mode algorithm works by averaging the points in a single time bucket together across multiple acquisitions. Using the mathematical process of averaging points together, you will effectively get better vertical resolution.To get a finer resolution, you must either sample at a slower rate or increase the record length. If you cannot increase the record length, then you'll have to decrease the sampling rate. Doing this, of course, reduces the frequency range of your FFT analysis.

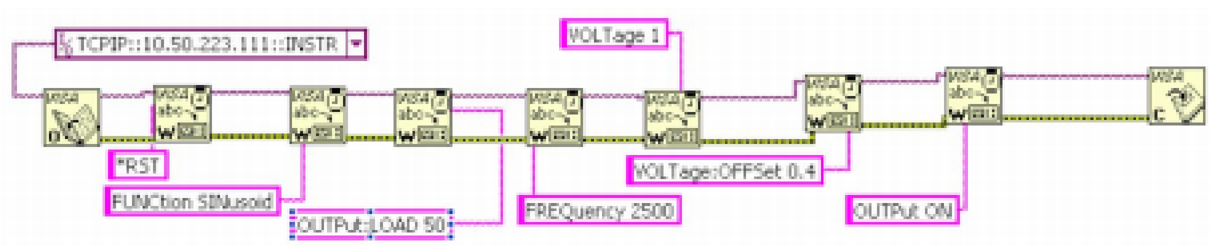
The FFT implemented in an oscilloscope has limited record length. This can cause issues in the spectrum display due to continuity issues at the start and end points of the acquired waveforms. If your signal contains frequencies above the Nyquist point, then the FFT will be inaccurate. In order to avoid this, use a low pass filter to restrict the frequency content of the signal to those frequencies that are below the Nyquist point.

Function Generator

We worked with the Agilent 33220A 20 MHz Waveform Function Generator. **In MATLAB**

```
h = visa('agilent','TCPIP::10.70.13.238::INSTR');
h.inputbuffersize = 1000000;
fopen(h);
%Reset the function generator
fprintf(h, '*RST');
%Select waveshape
fprintf(h,'FUNC SIN');
%Set the load impedance to 50 Ohms (default)
fprintf(h,'OUTPut:LOAD 50');
%Set the amplitude to 100 mV-pp
fprintf(h,'VOLTage 0.1');
%Set log spacing;
fprintf(h,'SWEep:SPACing LOG');
%Set Sweep time 200ms
fprintf(h,'SWEep:TIME 0.2');
%Set Start frequency 1 microHz
fprintf(h,'FREQuency:STARt 1E-6');
%Set Stop frequency 20 MHz
fprintf(h,'FREQuency:STOP 20e6');
%Set type of internal triggering
%fprintf(h,'');
%Turn on the instrument output
fprintf(h,'OUTPut ON');
%Turn sweep on
fprintf(h,'SWEep:STATe ON');
fclose(h);
```

In LABVIEW



Vector Network Analyser (Keysight)

In the lab, we also played with the long wire to get S-Parameter by 5 different methods:

- Cable Short
- Cable Open,
- Balun Open
- Balun Short
- Balun Terminated

We then later went on and simulated the results obtained in a MATLAB to be able to plot the results.

Using MATLAB

```
%taking frequency values from file
f = dlmread('bal_open.txt',';','A2..A2002');
S11_bal_op = dlmread('bal_open.txt',';','B2..B2002') ...
 +1i*dlmread('bal_open.txt',';','C2..C2002');
S11_bal_sh = dlmread('bal_short.txt',';','B2..B2002') ...
 +1i*dlmread('bal_short.txt',';','C2..C2002');
S11_bal_term = dlmread('bal-term.txt',';','B2..B2002') ...
 +1i*dlmread('bal-term.txt',';','C2..C2002');
S11_cabopen = dlmread('cabopen.txt',';','B2..B2002') ...
 +1i*dlmread('cabopen.txt',';','C2..C2002');
S11_cabshort = dlmread('cabshort.txt',';','B2..B2002') ...
 +1i*dlmread('cabshort.txt',';','C2..C2002');
Z0_op = 75*(1+S11_bal_op)./(1-S11_bal_op); %open impedance
Z0_sh = 75*(1+S11_bal_sh)./(1-S11_bal_sh); %short impedance
Z0_term = 75*(1+S11_bal_term)./(1-S11_bal_term); %terminated impedance
%ABCD parameters calculations
%Wireless Communications Book Page 9 (1.27,1.28,1.29,1.30)
A=Z0_op.*sqrt((Z0_sh-Z0_term)./(50*(Z0_term-Z0_op).*(Z0_op-Z0_
short)));
B=Z0_sh.*sqrt(50*(Z0_term-Z0_op)./((Z0_sh-Z0_term).*(Z0_op-Z0
_short)));
C=sqrt((Z0_sh-Z0_term)./(75*(Z0_term-Z0_op).*(Z0_op-Z0_sh)));
D=sqrt(75*(Z0_term-Z0_op)./((Z0_sh-Z0_term).*(Z0_op-Z0_sh)));
%Z1 and Z2(using Z1) calculations of cable using Wireline Communications Book
Z1_short = 75*(1+S11_cabshort)./(1-S11_cabshort);
Z1_open = 75*(1+S11_cabopen)./(1-S11_cabopen);
Z2_short = (B-D.*Z1_short)./(C.*Z1_short-A);
Z2_open = (B-D.*Z1_open)./(C.*Z1_open-A);
n = 0; %for alpha
m = 1; %for gamma
gamma = atanh(sqrt(Z2_short./Z2_open).*exp(1i*n*pi))+1i*m*pi;
alpha = real(gamma); beta = imag(gamma);
Z_w = sqrt(Z2_short.*Z2_open)*exp(1i*n*pi); %characteristic impedance
%Plots for Z_w magnitude, phase, Alpha and Beta
```

```
figure (1)
semilogx(f,abs(Z_w)) %magnitude plot for Z_w
title('Characteristic Impedance Magnitude')
xlabel('Frequency(Hz)')
xlim([10e3 10e7])
ylabel('Magnitude of Z_w')
figure (2)
semilogx(f,angle(Z w)*(180/pi)) %phase v Z_w
title('Characteristic Impedance Phase')
xlabel('Frequency(Hz)')
xlim([10e3 10e7])
ylabel('Phase of Z_w( )')
alpha_Np = alpha*(20/log(10)); %Neper Conversion
figure (3)
semilogx(f,alpha_Np) %plot for alpha
title('Attenuation')
xlabel('Frequency(Hz)')
xlim([10e3 10e7])
ylabel('Magnitude of Alpha(Np)')
figure (4)
semilogx(f,beta) %plot for beta
title('Beta')
xlabel('Frequency(Hz)')
xlim([10e3 10e7])
ylabel('Phase Constant(rad/m)')
```

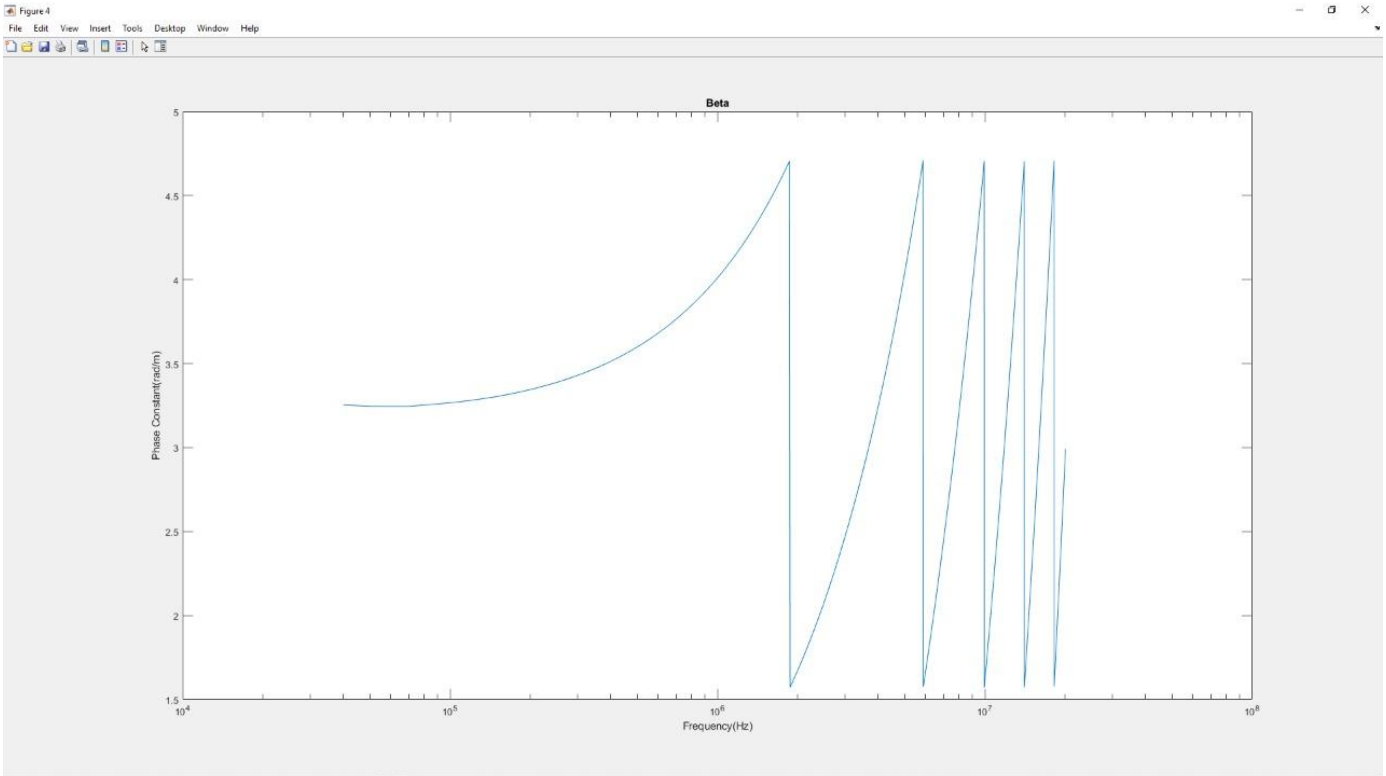


Figure 4
File Edit View Insert Tools Desktop Window Help
Beta
Phase Constant(rad/m)
Frequency(Hz)

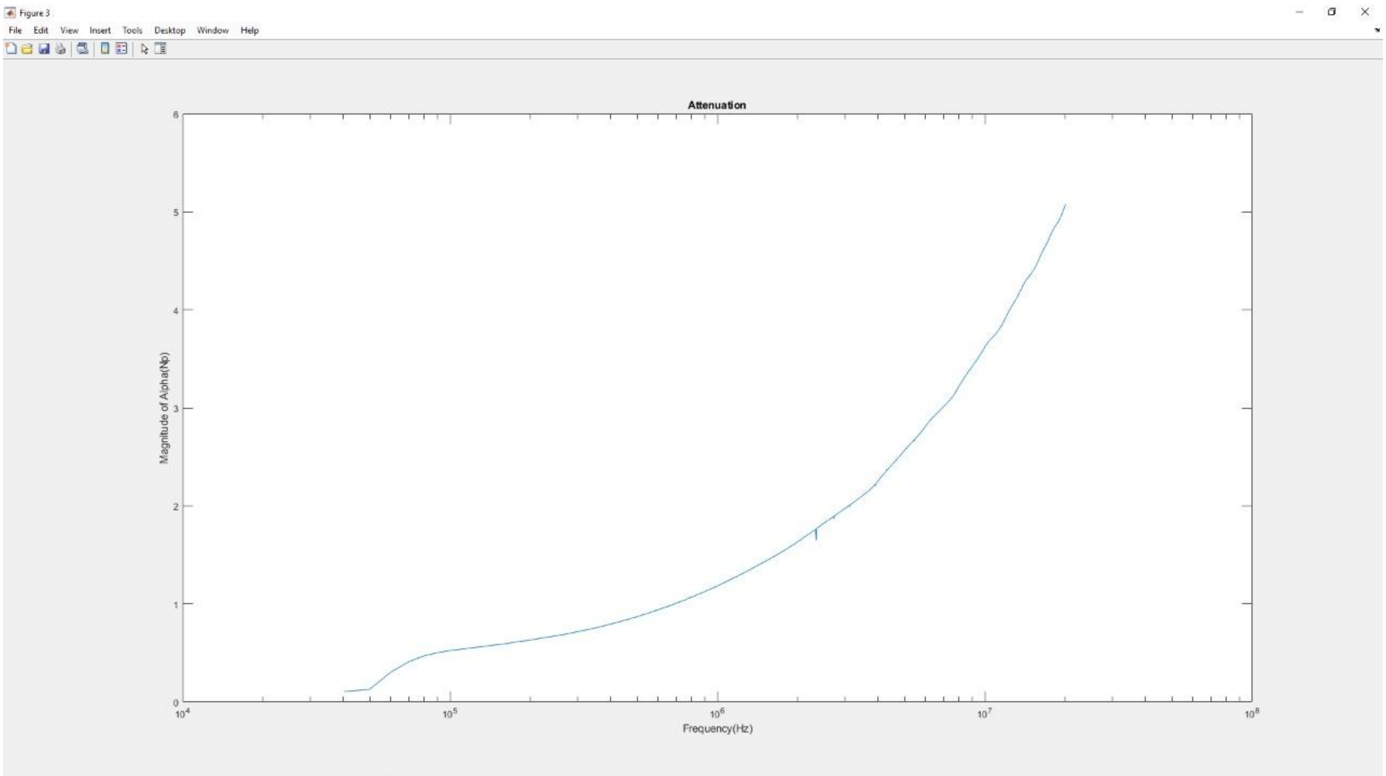


Figure 3
File Edit View Insert Tools Desktop Window Help
Attenuation
Magnitude of Alpha(Np)
Frequency(Hz)

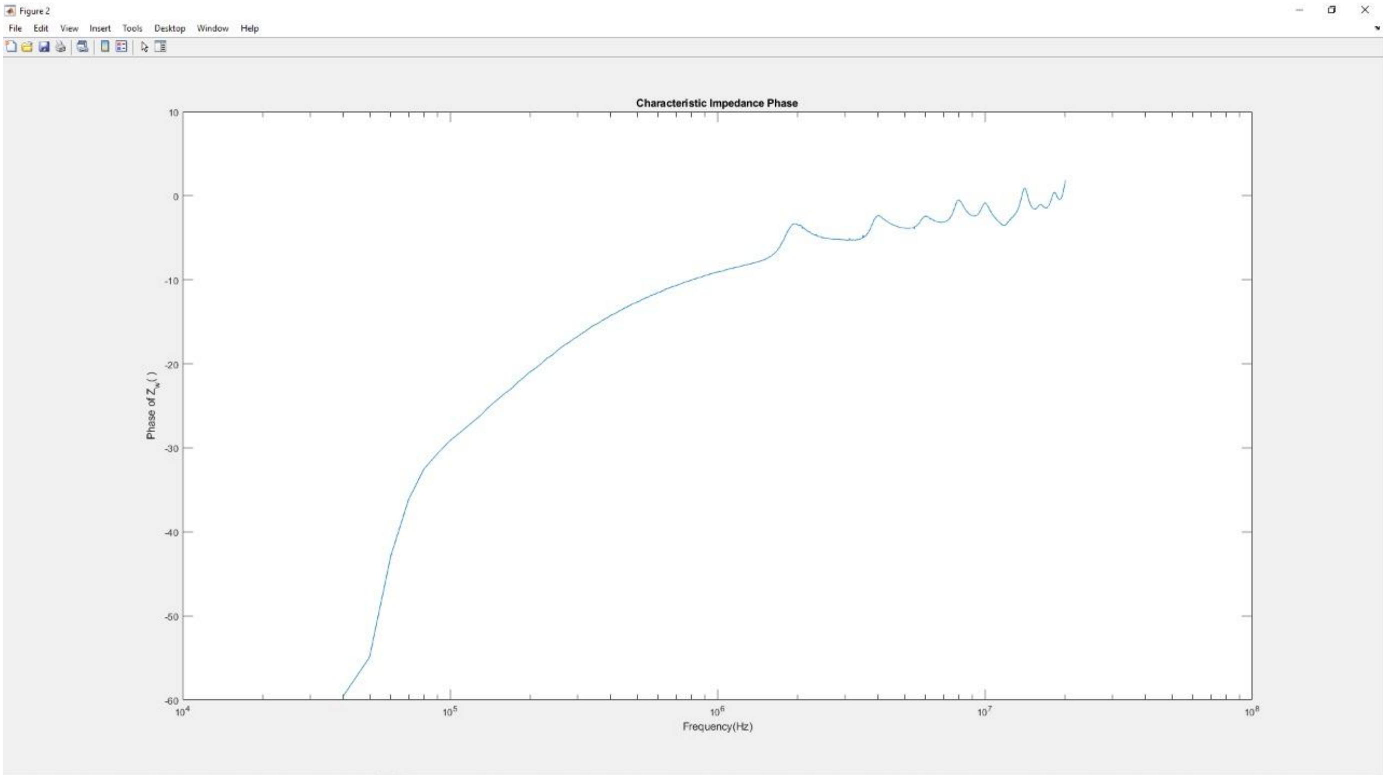


Figure 2
File Edit View Insert Tools Desktop Window Help
Characteristic Impedance Phase
Phase of Z_w ( )
Frequency(Hz)
10
0
-10
-20
-30
-40
-50
-60
10^4
10^5
10^6
10^7
10^8

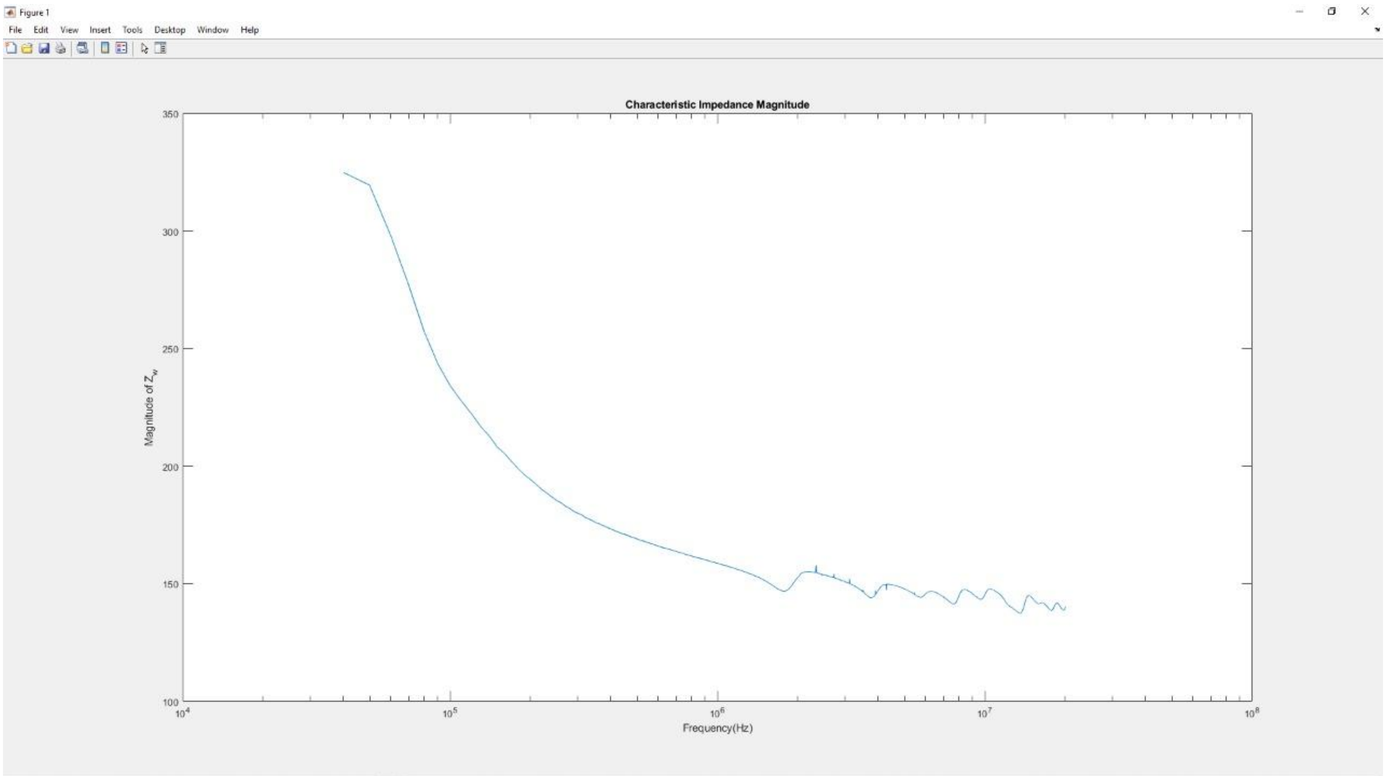
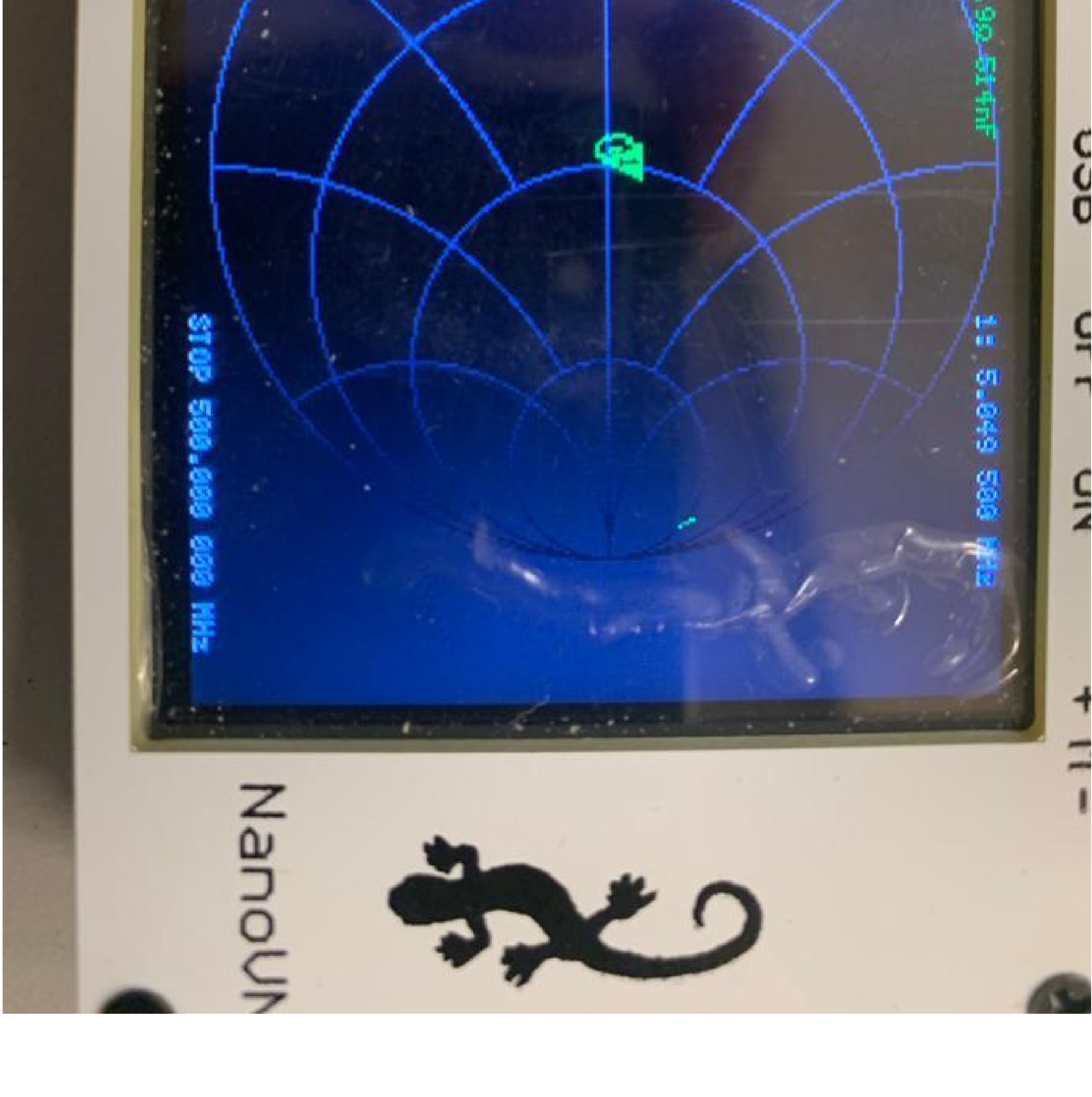
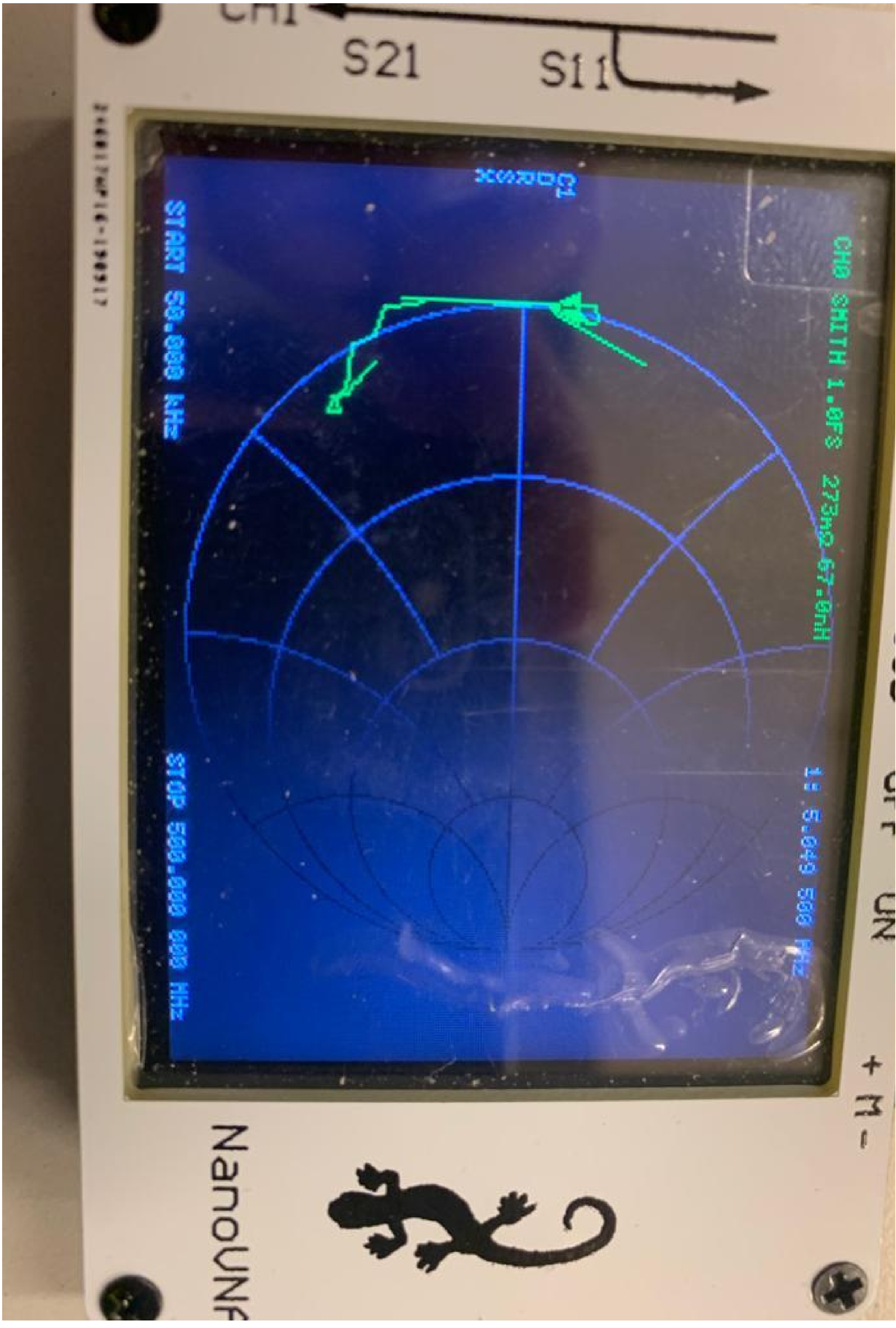


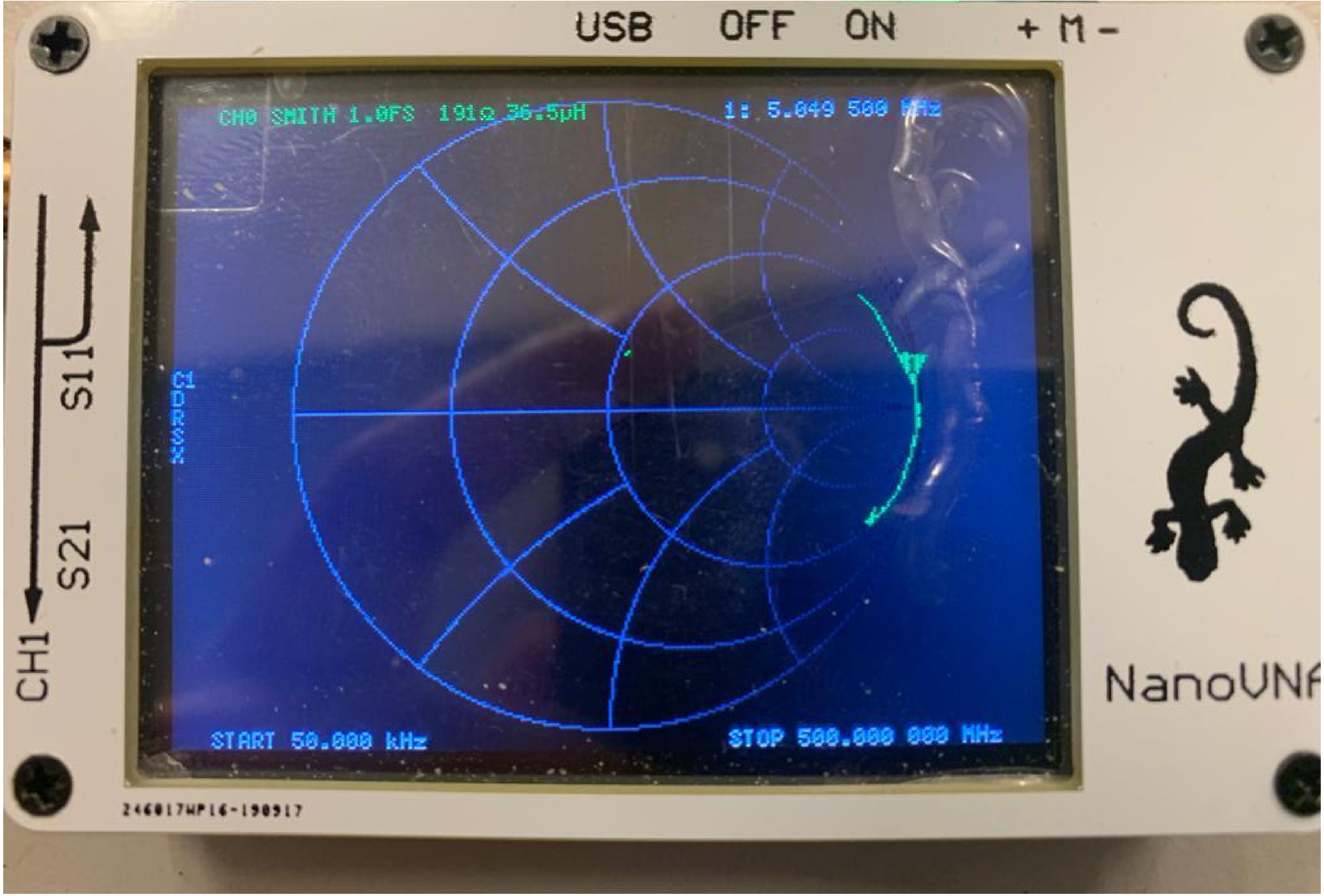
Figure 1
File Edit View Insert Tools Desktop Window Help
Characteristic Impedance Magnitude
Magnitude of Z_w
Frequency(Hz)
350
300
250
200
150
100
10^4
10^5
10^6
10^7
10^8

Using Physical Cable Measurement

- Transformer Measurements

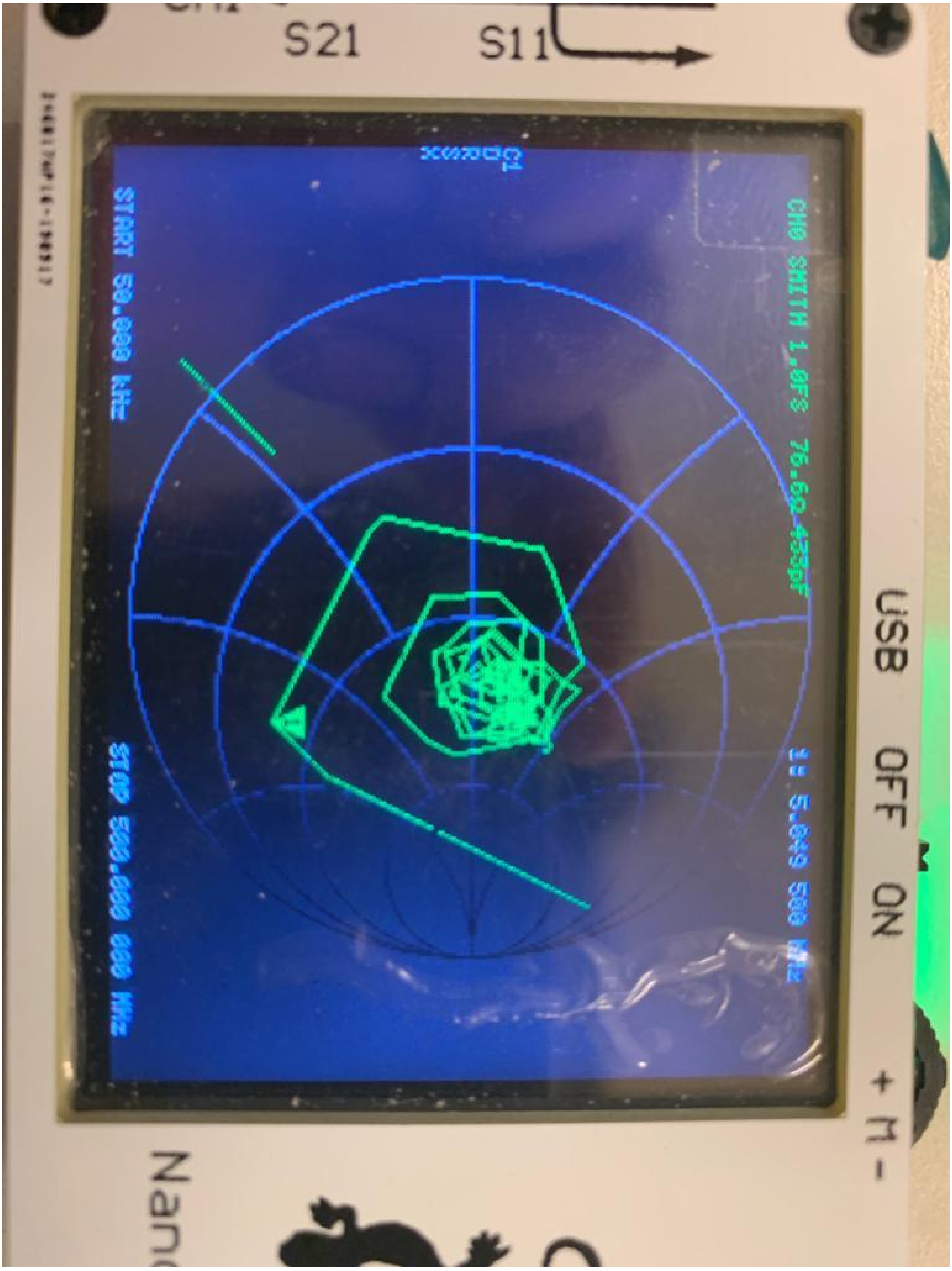

CH1
S21
S11
CH0 SMITH 1.0FS
START 50.000 kHz
STOP 500.000 000 MHz


CH1
S21
S11
C1
CH0 SMITH 1.0FS 273mΩ 67.0nH
START 50.000 kHz
STOP 500.000 000 MHz
1: 5.049 500 MHz


USB
OFF
ON
+ M -
CH0 SMITH 1.0FS 191Ω 36.5pH
1: 5.049 500 MHz
C1
D
R
S
X
START 50.000 kHz
STOP 500.000 000 MHz
CH1
S21
S11
NanoVNA
246017WP16-190917

- LAN Measurements


USB OFF ON
+ M -
CH0 SMITH 1.0FS 65.5Ω 1.66μH
1: 5.049 500 MHz
START 50.000 kHz
STOP 500.000 000 MHz
Nano



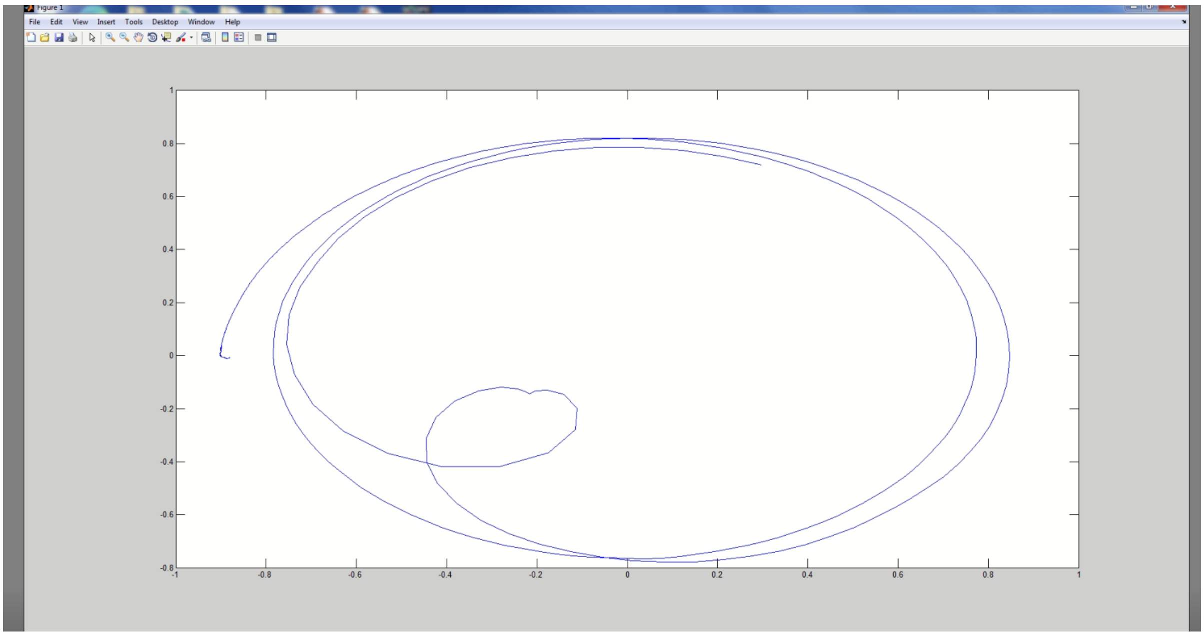
S21
S11
CH0 SMITH 1.0FS 76.6Ω 435pF
START 50.000 kHz
STOP 500.000 000 MHz
USB
OFF
ON

After configuring the setting of the Vector Network Analyzer via MATLAB

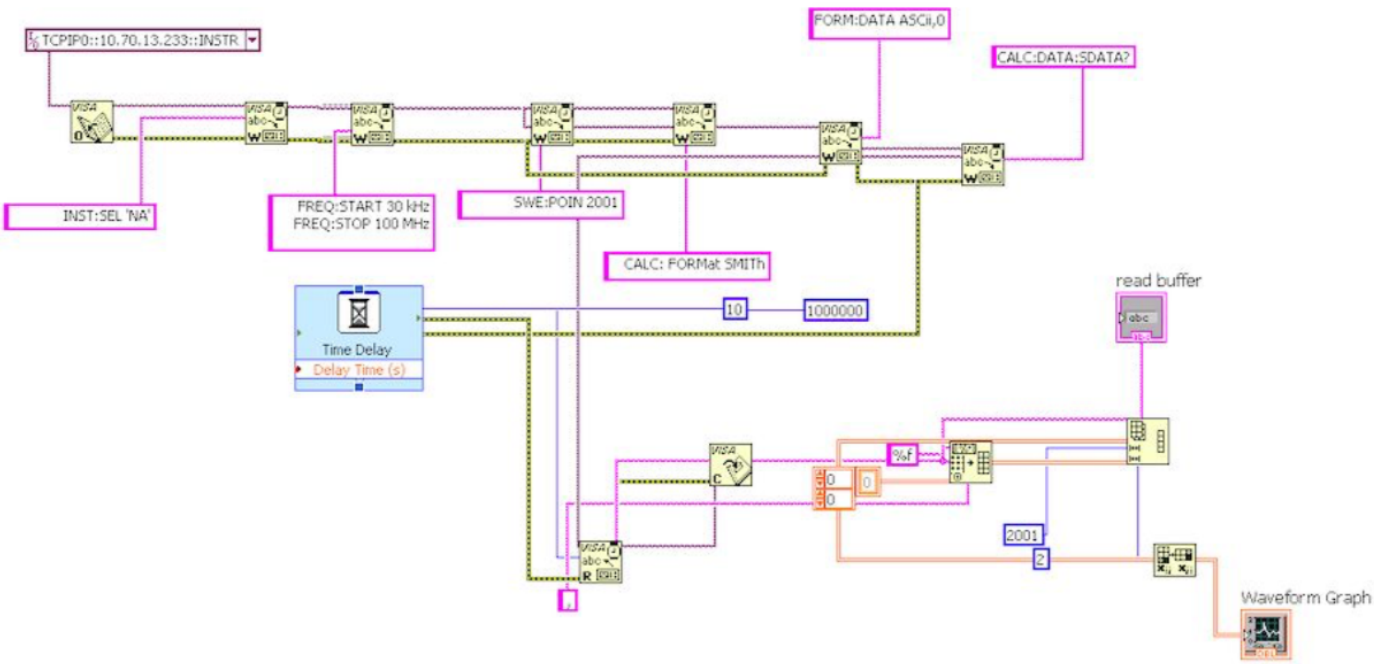
In MATLAB

```
clc; clear all; close all;

test_obj=visa('agilent','TCPIP0::10.70.13.233::INSTR'); % enter correct IP address

set(test_obj,'InputBufferSize',100000);

fopen(test_obj);

fprintf(test_obj, '*IDN?') ;

fscanf(test_obj);

fprintf(test_obj, 'INST:SEL ''NA''\n') ; % select network analyzer function

fprintf(test_obj, 'FREQ:STAR 30E3 \n'); % set start frequency

fprintf(test_obj, 'STOP 100E6 \n'); % set stop frequency fprintf(test_obj,

'SWE:POIN 10001\n'); % set number of points <= 10001 fprintf(test_obj, 'CALC:

FORMat SMITh\n'); % Smith chart

fprintf(test_obj, 'INIT:CONT 0 \n'); % turn off continuous mode

fprintf(test_obj, 'INIT:IMM; *WAI') ; % wait command

fprintf(test_obj, 'FORM:DATA ASCii,0') ; % define the data format as ASCii

fprintf(test_obj, 'CALC:DATA:SDATA?') ; % read data in Re/Im format s =
```

```
fscanf(test_obj);

data_points = str2num(s) ;

fclose(test_obj) ;

re = data_points(1:2:4002);

im = data_points(2:2:4002);

plot(re,im);

fid = fopen('cableshort.txt','w');

fprintf(fid,'%f',data_points);

fclose(fid);
```



In LabView



TCPIP0::10.70.13.233::INSTR
FORM:DATA ASCii,0
CALC:DATA:SDATA?
INST:SEL 'NA'
FREQ:START 30 kHz
FREQ:STOP 100 MHz
SWE:POIN 2001
CALC: FORMat SMITh
Time Delay
Delay Time (s)
10
1000000
read buffer
%f
2001
2
Waveform Graph

References

- http://trsys.faculty.jacobs-university.de/
- Wireline Communications - Werner Henkel