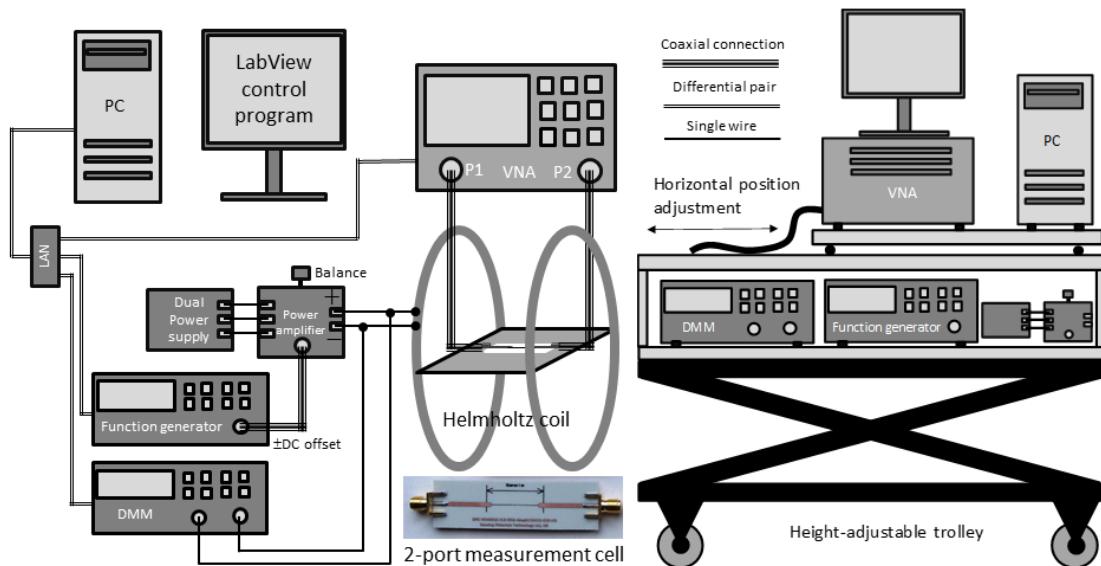


Multi-stage calibration for impedance measurements:

A practical guide using Rohde & Schwarz VNA

Contents

Introduction	2
References	5
Stage I: SOLT/TOSM calibration of VNA and bench coaxial cables	6
Stage II: Fixture removal	14
Stage III: Compensation of the delay time along the sample	16
Impedance measurements	27
Design of PCB measurement and calibration cells	34
Creation of S2P files for new PCB cells	36
Appendix 1: Measurements with RF probes. A technical report.....	38
Appendix 2: Python IDEs and libraries	51



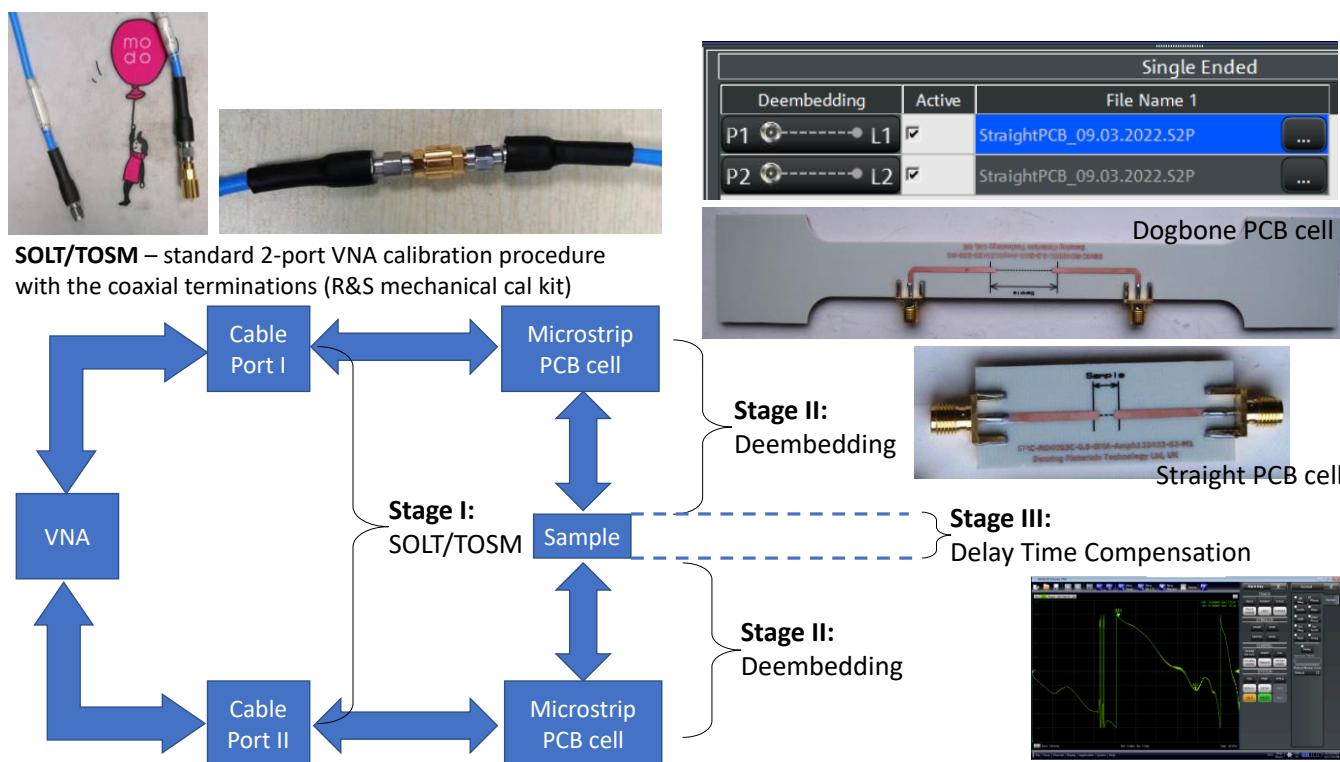
Contacts for consultations and collaborations:

- DM dmitriy.makhnovskiy@gmail.com (VNA measurements, numerical algorithms, Python, lab automation)
- YZ <https://yujiephysics.wordpress.com/> (CAD, CST studio, EM simulations, PCB layout, Python)
- KG kon837@gmail.com (IoT, project management, JavaScript, SQL)
- AU auddin@zju.edu.cn (material science, microwave measurements, lab management)

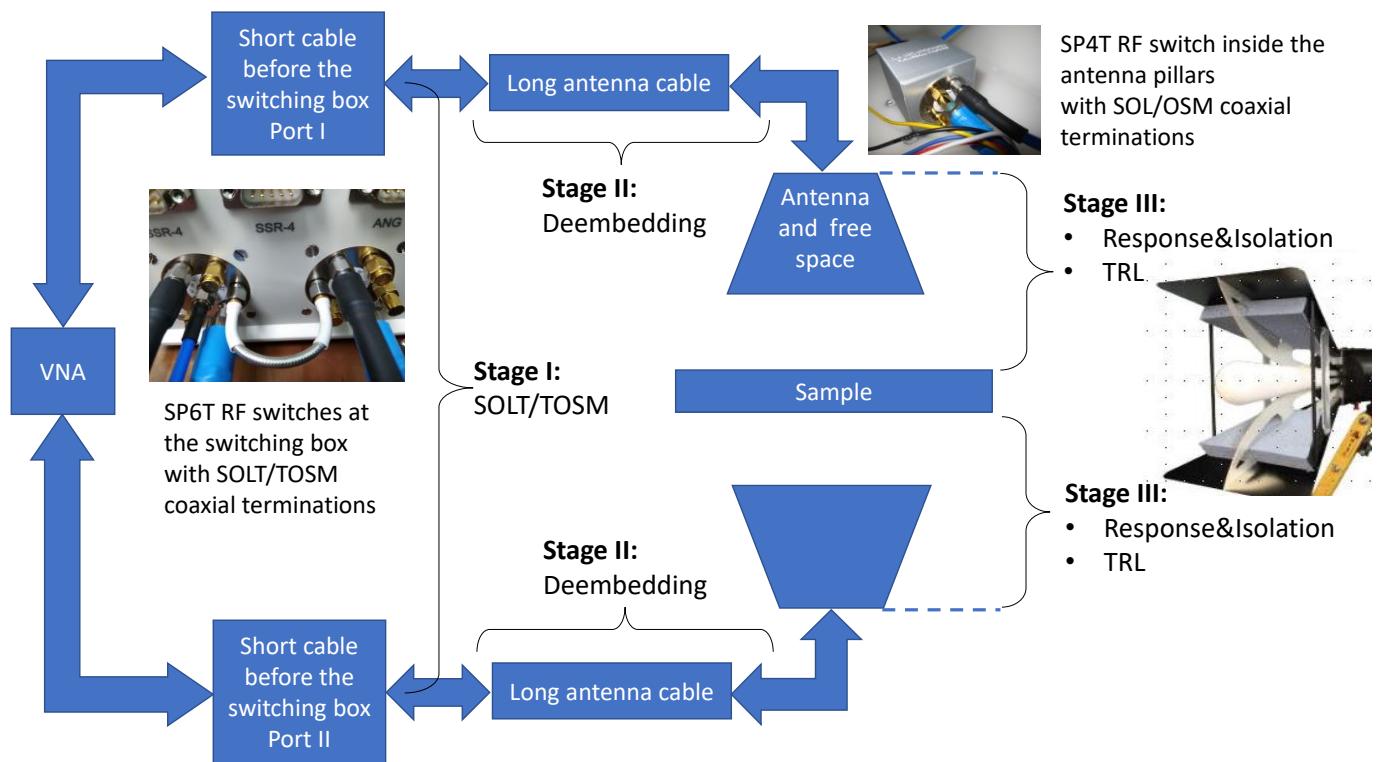
Introduction

In this manual, we discuss multi-stage VNA calibrations that we already use to measure the magnetic impedance (MI) of ferromagnetic wires placed on a PCB cell, as well as composite samples in free space. A characteristic feature of our measurements is the presence of non-coaxial waveguide structure that supplies microwave excitation to the sample under test. Thus, the problem arises of creating such calibration procedures that could extend the reference plane up to the sample interface. In addition, a wire sample above the ground plane (PCB cell) itself is a waveguide structure leading to an additional phase incursion.[1] The latter must be compensated if the measured S-parameter needs to be converted to the impedance.

The figures below show the 3-stage calibration schemes already implemented at the microwave lab in InCSI for impedance and free space measurements.[2, 3] In this guide, which is a further development of the method proposed in [1], we only explain impedance measurements. Although the algorithms used in the guide are quite simple, they play an essential role in the implementation of the evolving approach. Our program codes in Python have been saved at GitHub in a public repository.[4] The latest version of the Python interpreter can be downloaded from [5]. To run the Python programs, we suggest using PyCharm IDE.[6] Mark Lutz wrote a brilliant book on Python programming language.[7] When creating graphical user interfaces (GUI) for device automation, we actively use Node-Red.[8] For further learning, we recommend videos and webinars on various VNA measurement methods.[9, 10]



Impedance measurement scheme discussed in this guide: straight and dogbone PCB cells (100 kHz – 15 GHz) and wire samples (other types of the samples, e.g. thin films, are also possible).



The relevance of impedance measurements for composite materials was discussed in our recent presentation at PIERS conference.[11] Modern algorithms and commercial solvers (for example, CST Studio) are able to simulate the distribution of electromagnetic (EM) fields, including scattering in free space, from objects of complex geometry. To reproduce the EM properties of the constituting materials, it is often possible to get by with standard models that have a small number of adjustable parameters and are capable of describing dispersion and resonance behaviour. Thus, while remaining exclusively within the model framework, practically reliable results can be obtained. However, there are classes of problems that cannot be solved within the framework of this approach. This situation arises, for example, when studying the EM properties of polymer composite materials with ferromagnetic wire inclusions of different shapes. The magnetic domain structure of such inclusions may defy not only a simplified model description but even modelling from first principles. Meanwhile, it significantly affects the conducting and scattering properties, determining the most interesting effects. We proposed a new hybrid approach to the scattering theory in such systems. The wire geometry of the inclusions makes it possible to single out a parameter – the surface impedance, which can be accurately measured over a wide frequency range, from kilohertz to tens of gigahertz. The surface impedance includes both conductive and magnetic properties of inclusions, determining their microwave scattering properties through magnetically-tunable skin-effect. The scattered field from a single inclusion can be rigorously calculated by solving the integro-differential antenna equation with the impedance boundary condition (see below). Within the framework of this hybrid approach, combining metrological measurements of individual properties of inclusions and an accurate solution of an external electrodynamic problem, there is

no need to consider the most complex internal problem. Solving the single-particle scattering problem is the key stage to constructing a general theory that considers the collective response of many particles, as well as their interactions.



浙江大学
ZHEJIANG UNIVERSITY



On hybrid approach in microwave scattering theory for wire-filled composites

Azim Uddin and Faxiang Qin

*Institute for Composites Science Innovation (InCSI), School of Materials Science & Engineering,
Zhejiang University, 38 Zheda Road, Hangzhou, 310027, PR China*

Yujie Zhao

School of Physics and Astronomy, University of St. Andrews, Fife KY16 9SS, Scotland

功能复合材料与结构研究所
Institute for Composites Science Innovation

Antenna equation with the impedance boundary condition (cgs units)

An external EM field will induce the prevailing $(\bar{e}_x, \bar{h}_\phi)$ field polarization on the wire surface – longitudinal electric and circular magnetic fields. The same polarization can be induced by a linear current flowing along the wire axis with the volume density $j(x)\delta_s$, where $\delta_s(y, z)$ is two-dimensional Dirac's function. The function $j(x)$ will be further referred to as "linear current".

Antenna
equation in the
frequency
domain

$$\frac{\partial^2}{\partial x^2} (G * j) + k^2 (G * j) = \frac{i\omega\epsilon}{4\pi} \bar{e}_{0x}(x) - \frac{i\omega\epsilon\zeta_{xx}}{2\pi ac} (G_\varphi * j) + \frac{i\omega\epsilon\zeta_{x\varphi}}{4\pi} \bar{h}_{0x}(x)$$

Dielectric constant of composite matrix

Components of the surface impedance matrix

$k = (\omega/c)\sqrt{\epsilon}$ – wave vector

Impedance boundary condition
for the total surface electric and
magnetic fields (φ - circular, x -
longitudinal)

$$\begin{aligned}\bar{E}_\varphi &= \zeta_{\varphi x} \bar{H}_\varphi - \zeta_{\varphi\varphi} \bar{H}_x \\ \bar{E}_x &= \zeta_{xx} \bar{H}_\varphi - \zeta_{x\varphi} \bar{H}_x\end{aligned}$$

main contribution to
the scattered EM field

Convolutions
with the Green
functions

Electric and magnetic fields in the incident wave
projected on the wire direction

$$\zeta_{xx}^{\text{cgs}} = \frac{10^9 a Z [\Omega]}{2 cl}$$

ζ_{xx}^{cgs} - wire longitudinal surface
impedance (cgs units)
 Z – wire impedance in Ω
 l – wire length
 a – wire radius
 c – speed of light (cm/s)



功能复合材料与结构研究所
Institute for Composites Science Innovation

Hybrid approach to solving the antenna equation

The surface impedance includes both conductive and magnetic properties of the wire sample thus freeing us from the need to solve the EM internal problem.

ζ_{xx}^{cgs} - wire longitudinal surface impedance (cgs units)
 Z – wire impedance in Ω
 l – wire length
 a – wire radius
 c – speed of light (cm/s)

$$\zeta_{xx}^{\text{cgs}} = \frac{10^9 a Z [\Omega]}{2cl} \quad \text{Measured in experiment}$$

Used in numerical calculations

$$\frac{\partial^2}{\partial x^2} (G * j) + k^2 (G * j) = \frac{i\omega\epsilon}{4\pi} \bar{e}_{0x}(x) - \frac{i\omega\epsilon\zeta_{xx}}{2\pi ac} (G_\varphi * j) + \frac{i\omega\epsilon\zeta_{x\varphi}}{4\pi} \bar{h}_{0x}(x)$$

Can be neglected



The solution of the integro-differential antenna equation for the linear current will allow calculating the scattered electromagnetic field. It will take into account both the resistive and radiative losses. A similar antenna equation can be written for a wire of any spatial shape.

This term with the off-diagonal impedance component $\zeta_{x\varphi}$ is non-zero only for some specific ferromagnetic wires. And even for them the main contribution will be from the longitudinal impedance.



功能复合材料与结构研究所
Institute for Composites Science Innovation

References

- [1] Yujie Zhao et al 2020 Meas. Sci. Technol. 31 025901, “Novel broadband measurement technique on PCB cells for the field-and stress-dependent impedance in ferromagnetic wires”; <https://iopscience.iop.org/article/10.1088/1361-6501/ab4556>
- [2] Institute for Composites Science Innovation (InCSI): <http://www.composites.zju.edu.cn/en/index.asp>
- [3] DYK design team (UK – China): www.dykteam.com; We are tweeting on @dyk_team 😊
- [4] GitHub repository with the Python program codes used in this guide: <https://github.com/DYK-Team/vna-impedance-measurments>
- [5] Python free download: <https://www.python.org/downloads/>
- [6] PyCharm IDE to run the Python programs (choose “Community” for free download); <https://www.jetbrains.com/pycharm/download/#section=windows>
- [7] Mark Lutz, “Learning Python: Powerful Object-Oriented Programming”, https://www.amazon.co.uk/Learning-Python-Mark-Lutz/dp/1449355730/ref=sr_1_9?keywords=Python&qid=1647295818&sr=8-9
- [8] Node-Red: <https://nodered.org/>
- [9] VNA measurement techniques: <https://coppermountaintech.com/videos/>
- [10] Webinars on VNA measurements: <https://coppermountaintech.com/webinars/>
- [11] Azim Uddin et al., “On Hybrid Approach in Microwave Scattering Theory for Wire-filled Composites”, 2021 Photonics & Electromagnetics Research Symposium (PIERS), <https://ieeexplore.ieee.org/document/9694931>

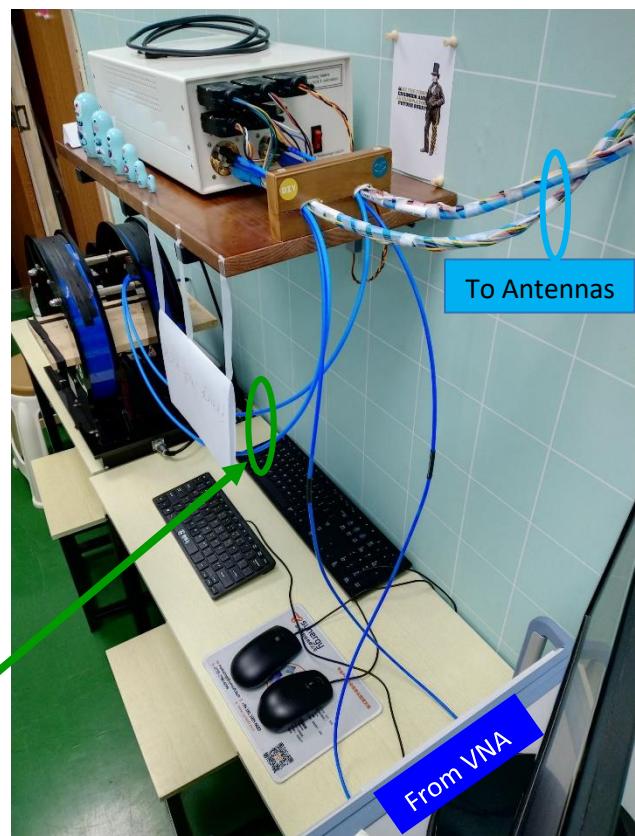
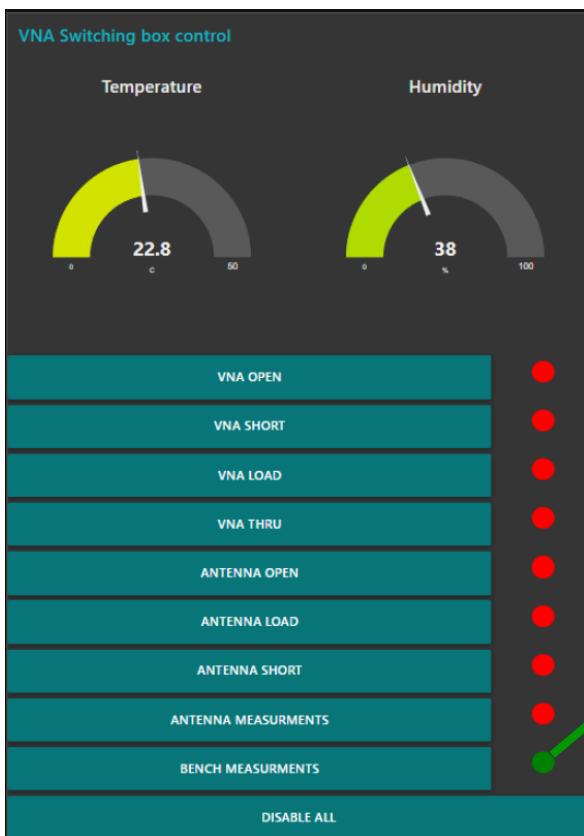
Stage I: SOLT/TOSM calibration of VNA and bench coaxial cables

SOLT – abbreviation for the full 2-port VNA calibration procedure with SHORT, OPEN, LOAD, and THROUGH standards using the 12-term error correction model. Rohde & Schwarz (R&S) calls it TOSM – THROUGH, OPEN, SHORT, and MATCH.

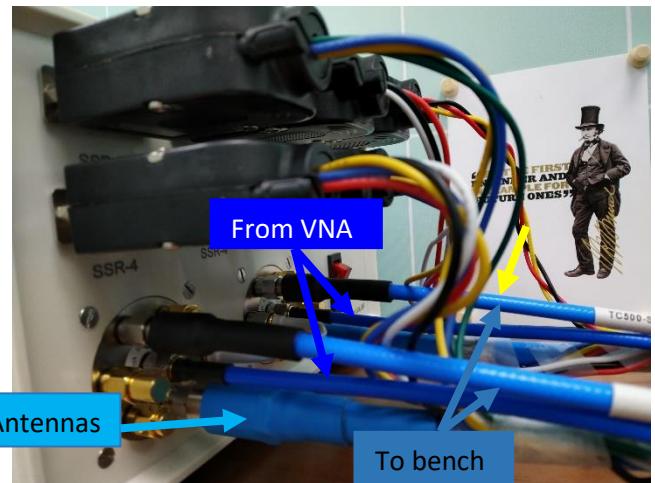
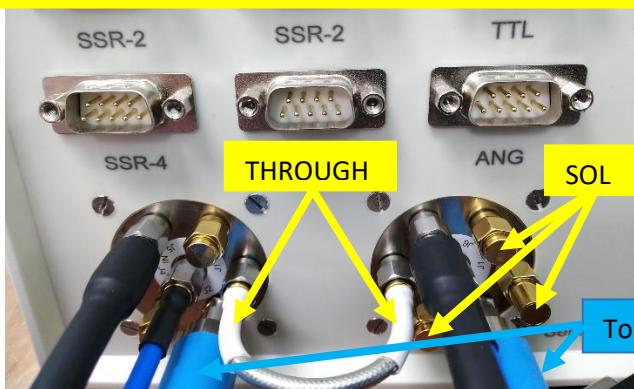
1. Turn ON the switching box and activate the switching box control panel by clicking the icon “SwitchingBox” on PC Desktop. Node-Red interface will start. Wait until the temperature and humidity indicators will show their values.



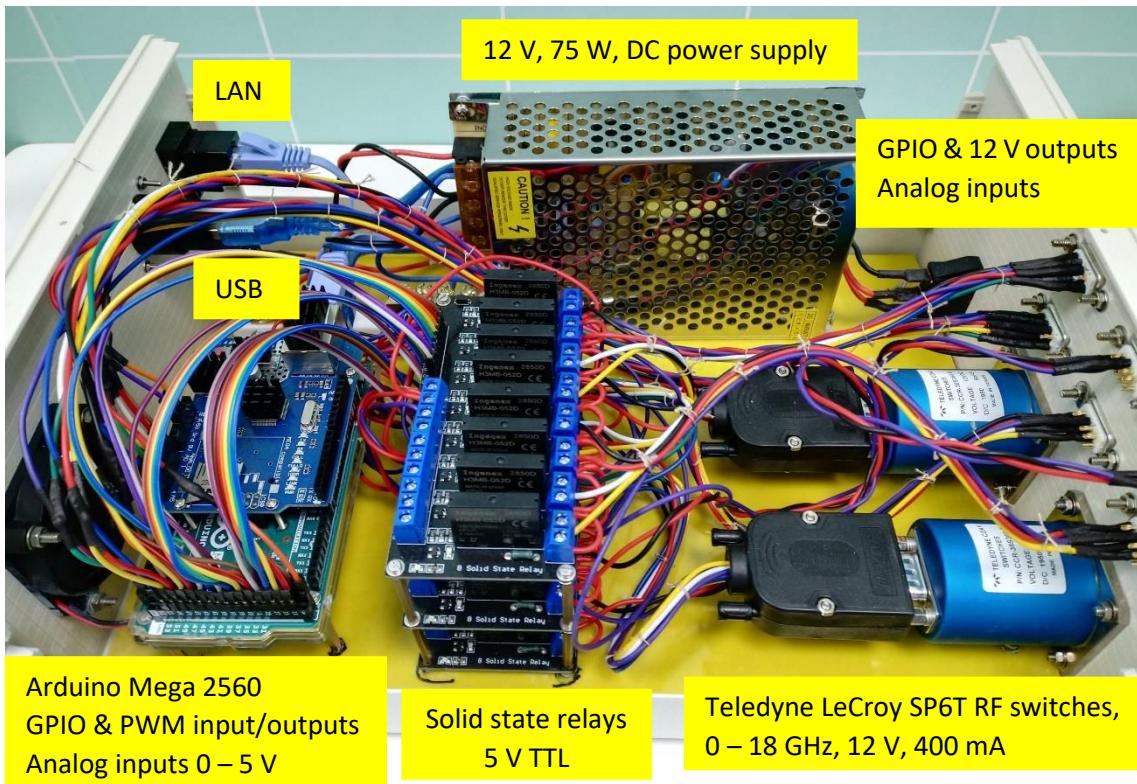
2. Click on the BENCH MEASUREMENTS (indicator turns green).



The box with two SP6T RF switches terminated with coaxial SOLT.



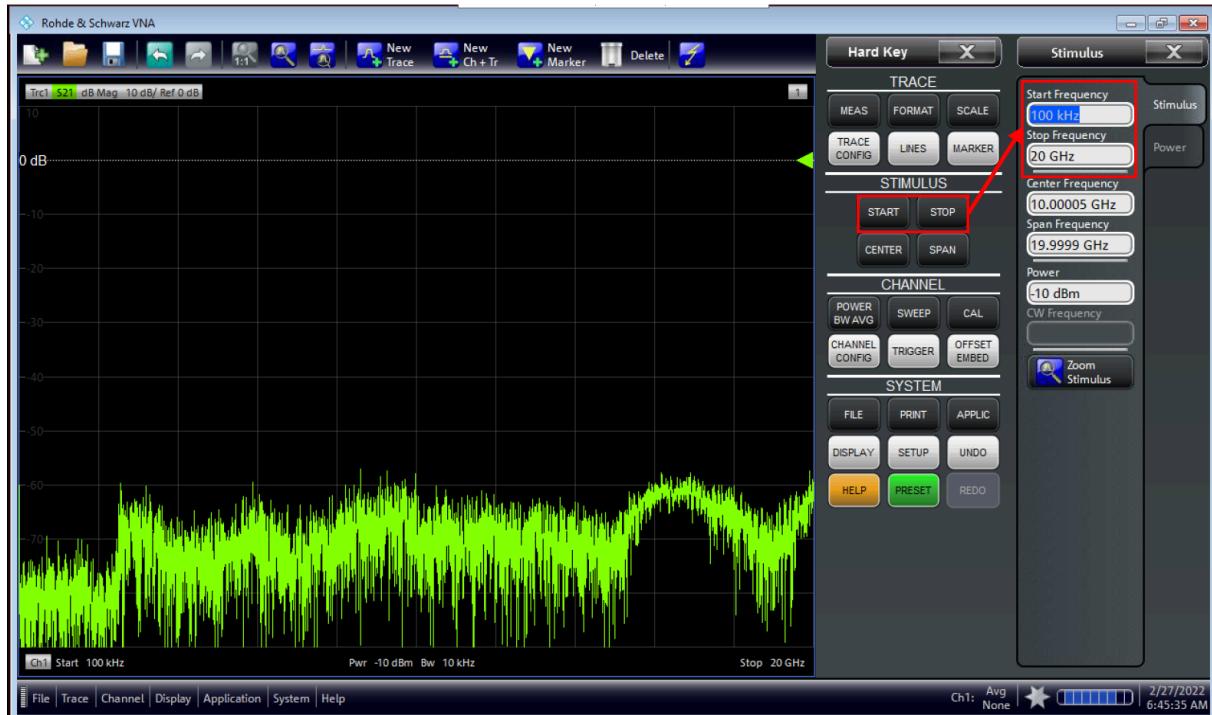
These SOLT terminations are used only for the free space Stage I (see the figure in Introduction). For the impedance measurements, the switches are configured to pass signals further down the bench cables, bypassing the SOLT terminations at the box.

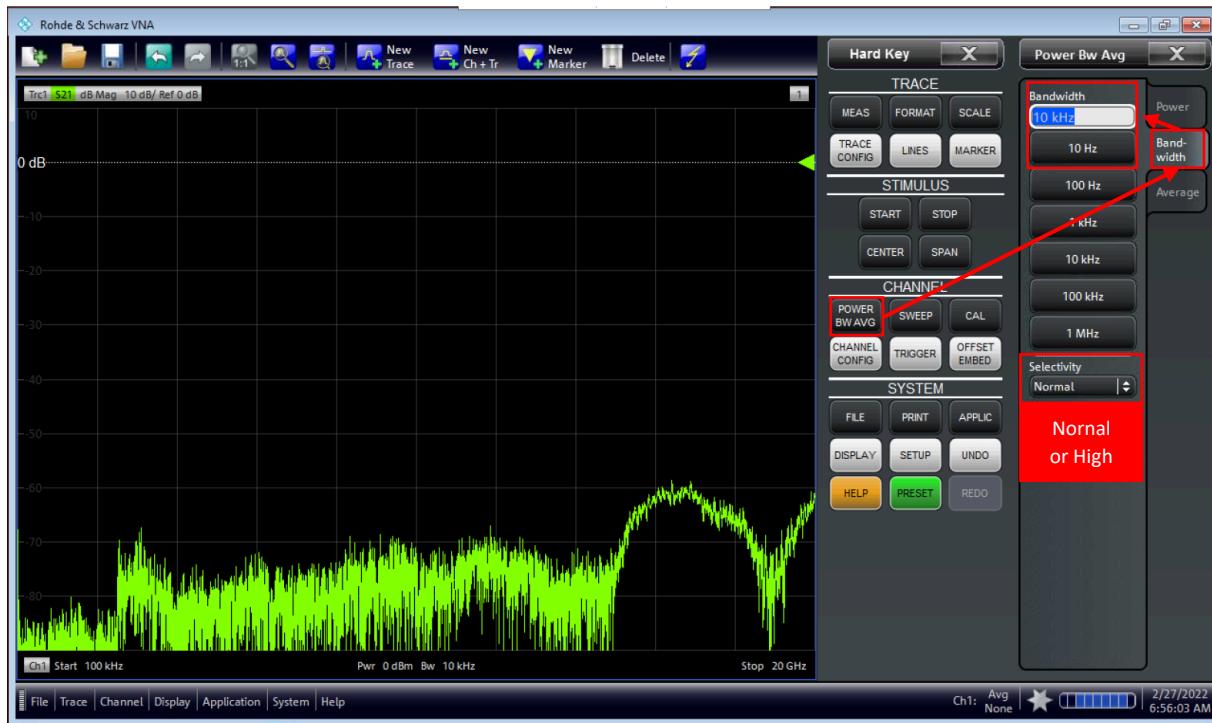
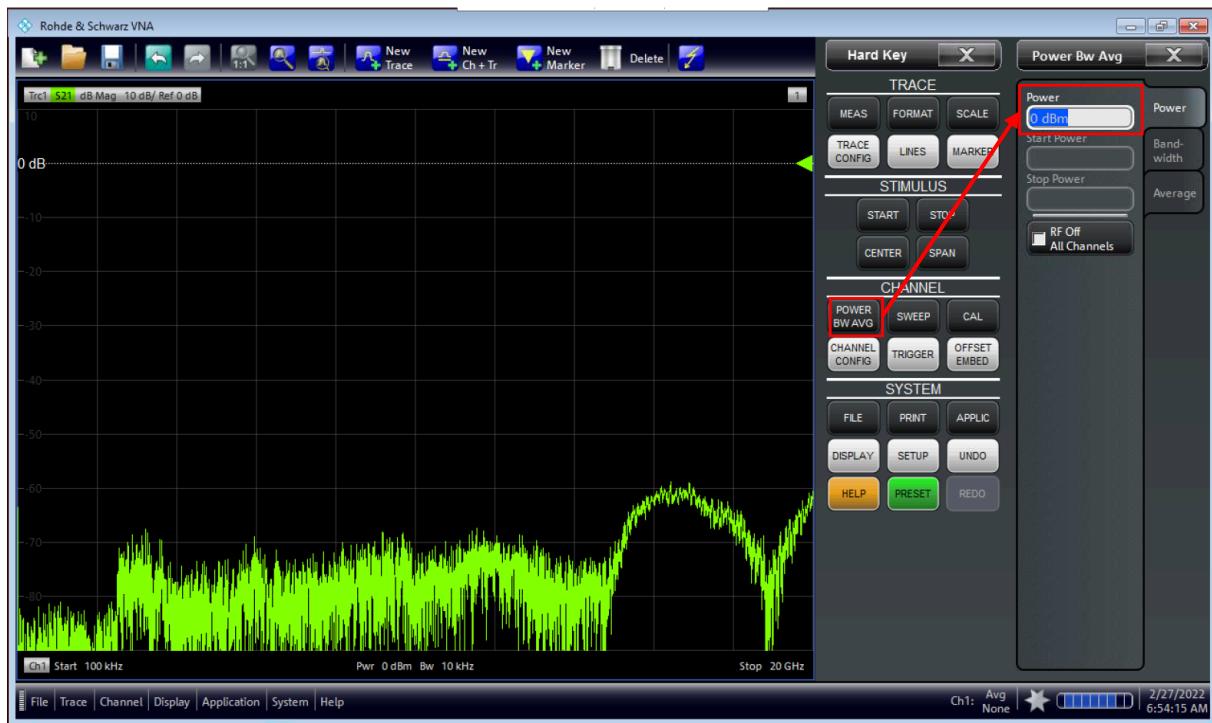


The switching box that controls the internal SP6T RF switches terminated with the coaxial SOLT (automatic 2-port VNA calibration) and SP4T RF switches in the antenna pillars terminated with the coaxial SOL (deembedding the long antenna cables).

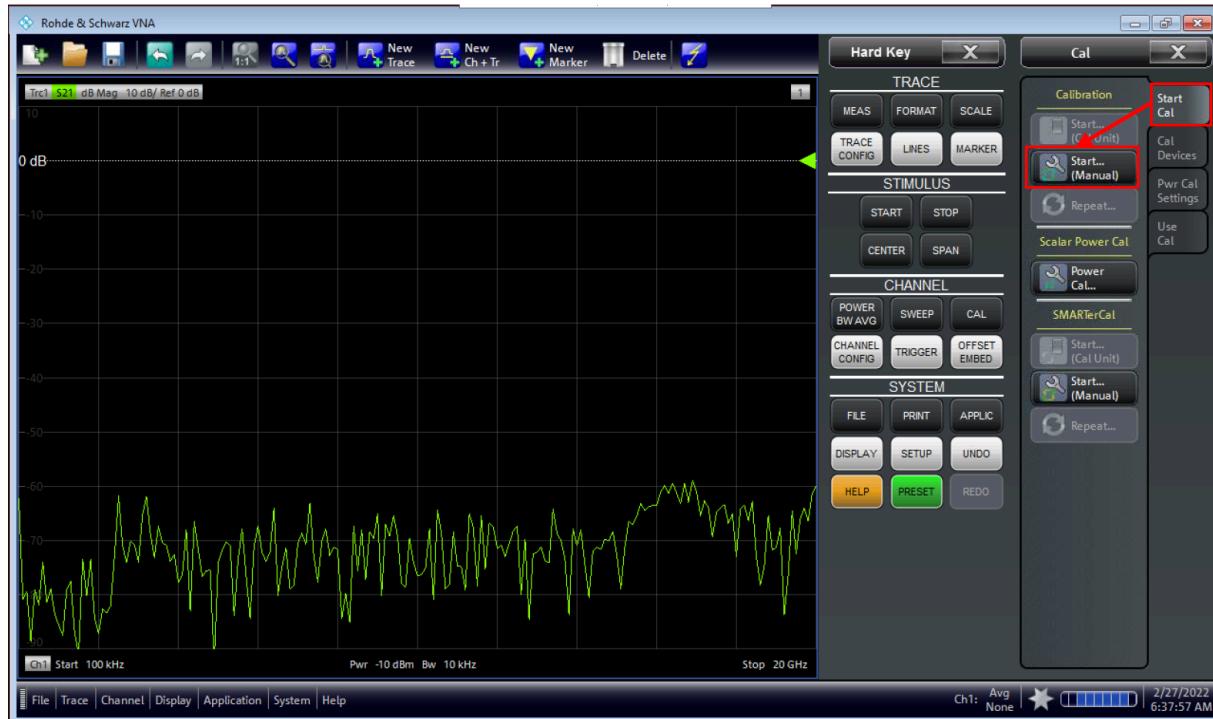
Designed and assembled in China by DYK@2020

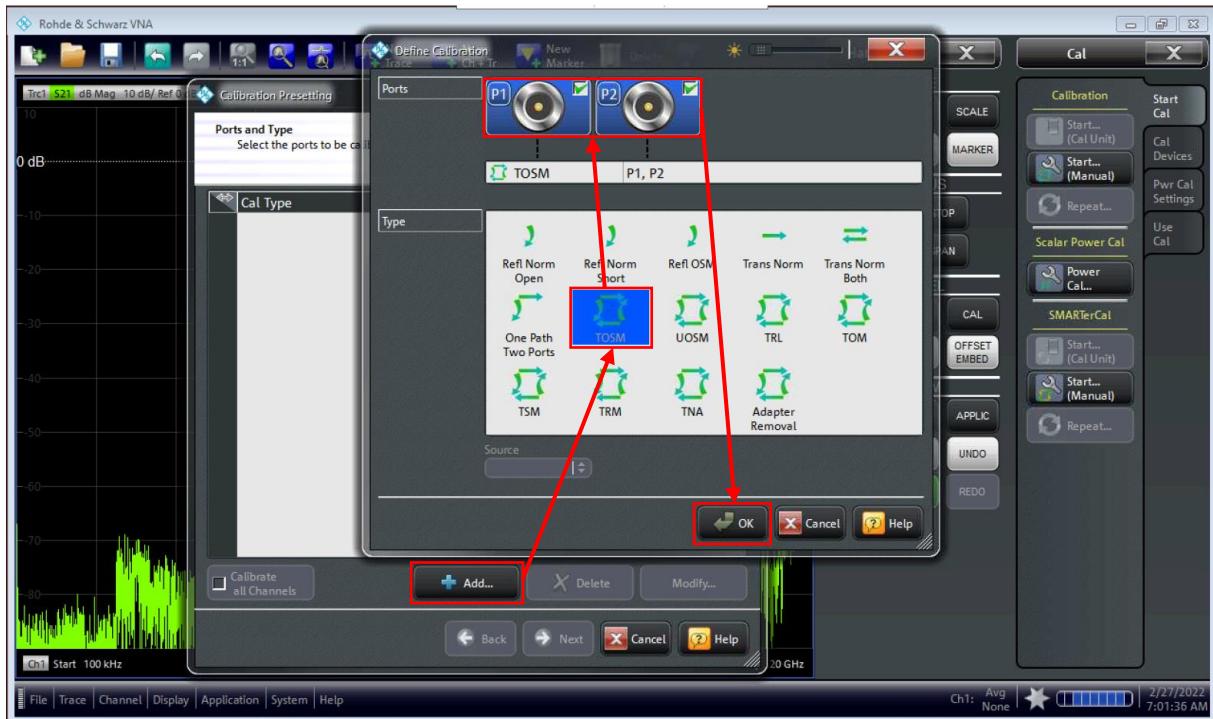
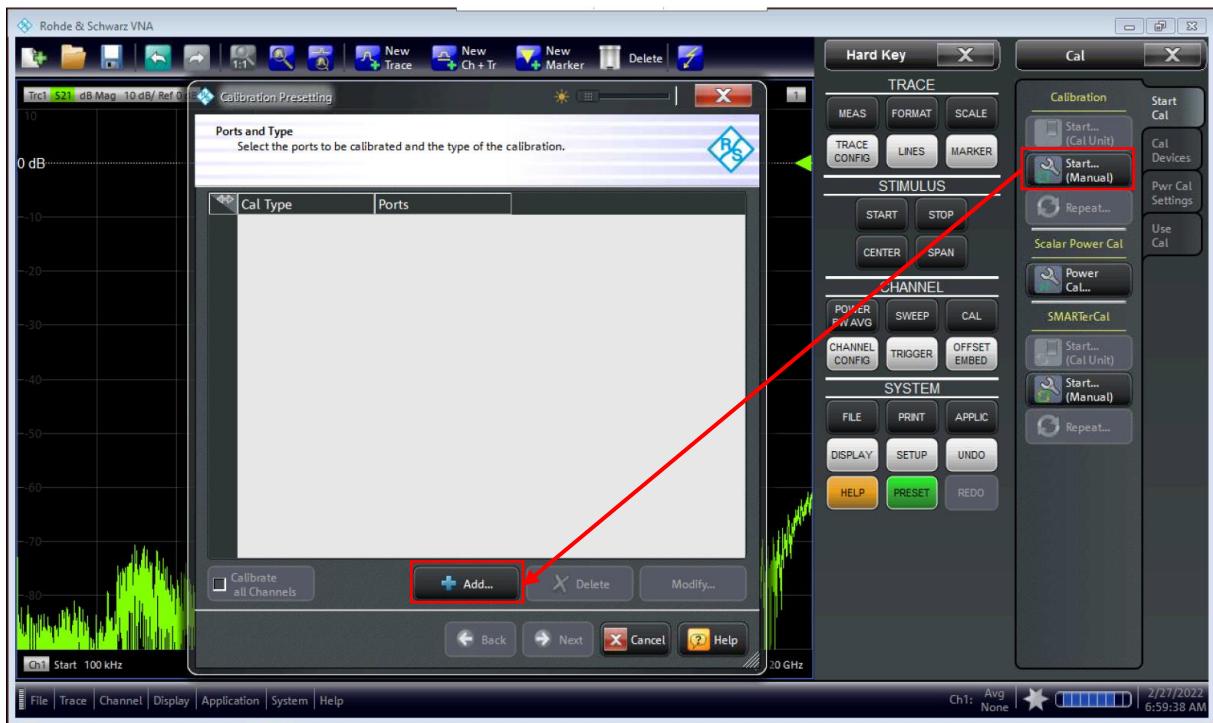
3. On VNA, choose the start and stop frequencies (maximum range 100 kHz – 20 GHz or a narrower range), number of frequency points (we usually use 5000, but it is up to you), power (0 dBm or less), IFBW (1 kHz or 10 kHz is recommended), Selectivity (Normal or High).

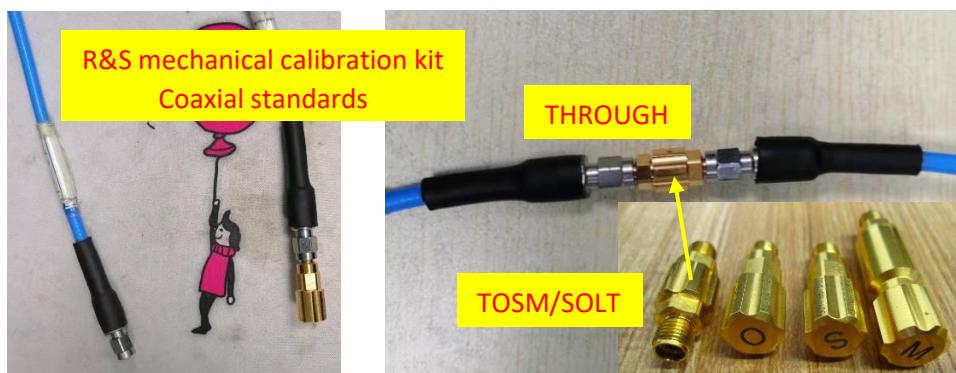
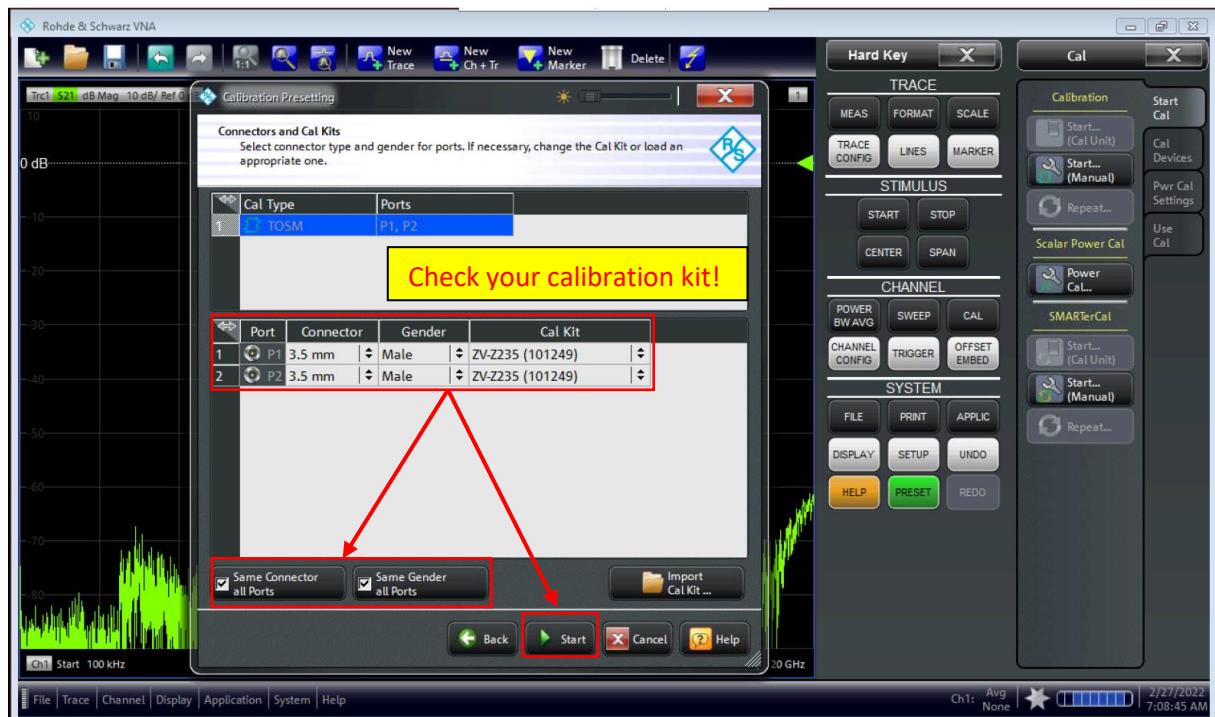
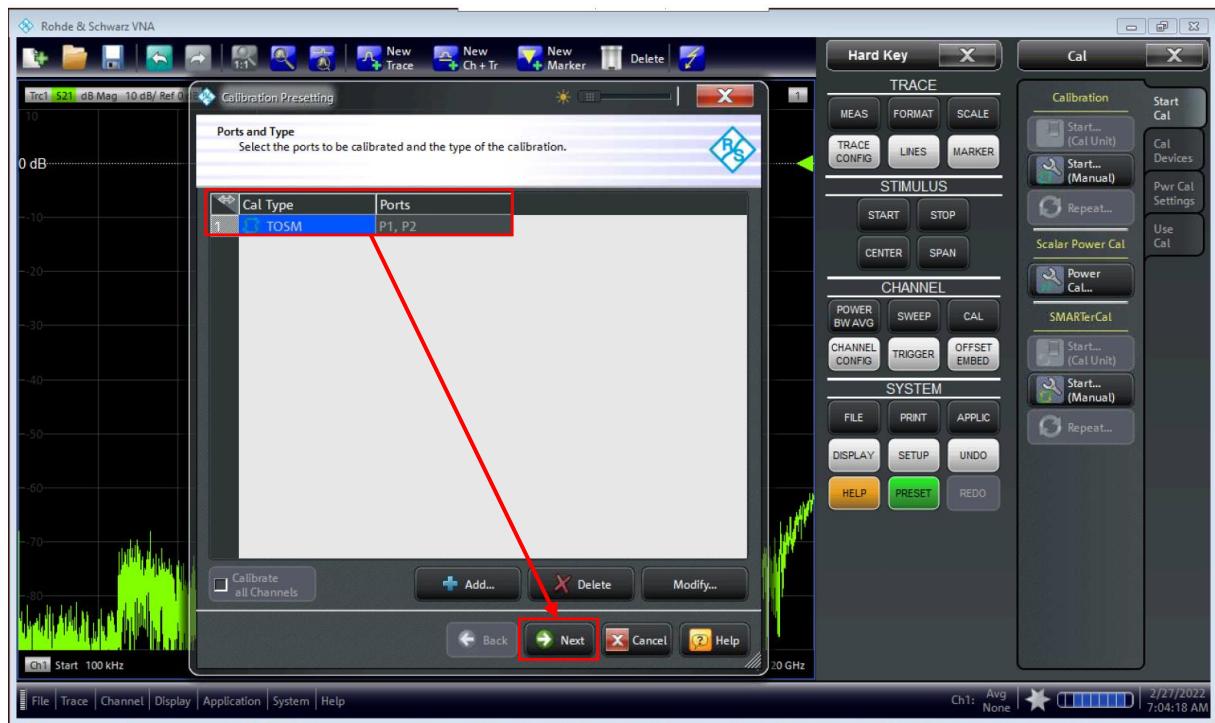


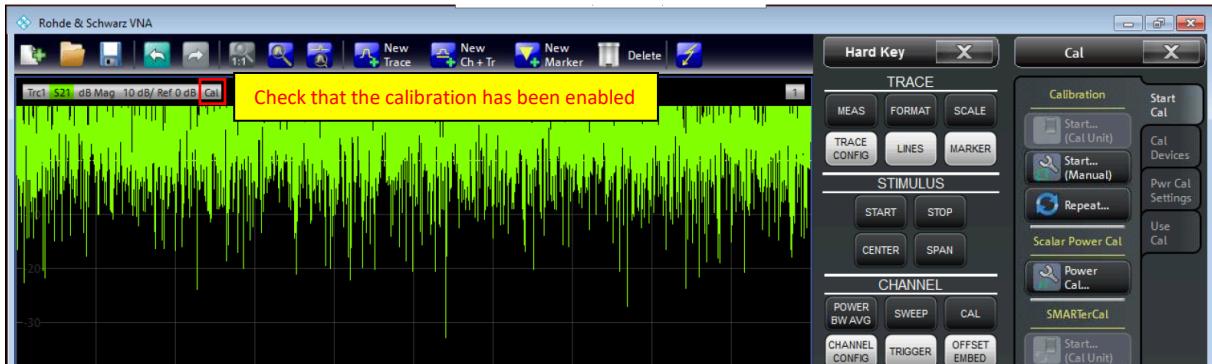


4. Start performing SOLT – SHORT, OPEN, LOAD, THRU calibration (TOSM in terms of R&S), using the coaxial standards from R&S mechanical calibration kit which are connected to the bench cables.









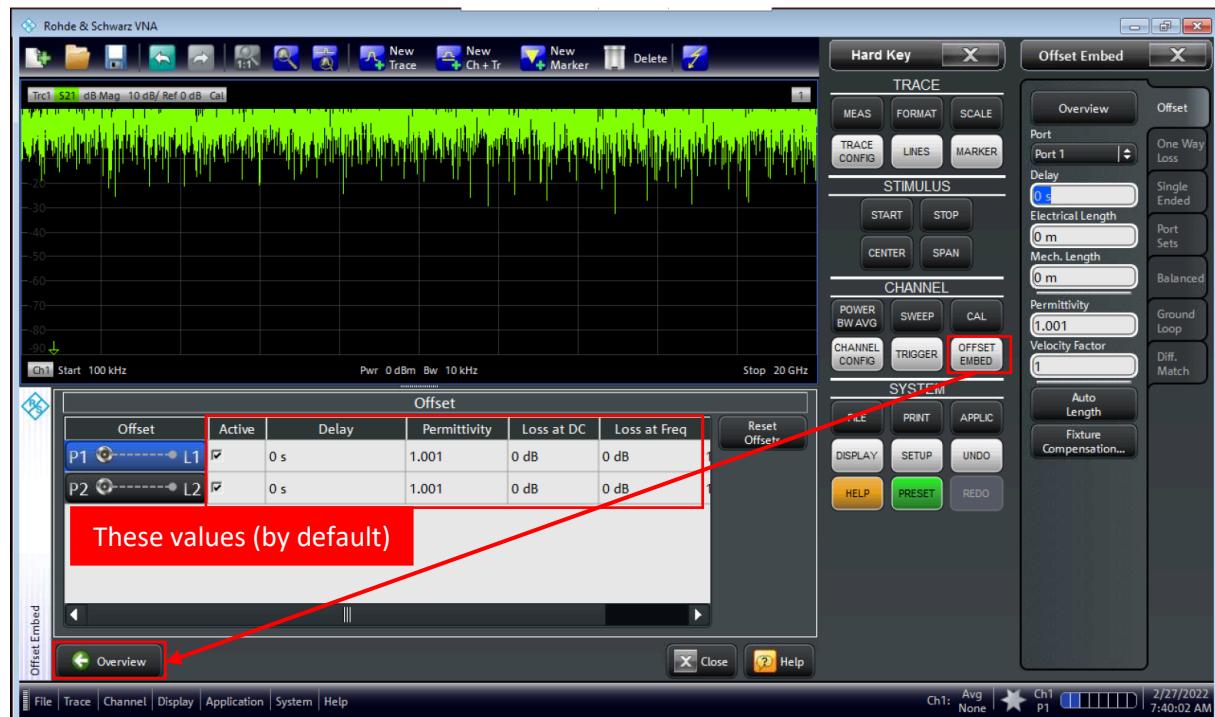
Stage II: Fixture removal

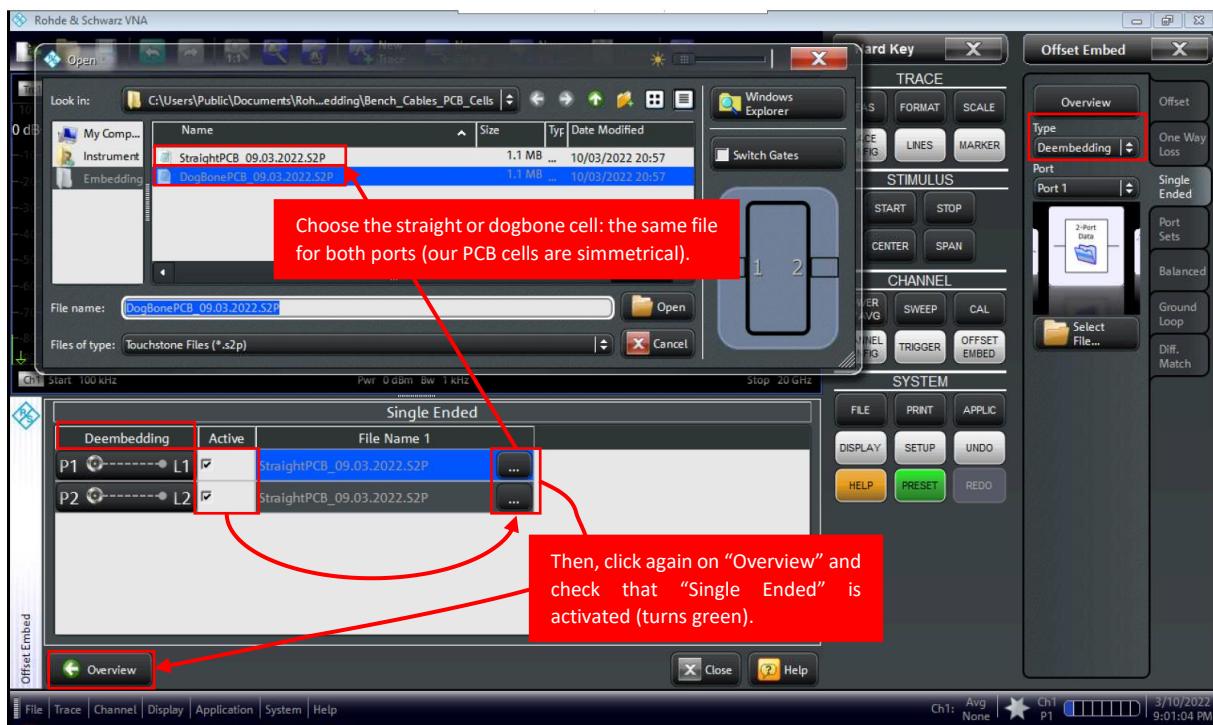
S2P files, containing S-parameters of the PCB cells (straight and dogbone), are saved on VNA in the folder:

C:\Users\Public\Documents\Rohde-Schwarz\Vna\Deembedding\Bench_Cables_PCB_Cells

Later in this guide, we will explain how to create S2P files for new cells, if necessary.

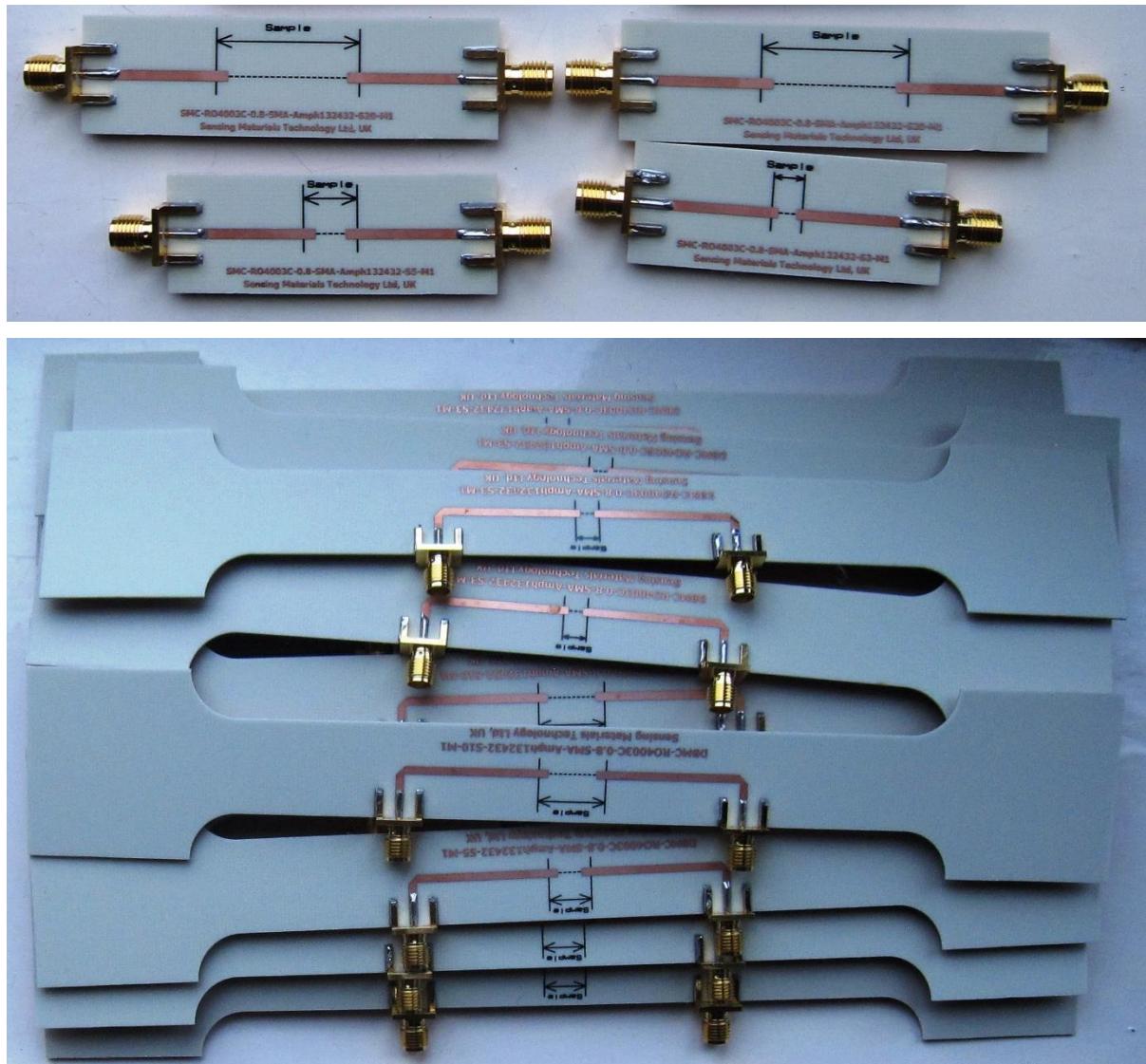
To activate the deembedding, start with “OFFSET/EMBED” button on the VNA’s front panel.





Stage III: Compensation of the delay time along the sample

1. After activating the calibration and deembedding, as explained above, you can start measuring the samples (wires or thin films) in the PCB cells. The cells for the various sample lengths that we currently use are shown in the figures below. The measured parameter is S_{21} . The cells are 50Ω microstrips on a dielectric substrate and SMA-Female connectors with a continuous ground plane on the reverse side. The dogbone cell is used for measuring the stress-impedance.[1]



2. The sample over the ground plane is the waveguide itself, which introduces the phase incursion. When converting the measured S_{21} parameter to the impedance, which is always assumed to be lumped, this phase incursion must be compensated: $\tilde{S}_{21} = S_{21} \times \exp(i\omega\Delta t)$. Here, where Δt is the delay time along the sample and \tilde{S}_{21} is the recovered parameter. In turn, the delay time can be derived from the unwrapped phase of S_{21} . To do this, we have developed the algorithm in Python, which has been saved on VNA's PC and in a public repository at GitHub.[4]

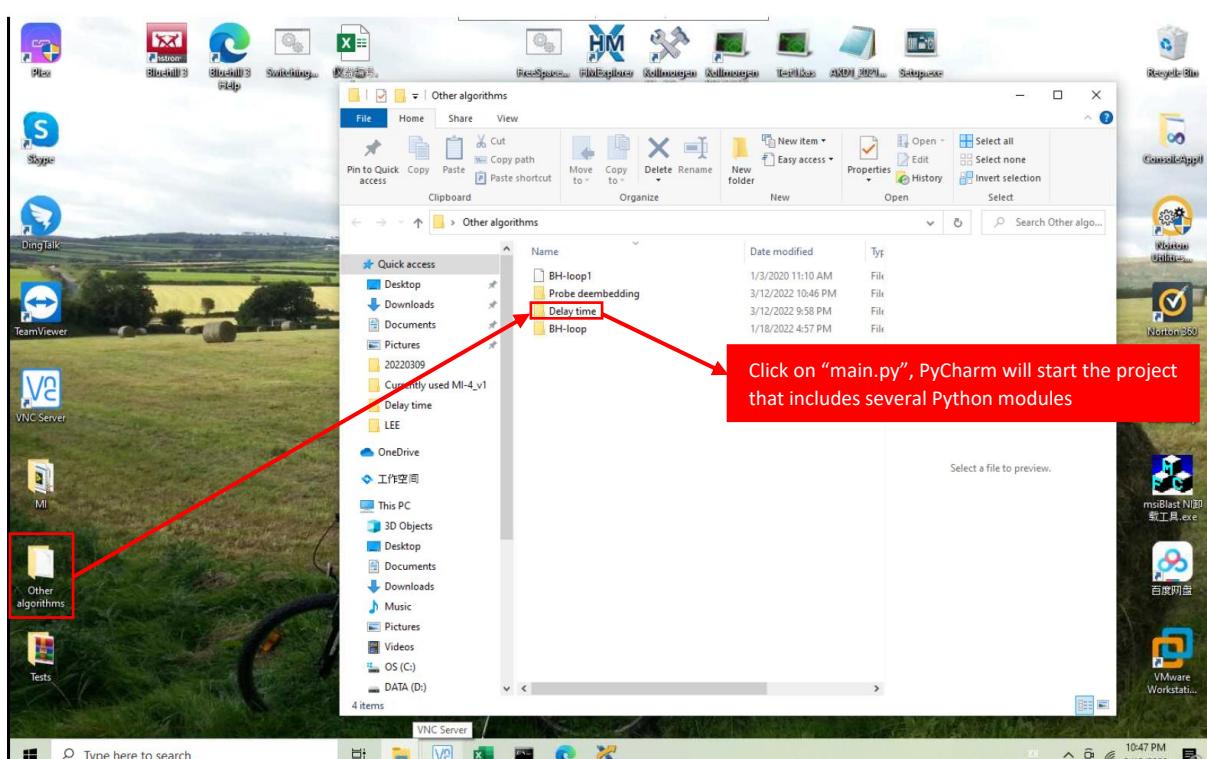
Mount the sample under test onto the PCB cell (we use silver paint), connect it to the bench cables, and save S2P file of the sample under the name “S.s2p”. Use the entire frequency range for the sweep, the same as during the calibration. Be sure that cal and deembedding are activated.



3. Open the saved S.s2p (9 columns) with Notepad and delete the header. Save it.

#	HZ	S	R1	R	50.00							
! Rohde & Schwarz Vector Network Analyzer						Delete						
! Created: UTC 3/12/2022, 1:48:11 PM												
!	freq[Hz]	re:S11	im:S11	re:S21	im:S21	re:S12	im:S12	re:S22				
1.000000000000000E5	3.276091567478915E-1	6.045994088386428E-3	6.721530458607924E-1	-1.147061010482287E-3	6.681472965161							
4.100780156031206E6	3.497412267211859E-1	2.540767324593893E-2	6.500083295177960E-1	-2.563850565618409E-2	6.501039897558							
8.101560312062413E6	3.623767458370507E-1	2.743056832396509E-2	6.388104333579362E-1	-2.786989932251379E-2	6.305128835002							
1.210234046809362E7	3.689017813812323E-1	2.850607487402986E-2	6.330171291295427E-1	-2.957431548705082E-2	6.197708936422							
1.610312062412483E7	3.736939351164147E-1	3.031613453026474E-2	6.291188870491979E-1	-3.172555910156595E-2	6.123717355354							
2.010390078015603E7	3.774679541480149E-1	3.201082585380635E-2	6.258979629037265E-1	-3.359754103486454E-2	6.06130904378E							
2.410468093618724E7	3.806880625554396E-1	3.359739420116682E-2	6.234319225242108E-1	-3.599450373168663E-2	6.00814673441E							
2.810546109221844E7	3.820500117707462E-1	3.500396538840299E-2	6.201619576293418E-1	-3.797581175114298E-2	5.951486794972							
3.210624124824965E7	3.845494284262933E-1	3.703382939503351E-2	6.178663576495382E-1	-4.016674190824200E-2	5.907751675508							
3.610702140428086E7	3.867271780991687E-1	3.941613442561704E-2	6.160821700200836E-1	-4.049508244497926E-2	5.877862383237							
4.010780156031206E7	3.889538074284432E-1	4.042858144360689E-2	6.133990719264070E-1	-4.200302460764390E-2	5.84851088951E							
4.410858171634327E7	3.911370411176998E-1	4.117793690696581E-2	6.110752460320812E-1	-4.282449773511629E-2	5.821174841552							
4.810936187237448E7	3.928109726027624E-1	4.147633597431368E-2	6.086235023253330E-1	-4.393712675225529E-2	5.793556442502							
5.211014202840568E7	3.945589619283250E-1	4.176100911900325E-2	6.067713549551214E-1	-4.489321923843579E-2	5.761440170085							
5.611092218443689E7	3.961744216916867E-1	4.235172151875829E-2	6.052613313243817E-1	-4.554307308711227E-2	5.729220621278							
6.011170234046809E7	3.975951353713452E-1	4.333474746631446E-2	6.042728995244605E-1	-4.669250532140768E-2	5.700483028346							
6.411248249649930E7	3.991139444795525E-1	4.468244277564946E-2	6.033920856351517E-1	-4.820613776531217E-2	5.671253850385							
6.811326265253051E7	4.006458676507633E-1	4.641414824646428E-2	6.020293535614902E-1	-4.966027821414473E-2	5.648361425901							

4. Open the saved S.s2p (already without the header) in Excel (Data → From Text/CSV). Since we need only the dispersion of S_{21} , other columns **except 1, 4, and 5** must be deleted (first, delete the last four columns, and then the second and third columns). Do this and then save the file as S.csv with the comma delimiter. Finally, open S.csv in Excel and the delete the header (Column 1, Column 2, ...) leaving only the numbers in the three columns. Save it. Now, the file S.csv is ready for the numerical calculations using our interactive algorithm in Python (“Delay time”).
5. On the VNA’s PC we have installed PyCharm IDE to run Python programs.[6] Open “Delay time” algorithm as shown below. Alternatively, you can open **main.py** by right-clicking and selecting “Open with → Python”.



```

1 # Delay time algorithm
2 #
3 # The algorithm calculates the delay time along a fixture/sample under test (wire or microstrip above the ground plane,
4 # coaxial probe, waveguide structure, or free space) using the unwrapped phase of the S11 or S21 parameter
5 # measured on VNA.
6 #
7 # Yujie Zhao, University of St. Andrews, Scotland, 11 March 2022
8 # Dmitriy Makhnovskiy, Sensing Materials Technology Ltd, UK
9 # DYK team: http://dykteam.com/
10 #
11 #
12 import numpy as np
13 from Acquisition import acq
14 from PhaseUnwrapping import unwrap
15 import matplotlib.pyplot as plt
16
17 print('')
18 print('*'*50)
19 print('1) VNA together with the cables must be already calibrated with a coaxial calibration kit.')
20 print('    For a wide frequency range, use SOLT calibration (TDSM in terms of Bohde&Schwarz).')
21

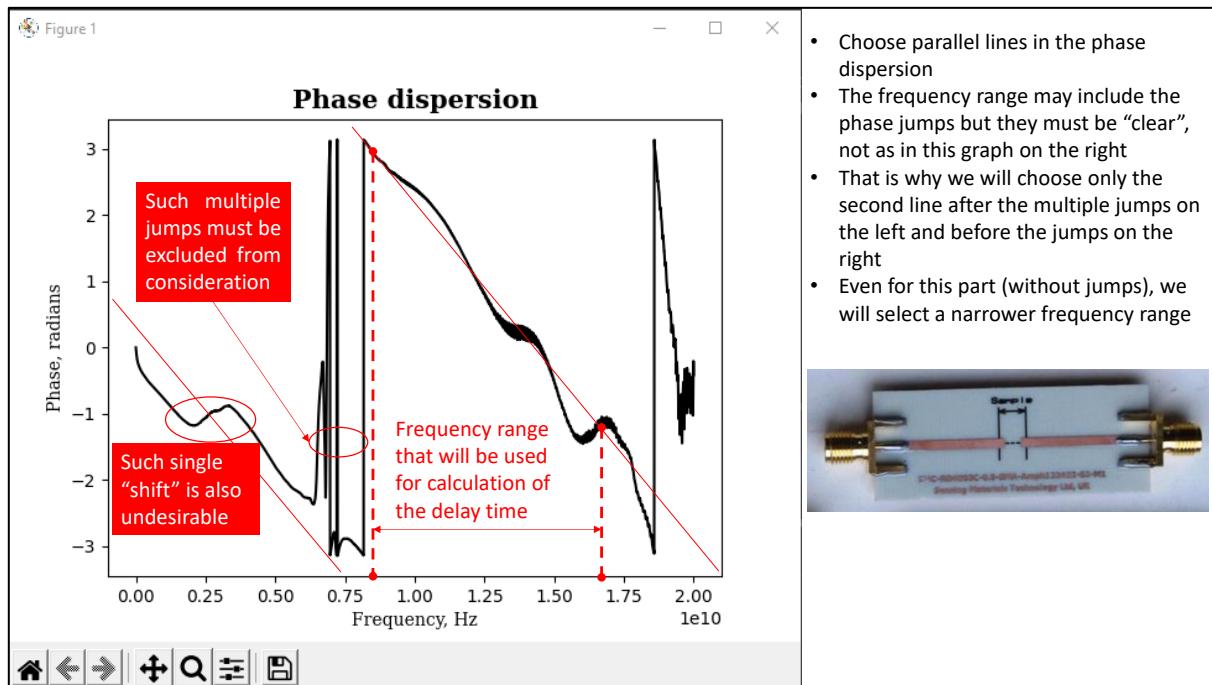
```

The screenshot shows the PyCharm IDE interface. In the top navigation bar, there are tabs for 'File', 'Edit', 'View', 'Navigate', 'Code', 'Refactor', 'Run', 'Tools', 'Git', 'Window', and 'Help'. The current file is 'main.py'. Below the tabs, there's a 'Project' view showing files like 'main.py', 'PhaseUnwrapping.py', and 'Acquisition.py'. A red box highlights the text 'DO NOT CHANGE ANYTHING IN THE PROGRAM BODY!' in the code editor. In the bottom right corner of the code editor, there's a 'Stop' button. A red arrow points to this button from the text 'The interactive window is launched when the program starts'. In the bottom right corner of the PyCharm interface, there's a red box containing the text 'Copy and paste the full path to S.csv and then push Enter. Then, follow further instructions.' A red arrow points to the text 'Enter/paste the path of your folder where the file S.CSV is located:' in the terminal window.

Enter/paste the path of your folder where the file S.CSV is located: (copy & paste the full path to S.csv)

Please choose the delimiter used in S.CSV (, ; tab space): , (choose the comma ",")

- First, the program will draw the raw phase dispersion of S_{21} , as shown below. We must choose the proper frequency range for calculation of the delay time.



We suggest comparing the program graph with the VNA display to make sure you have prepared the S.csv file correctly.

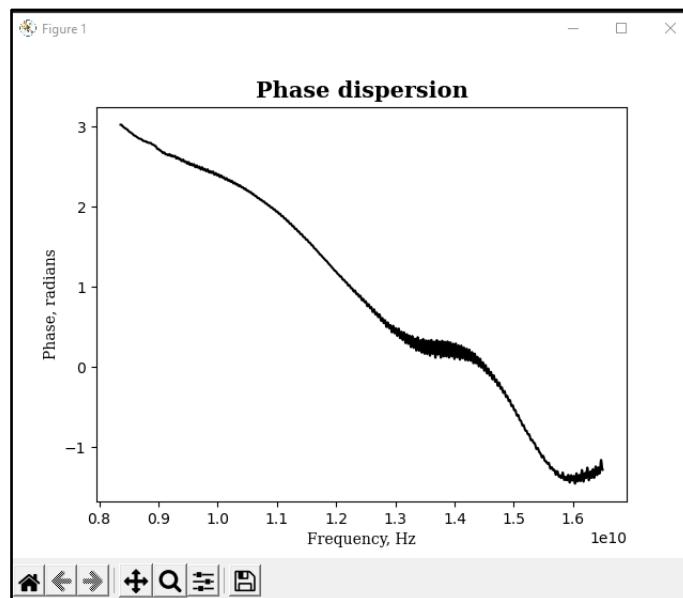


7. Choose the frequency range, as explained above, and close the graph. Then enter these values (use the science format “**.*e**”), each time pushing Enter.

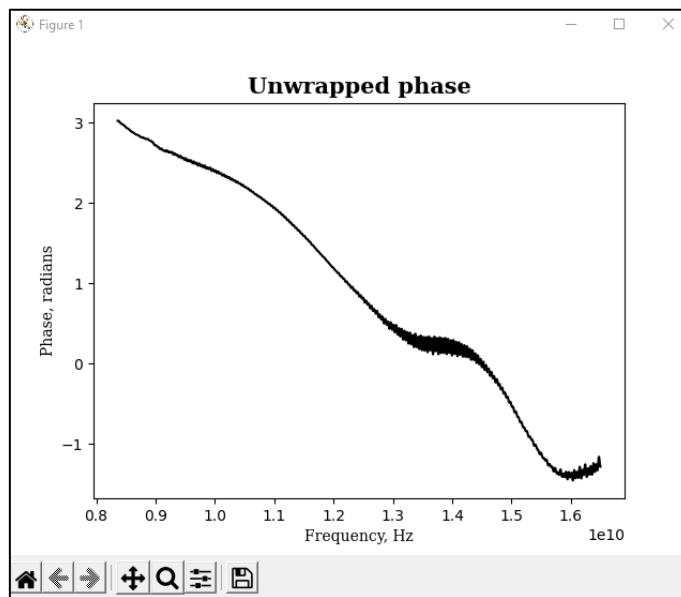
```
The initial phase has been saved in Phase_initial.csv

Enter the start frequency in Hz: 8.36e9
Enter the stop frequency in Hz: 1.65e10
```

The program will draw the selected part of the phase dispersion to check the correctness of the choice.



8. Close the previous graph. After that, the program will draw the next one with the so-called unwrapped phase, which will look the same if no jumps were included.

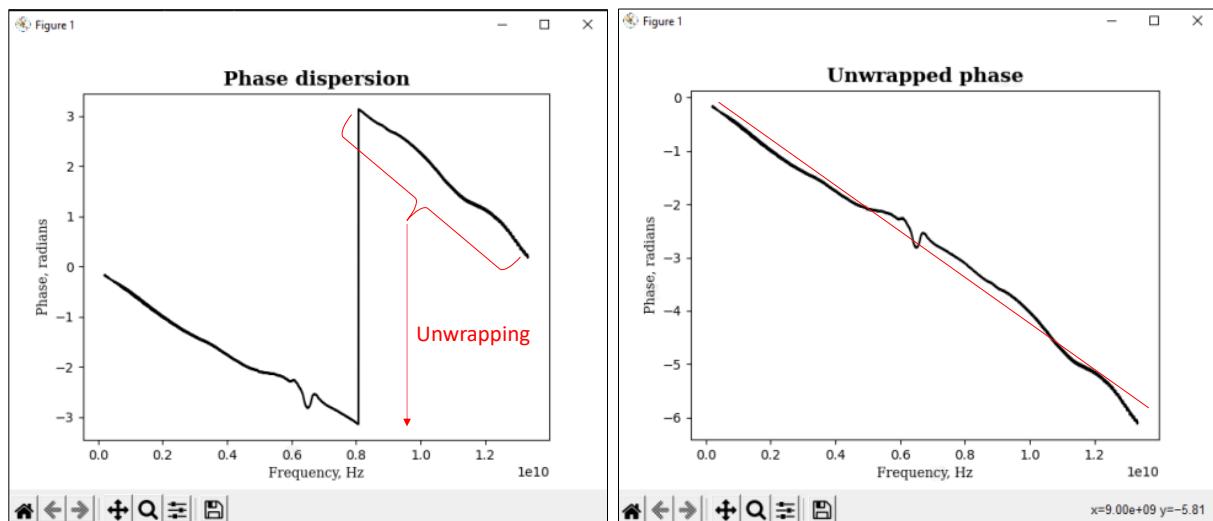
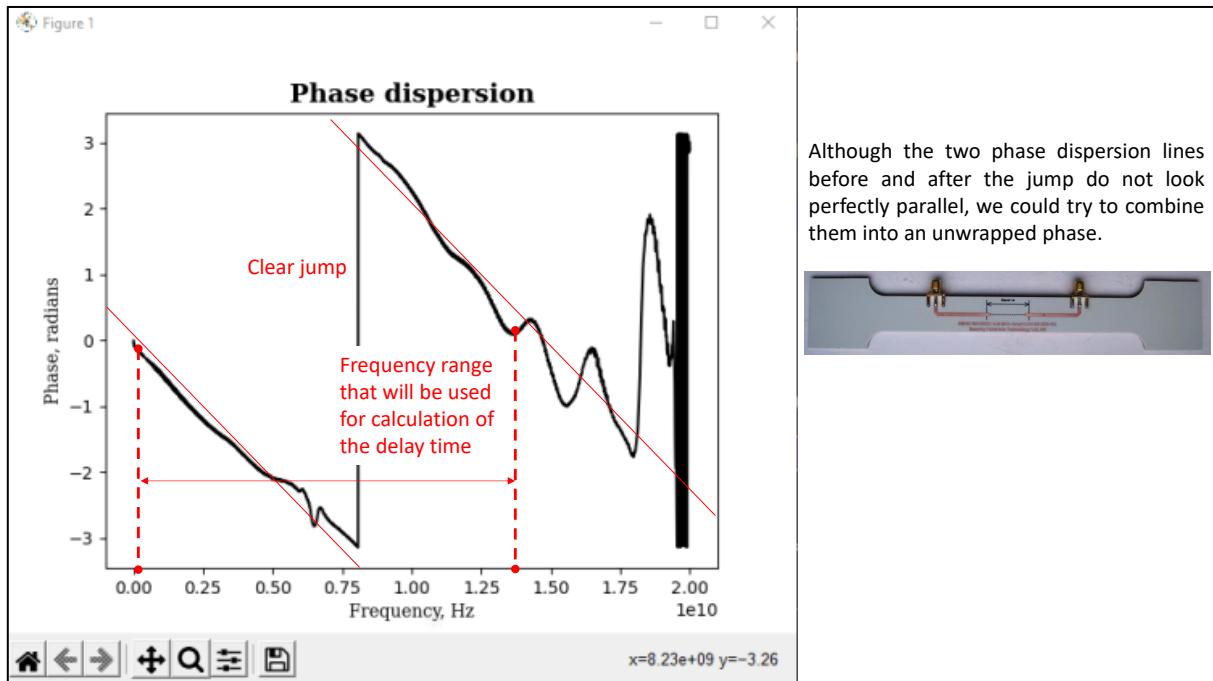


9. Close the previous graph. The program calculates the delay time Δt along the sample from the slope of the straight trend line: $\Delta t = |a|/(2\pi)$, where $\varphi \approx a \times f + b$ is the trend line for the unwrapped phase and f is the frequency. The value of b introduces a slight vertical shift in the trend line and does not play any role.

```
Enter the start frequency in Hz: 8.30e9
Enter the stop frequency in Hz: 1.65e10
Number of phase jumps =  0
The unwrapped phase has been saved in Phase_unwrapped.csv
Delay time =  92.01396295731027 ps
```

The delay time is used in our LabVIEW programs for measuring the magneto-impedance and the impedance dispersion. However, it needs to be further adjusted, as will be explained later. Note that the program also saves the raw and unwrapped phases in the CSV files.

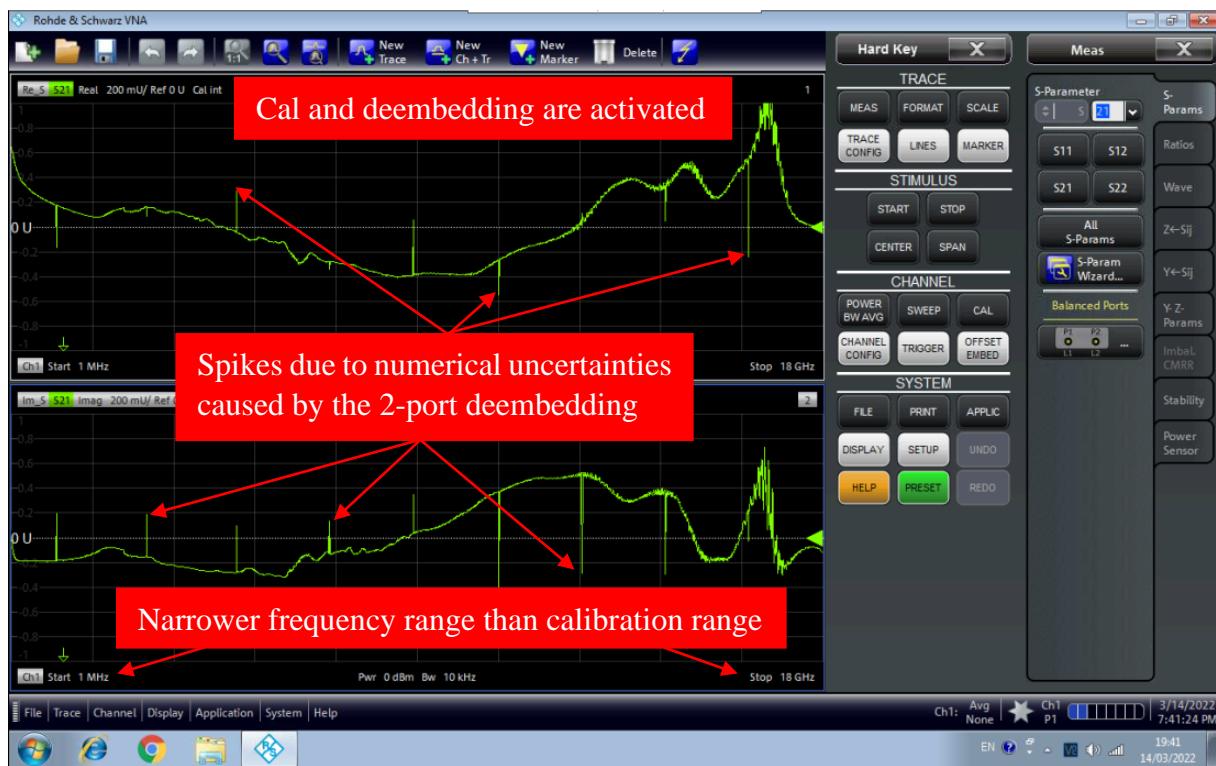
10. The quality of the phase dispersion depends on the design of the PCB cell. Quite unexpectedly, but we found that the dogbone cell, despite its more complex design, exhibits better phase dispersion up to 13 GHz or even higher, as shown below. For this dispersion, we could include the single middle jump.



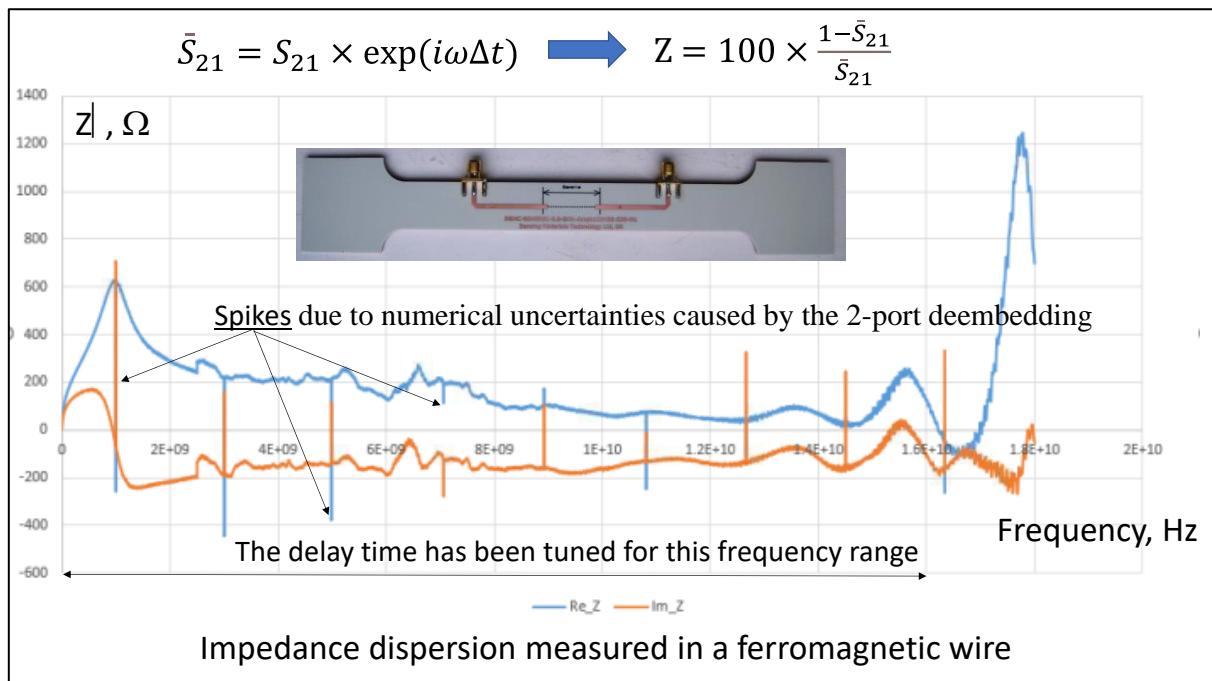
```

Enter the start frequency in Hz: 2.0e8
Enter the stop frequency in Hz: 1.333e10
Number of phase jumps = 1
The unwrapped phase has been saved in Phase_unwrapped.csv
Delay time = 66.48988296951843 ps
  
```

11. VNA has a function to estimate the delay time along a fixture. However, when trying to narrow the frequency range to select the correct part of the phase dispersion, spikes will appear due to numerical uncertainties caused by the 2-port deembedding. These spikes will prevent the phase from being unwrapped directly on the VNA. Thus, this function on VNA turns out to be useless in our case. We have to use an external algorithm.

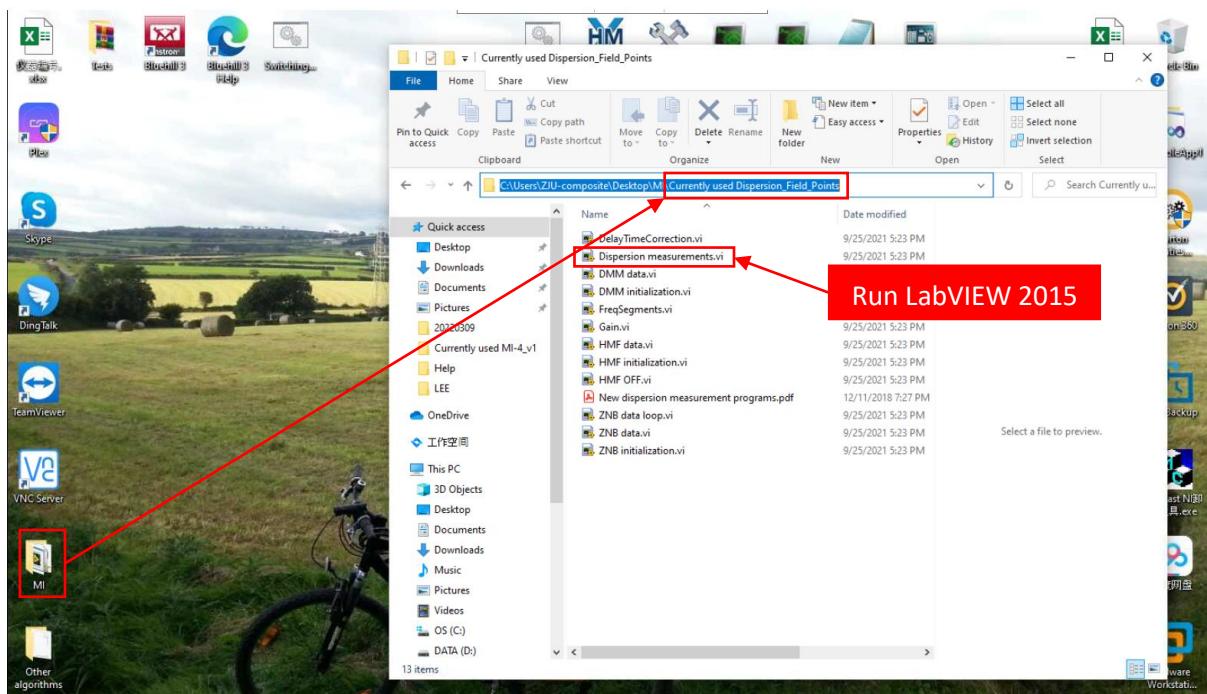


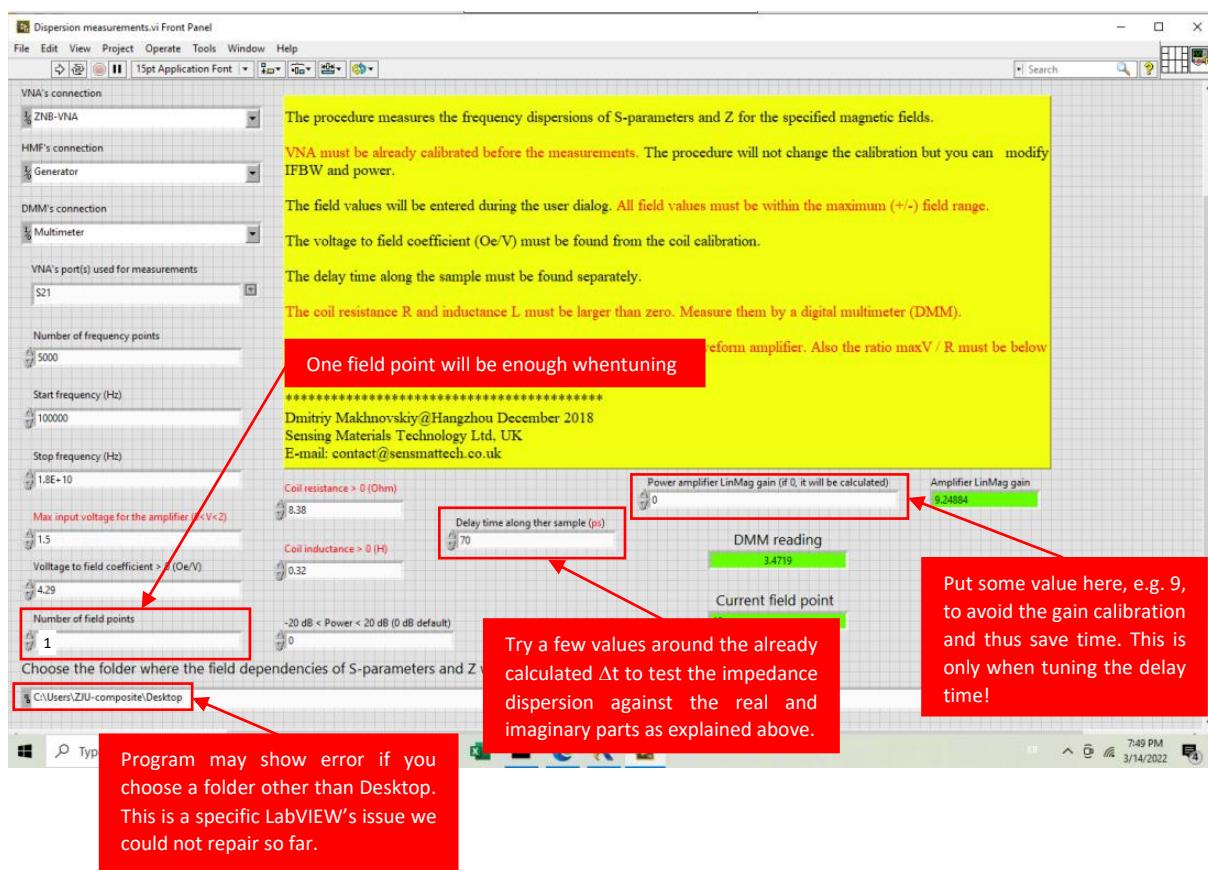
12. To further tune the delay time along the sample, we should use the impedance dispersion calculated from S_{21} . The main criterion for tuning is the positiveness of the real part of the impedance for the entire frequency range. In the case of a ferromagnetic wire exhibiting resonance, we can also control the imaginary part, which must be negative after resonance.



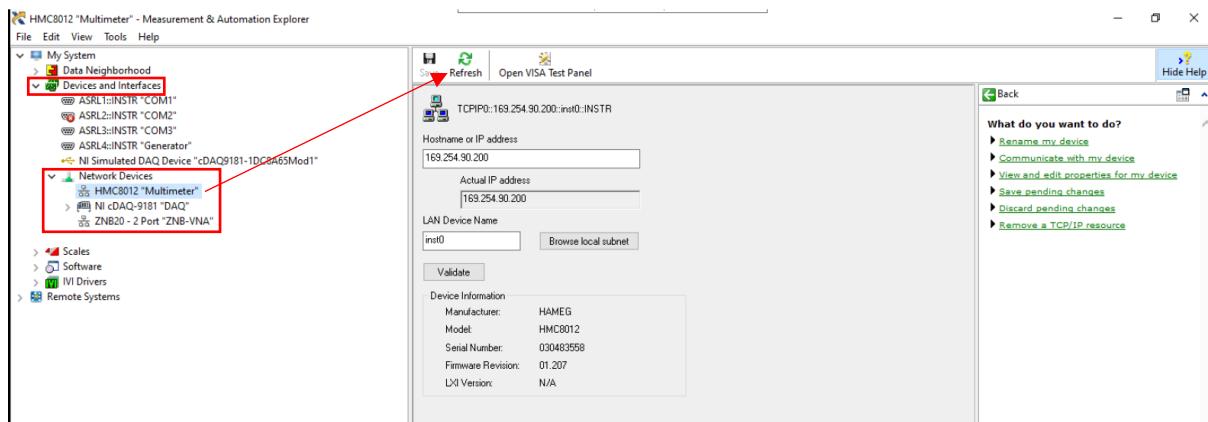
The spikes can be easily suppressed using a smoothing spline interpolation.

On the VNA's PC we have several LabVIEW programs to measure the impedance dispersion on a R&S VNA. You can choose any of them for tuning Δt : change slightly Δt around its value estimated by the Delay time algorithm.

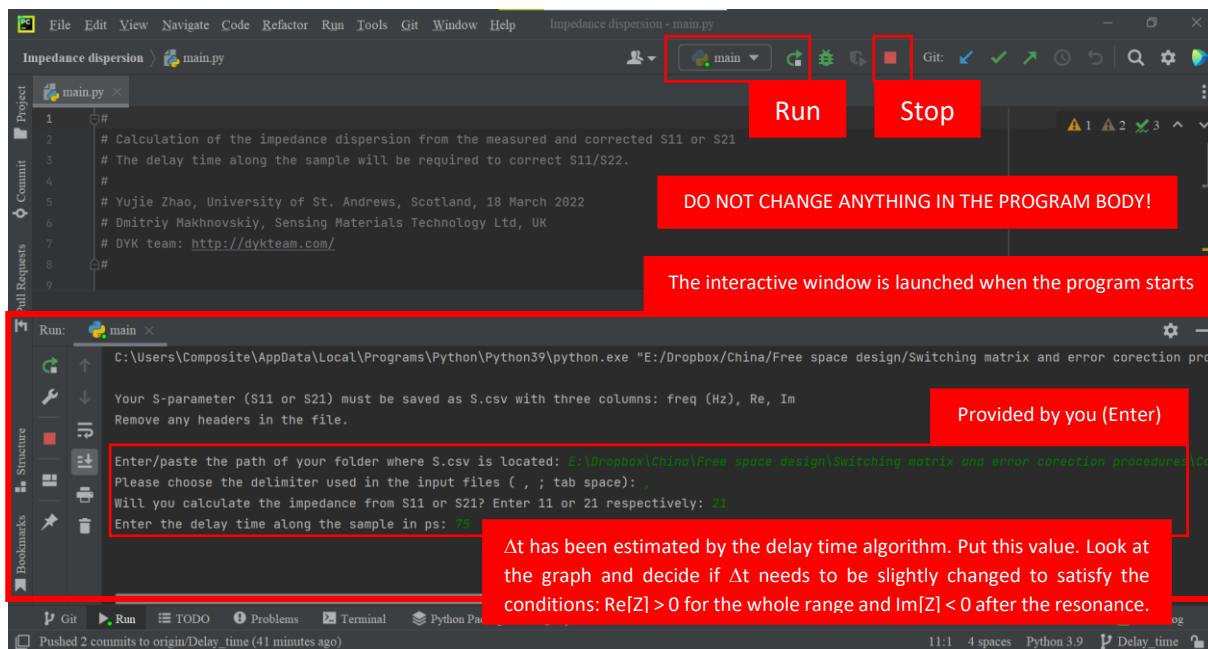




During measurements, communication with devices may be lost, most frequently with R&S Digital Multimeter (DMM). In this case, you can try to reconnect using NI MAX. Alternatively, switch off DMM, then switch on again, run NI MAX, and refresh Network Devices. Ask someone who has worked with our LabVIEW programs if you run into problems.

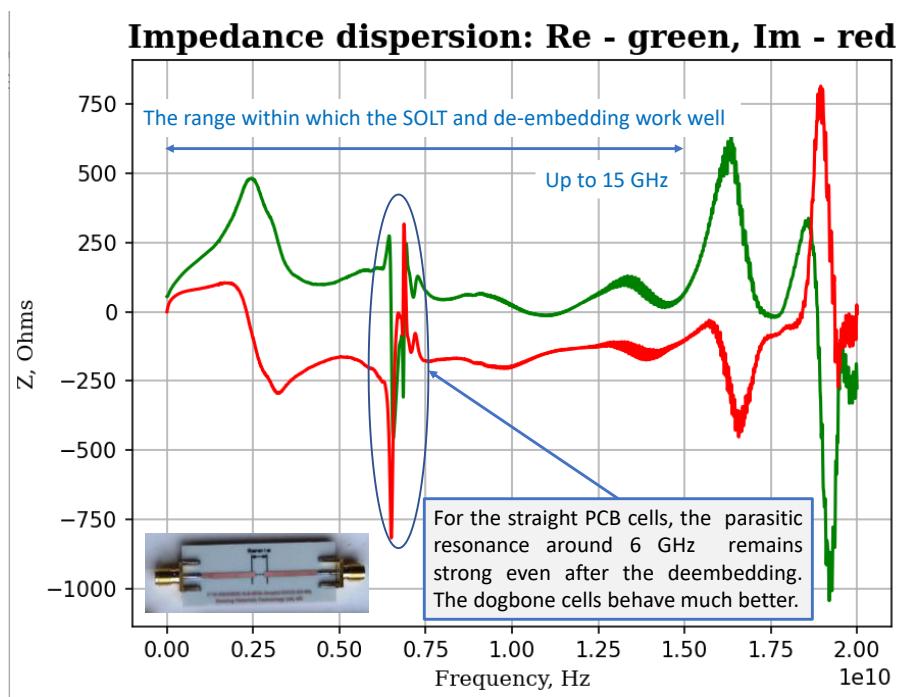


13. To simplify tuning the delay time along the sample, we have developed an algorithm in Python that calculates the impedance dispersion from S.csv (see (2)). The program has been saved on the VNA's PC (Desktop → Other algorithms → Impedance dispersion) and at GitHub in a public repository.[4] To open the program in IDE (we have installed PyCharm), click on **main.py** and then run it. Follow the instructions in the interactive window. Alternatively, you can open **main.py** by right-clicking and selecting “Open with → Python”.



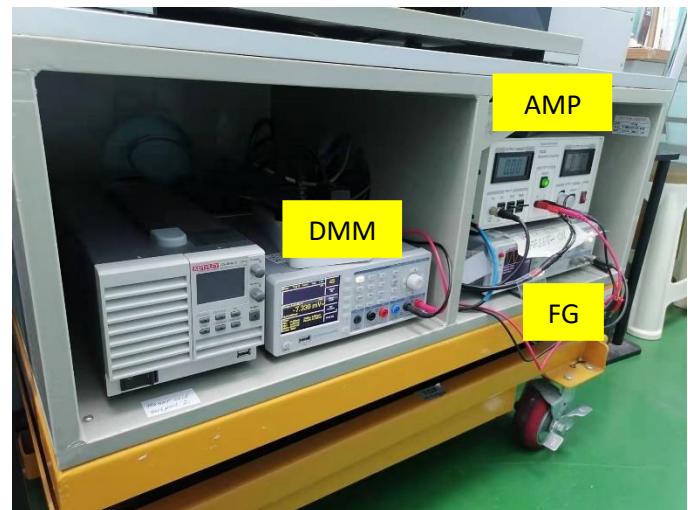
The program runs in a loop, each time showing the graph of impedance dispersion (see below) after the delay time has been entered. To enter a new value, close the graph. Try different values around the value estimated by the delay time algorithm that works with the same S.csv. After a few iterations, you will find a value that provides the correct impedance behaviour: $\text{Re}[Z] > 0$ for the whole frequency range and $\text{Im}[Z] < 0$ after the resonance (if a ferromagnetic wire is used). Then, stop the program.

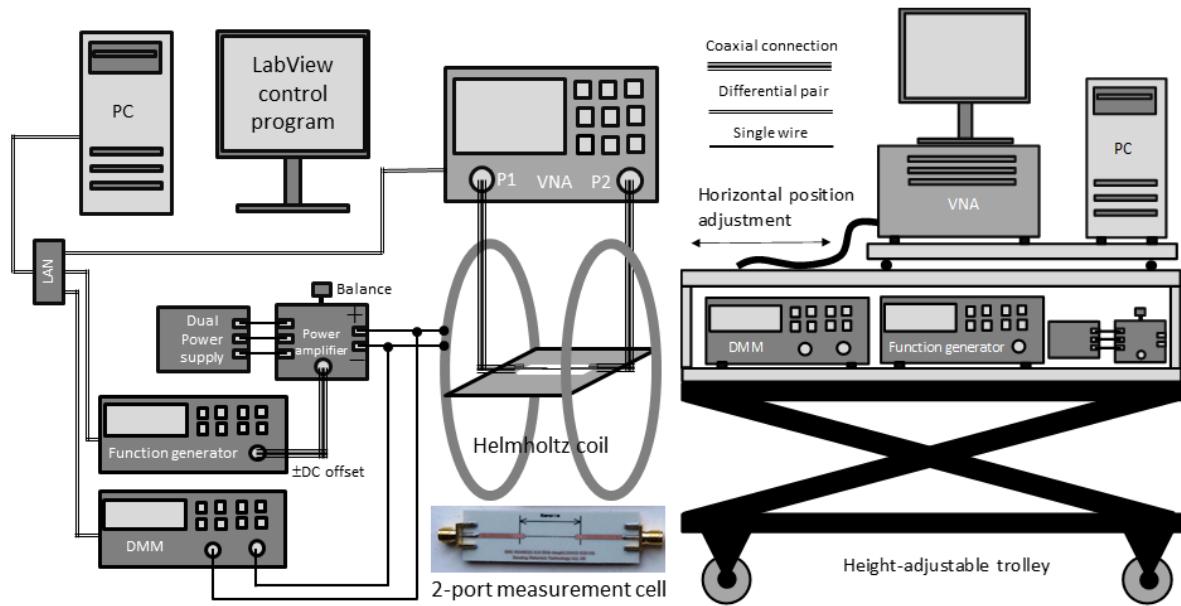
In the graph below the delay time has been already tuned. The SOLT calibration and the deembedding work well up to 15 GHz. However, in the case of straight PCB cells, we can still observe parasitic resonance around 6 GHz. Surprisingly, the more complex dogbone cells exhibit better dispersion behaviour. To minimize this kind of distortions, an optimised cell design is needed, especially when it comes to launching the SMA connectors.



Impedance measurements

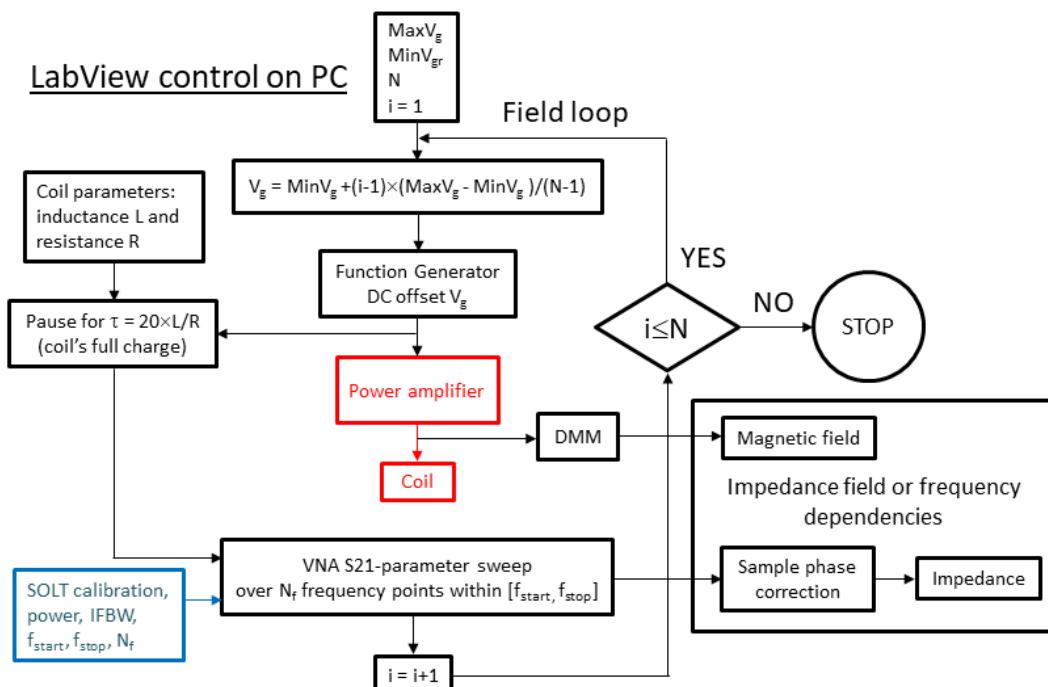
After three stages of correction (SOLT/TOSM → Deembedding the microstrips/fixtures → Compensation of the delay time along the sample) we are ready for impedance measurements which involve three R&S devices: **VNA** ZNB20 (2-port, 100 kHz – 20 GHz), HMF 2550 Function Generator (**FG**, 50 MHz), and HMC 8012 Digital Multimeter (**DMM**). We also use Accel Instruments Waveform Amplifier TS250-2 (**AMP**) that feeds a large Helmholtz coil. All devices are mounted on a trolley.

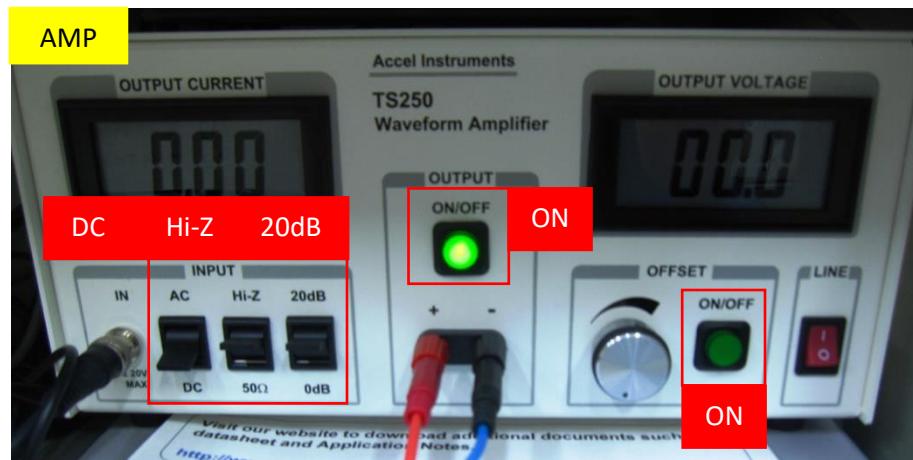




The operation of all the devices is synchronised by the programs created in LabVIEW 2015. Free space measurements use Python for algorithms and controlling devices (using SCPI) and Node-Red for GUI. In the future, we plan to transfer all software to Python with Node-Red interfaces. However, a newer version of LabVIEW starting from 2018 may be of interest as a platform for developing GUI to run Python codes. LabVIEW 2015 we use in our lab does not allow this.

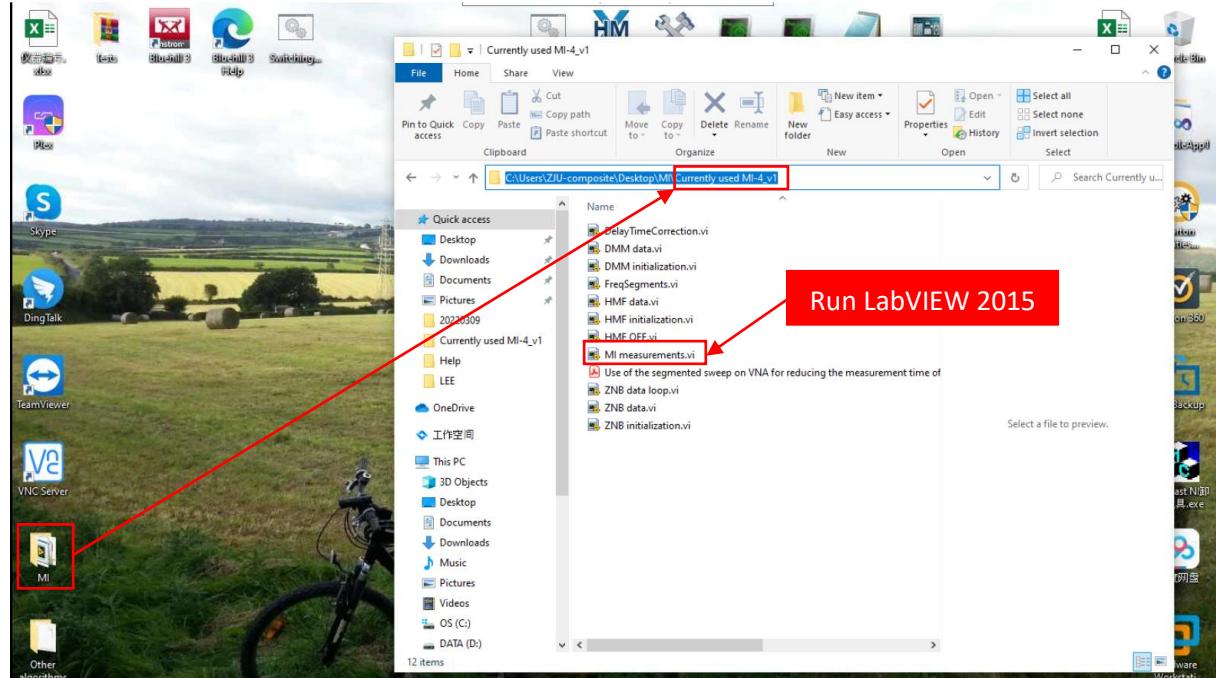
1. We carry measurements for two types of external stimuli: dependence of the impedance on (1) magnetic field or (2) both tensile stress and magnetic field. The block diagram below shows the system operation principle under the control of the LabVIEW program for measuring the field dependence of the impedance.

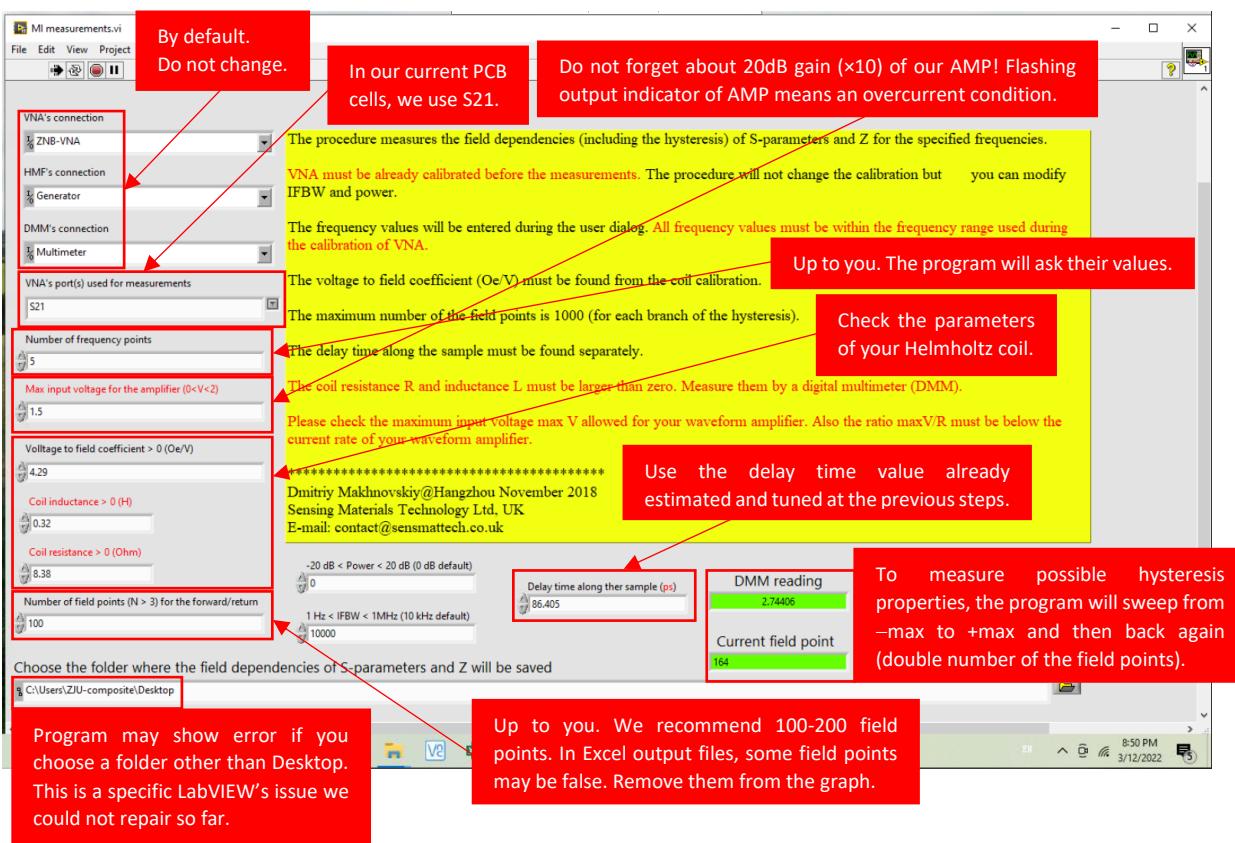




FG provides a DC bias voltage which is then passed to AMP before being applied to the coil. On the front panel of AMP we choose 20dB gain, but we found that its actual value is unstable. In any case, DMM measures the actual voltage applied to the coil after AMP. Switch ON Output and Offset. Looking at the DMM, use the offset adjuster to zero the output voltage, but a few \pm mV would be acceptable. If you noticed that the output indicator starts flashing while the AMP is running, it means an overcurrent condition. Turn off AMP and take measures to reduce the input voltage.

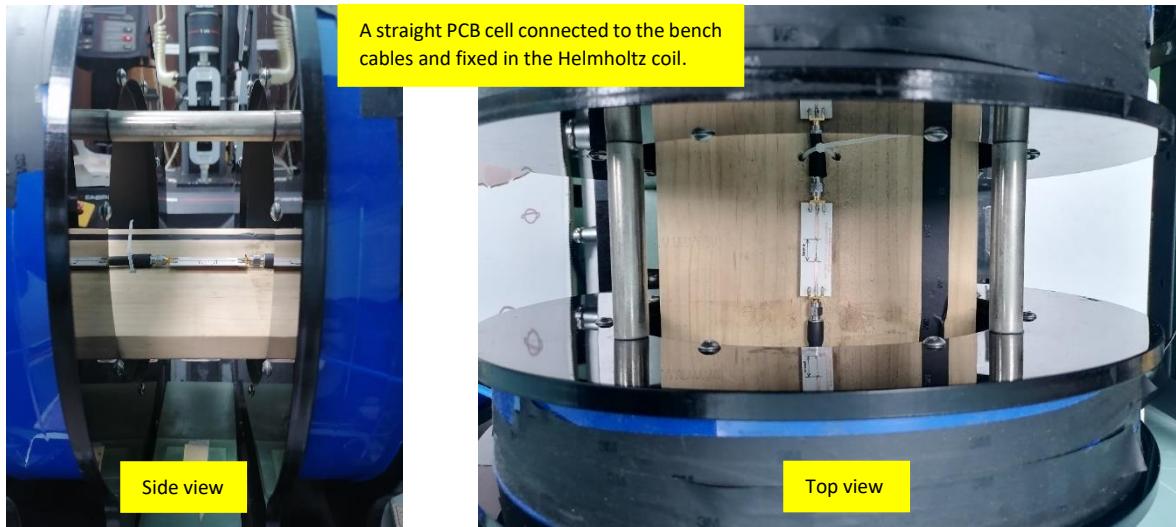
Magnetic field sweep at fixed frequency points is implemented by the latest version of the MI-4_v1 LabVIEW program. Click on the head module “MI measurements.vi” to run the program.





During measurements, communication with devices may be lost, most frequently with DMM. In this case, you can try to reconnect using NI MAX. Alternatively, switch off DMM, then switch on again, run NI MAX, and refresh Network Devices. Ask someone who has worked with our LabVIEW programs if you run into problems.

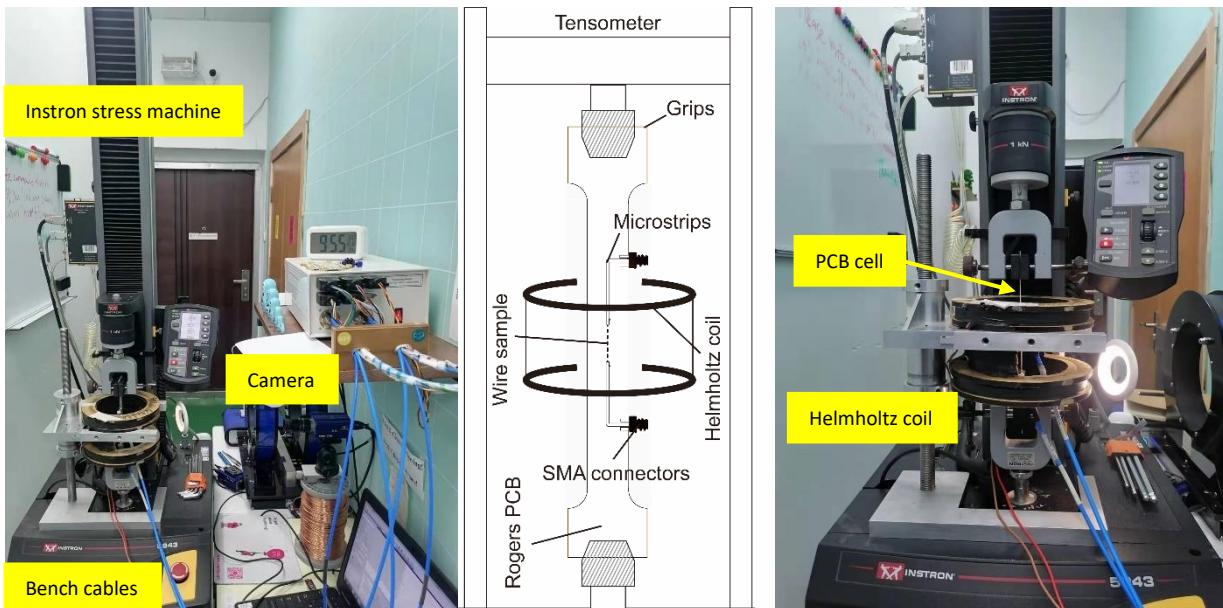
For measurements, you can use a straight or dogbone PCB cell. The latter is universal and can be used for both field and stress measurements. Moreover, we found that the dogbone cells demonstrate better dispersion properties without a parasitic resonance. The wire sample is connected to the microstrips of the cell using a conductive silver paint that has proven itself for high frequency applications. You could also try a conductive epoxy.



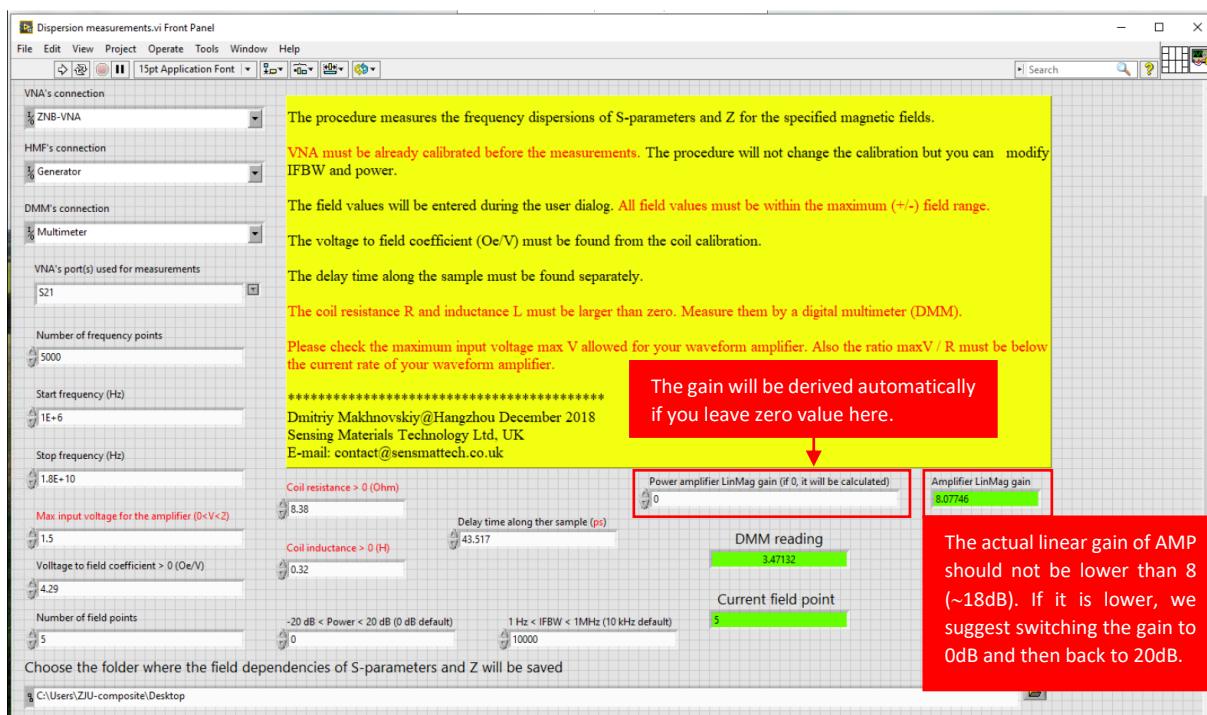
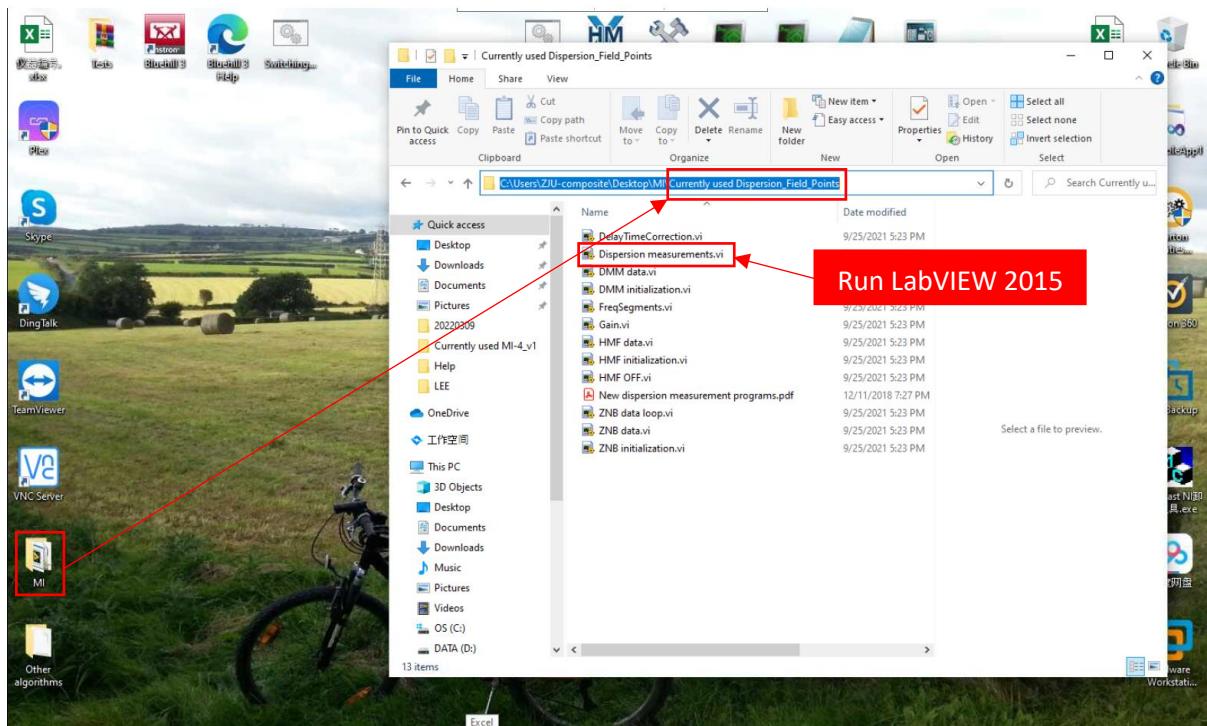


The program outputs are Excel files with S_{21} -parameters and impedances calculated from them with self-explanatory names. Save them in a different folder, as they will be overwritten when you start the program again.

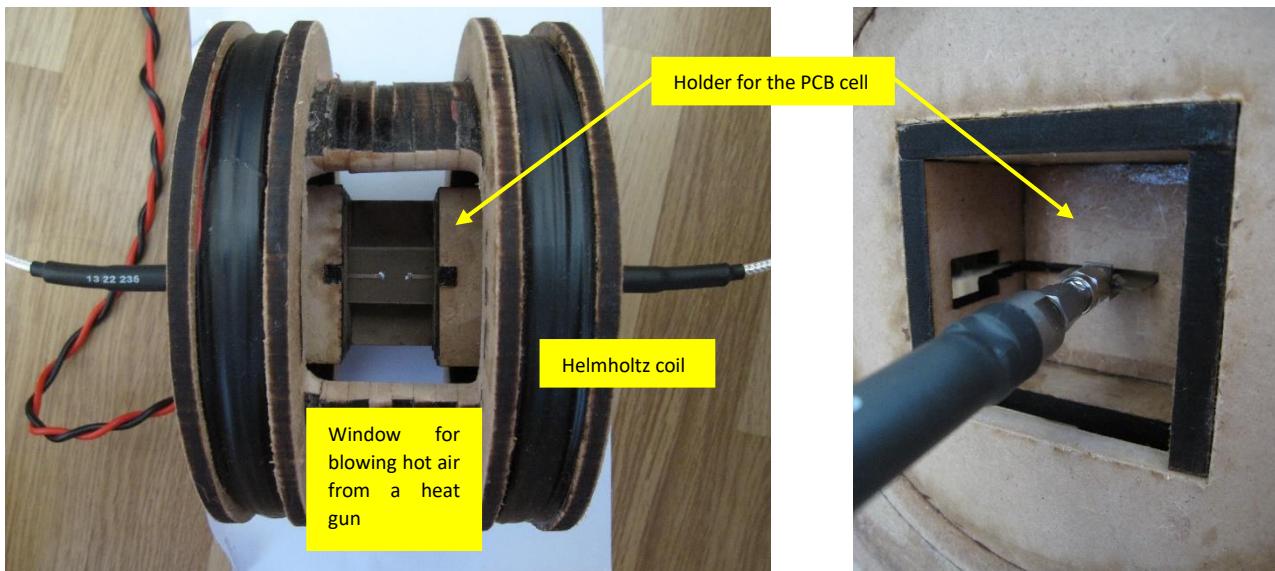
2. Stress-impedance measurements, see below, use the dogbone PCB cells and require a stress machine. We use a compact Instron on a trolley that can be moved closer to VNA. To measure strain, optical methods, extensometers, or strain gauges can be used. Currently, the operation of Instron is not synchronised with other devices of the trolley and hence must be controlled separately. With Bluehill Universal's Application Programming Interface (API), it should be possible to access the Instron's test methods from a programming language that is capable of connecting with Windows Connection Foundation (WCF). But we have not explored these possibilities yet.



For stress-impedance, we are usually interested in measuring the impedance dispersion for a fixed field and tensile stress. On the VNA's PC, we have several LabVIEW programs for measuring impedance dispersions with similar interfaces, for example one that shown below. In this program, the field points, the number of which you can choose, will be evenly distributed between zero and the maximum field calculated from the maximum output voltage (FG's voltage \times linear gain) and the coil's transfer coefficient Oe/V. In other LabVIEW programs, the field points can be chosen individually.



3. In addition to the magnetic field and tensile stress, it would be useful to investigate the temperature dependence of the impedance. In principle, the design of an apparatus that would include a Helmholtz coil, a small heating chamber, and temperature control is not difficult. But this project was not completed. All components required for such an installation are shown below.



Design of PCB measurement and calibration cells

From a microwave point of view, a PCB measurement cell must provide a good 50Ω matching over a wide frequency range and, if possible, be free from resonances. Also, we need to take care of the optimal return path and grounding (including vias), and the correct launching of the coaxial connectors. Along with microwave design issues, we have to provide additional functionalities associated with external stimuli: magnetic field, tensile stress, temperature, or bias current through the sample. Microstrips deliver high-frequency excitation directly to the sample. For their de-embedding, the measurement cell of a certain type must be supplemented with the corresponding calibration cell. The latter must have identical design of the microstrips that are terminated with SHORT, OPEN, and LOAD standards for creating their full S-parameter model. The termination standards, which are not coaxial, should be broadband and possessing minimal parasitic inductance and capacitance. At high frequencies in the gigahertz range, we face the problem of radiation losses.

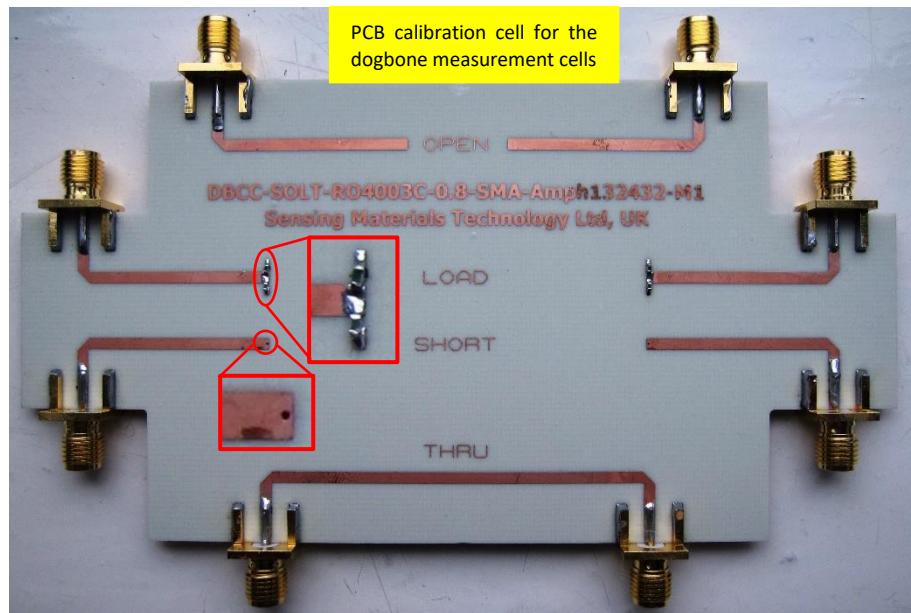
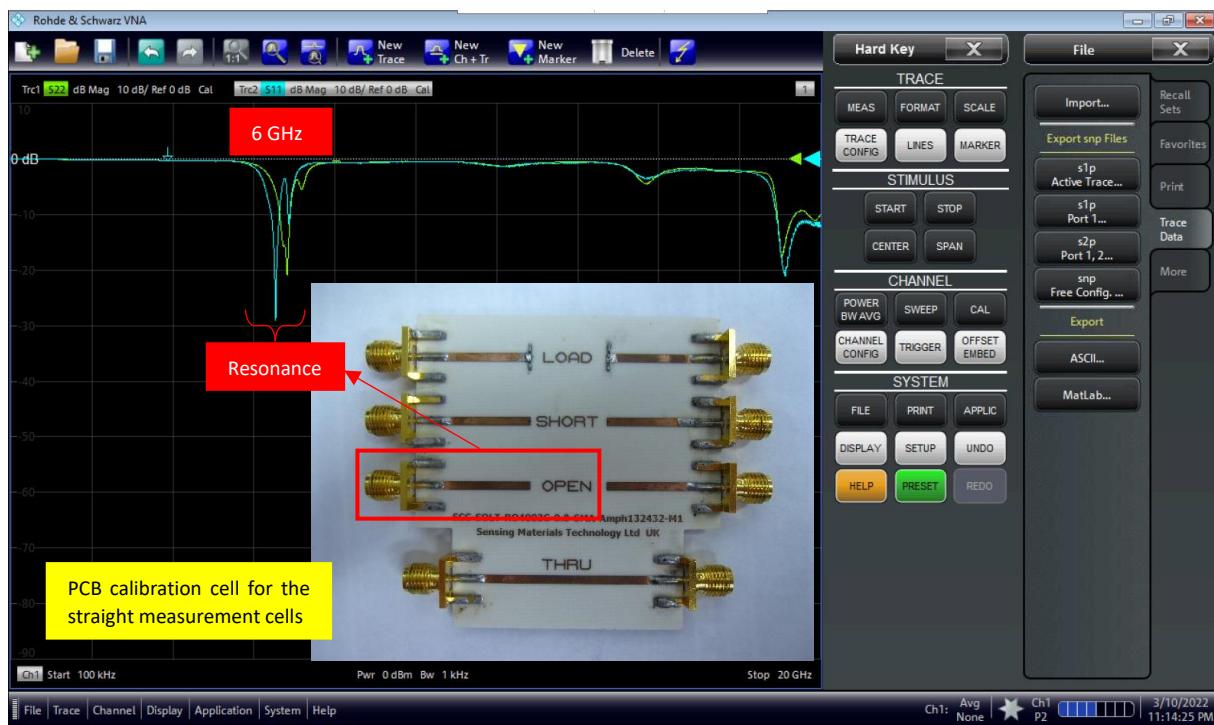
We have tried to fulfil the criteria mentioned above when designing two types of PCB cells. We chose a Rogers's dielectric (RO4003-C, 0.812 mm thick, $D_k = 3.55$; <https://www.rogerscorp.com/>) as the board material. Rogers' PCBs are widely used in microwave engineering due to their low loss and low thermal drift of electrical and mechanical properties. The Rogers' microwave impedance calculator

<https://www.globalcommhost.com/rogers/acs/techsupporthub/en/calculatorMWI.php>

is a useful tool for calculating the geometrical parameters of the microstrip waveguide structure. Launching the coaxial connectors (e.g. SMA) onto the board is an important element of any high-frequency board design. Here, we should follow some practical rules which are not covered by microwave calculators.

In general, the design of our PCB cells can be considered successful, with the exception of one problem associated with the presence of parasitic resonance in the region of 6 GHz for the straight cells. This resonance is clearly visible when measuring $S_{11/22}$ from the open end of the microstrip, see below, and remains after the sample is connected. Surprisingly, this resonance is practically absent in the dogbone cells, all the difference of which lies only in the 90° curved microstrips. We are not completely sure what the reason for this resonance is. This once again shows that microstrip calculators are not quite useful when it comes to parasitic parameters.

In the PCB cells where it is necessary to apply a DC bias current through the sample, surface mounted decoupling capacitors soldered in series with the microstrips will be required to isolate the sample from VNA's ports. A sample equipped with electronic microchips can already be a complete device, for which our cells are also suitable for testing. In this case, S-parameters (transfer functions) will be of primary interest, not the impedance per se.



For deembedding a microstrip, we need to recover its S-parameters and save them in S2P file. This can be done using the 3-term error model described later in this guide. The model will require measurement of S_{11} -parameter of the microstrip when it is terminated with SHORT, OPEN, and LOAD standards. The PCB calibration cells we used are shown above, where SHORT is a via (copper plated hole) to the ground plane on the reverse side of PCB, OPEN is just non-terminated microstrip, and LOAD is two 100Ω surface mounted resistors connected in parallel to the ground plane through vias. The THRU standard is currently not used. The microstrips on the PCB calibration cells have the same geometry and dimensions as on the measurement cells.

Creation of S2P files for new PCB cells

The development of a new measurement PCB cell will be accompanied by an appropriate PCB calibration cell. The cells may be symmetrical or asymmetrical depending on your requirements. It is necessary to measure the reflection coefficients ($S_{11/22}$) from microstrips terminated with SOL standards and save them as the following CSV files: **MS11S.csv** (SHORT), **MS11O.csv** (OPEN), **MS11L.csv** (LOAD) and **MS22S.csv**, **MS22O.csv**, **MS22L.csv**. Choose the comma delimiter. If the cell is symmetrical, only one set would be enough (11 or 22). Each file must include three columns: frequency (Hz), Real part, and Imaginary part. Remove any headers in the file and leave only numbers. When measuring the reflection coefficients, VNA and the bench cables must be already calibrated, and the calibration enabled.

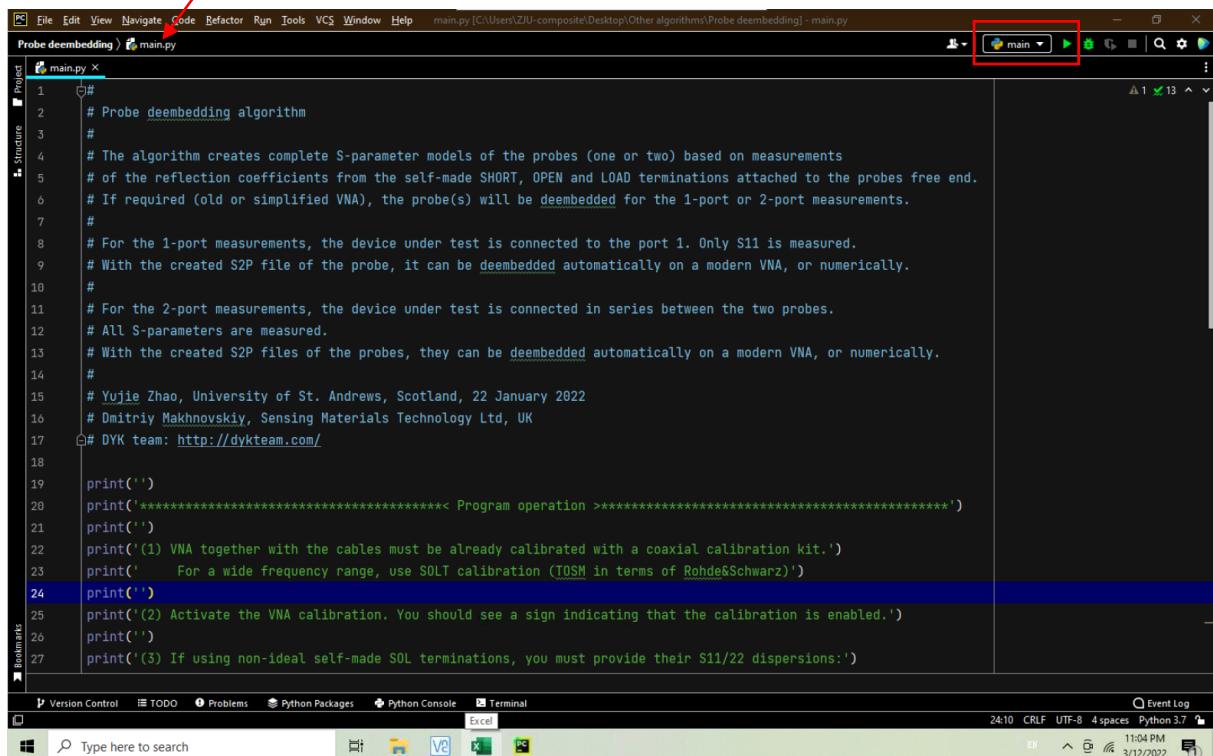
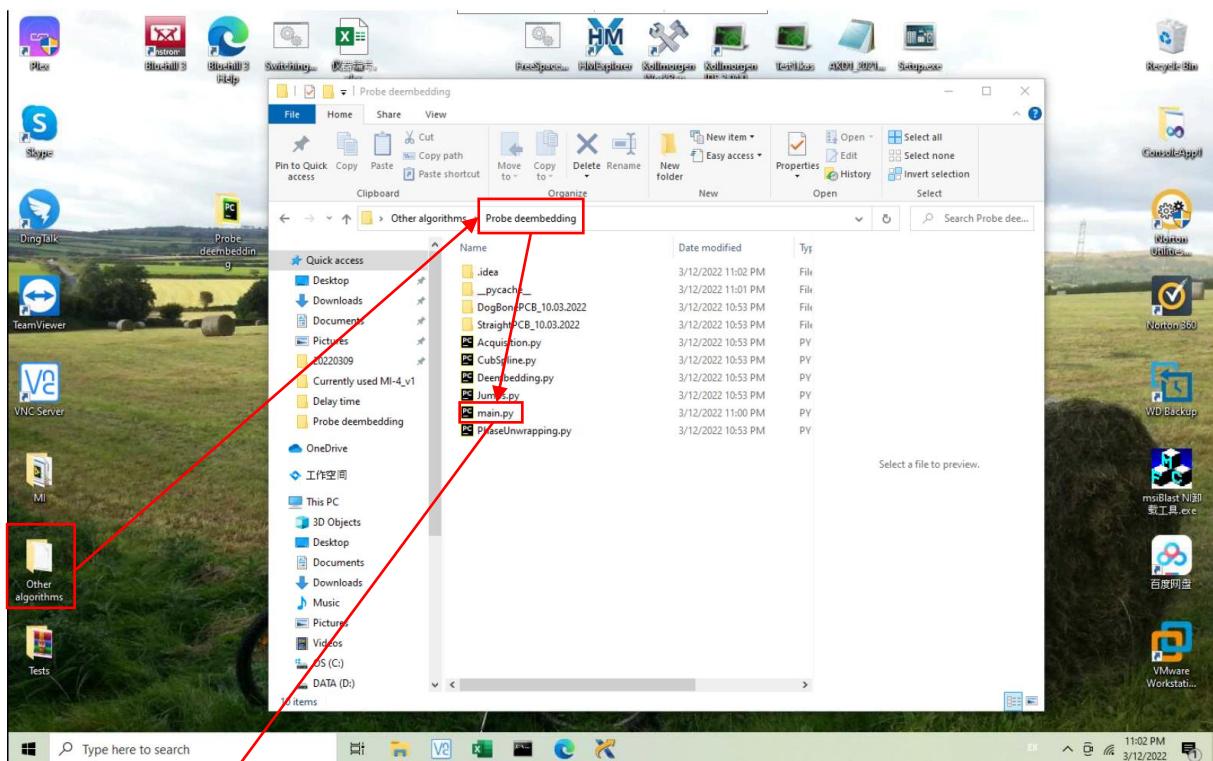


Use these operations to save S-parameters measured on VNA. The file will contain 9 columns, but we need only S11 and S22. All headers must be removed.

We have developed an algorithm in Python that creates S2P files from the measured MS11S.csv, MS11O.csv, MS11L.csv, MS22S.csv, MS22O.csv, and MS22L.csv. The algorithm, called “Probe deembedding”, has been saved on the VNA’s PC and at GitHub in a public repository.[4] Run the program (see below) and follow the instructions in the interactive window. The program will create S2P files (choose “tab” delimiter for the output files as required by R&S VNA) that must be saved under unique names and uploaded to the VNA’s folder:

C:\Users\Public\Documents\Rohde-Schwarz\Vna\Deembedding\Bench_Cables_PCB_Cells

Do not change the header in S2P files – it is functional!



Alternatively, you can open **main.py** by right-clicking and selecting “Open with → Python”.

In next section we provide a detailed report explaining the algorithm principle. There, the fixtures (microstrips) are called “probes”. Despite the fact that in the report we considered the cable probes, the principle of deembedding remains the same.

Appendix 1: Measurements with RF probes. A technical report.

Composed by Yujie Zhao, University of St. Andrews, Scotland.

A two-port network, schematically shown in Fig. 1, takes in the input signals $a_{1,2}$ and then transforms them to the output signals $b_{1,2}$. The operation of a two-port linear network can be described by the following linear relations that can be written in the matrix form:

$$\begin{cases} b_1 = S_{11}a_1 + S_{12}a_2 \\ b_2 = S_{21}a_1 + S_{22}a_2 \end{cases} \Rightarrow \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad (1)$$

where

$$\begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} = \hat{\mathbf{S}} \quad (2)$$

is the S-matrix or scattering matrix. Its components S_{ij} (complex coefficients depending on the frequency) are called the S-parameters which are defined as the following ratios: $S_{11} = (b_1/a_1)_{a_2=0}$ and $S_{22} = (b_2/a_2)_{a_1=0}$ (reflection coefficients), $S_{12} = (b_1/a_2)_{a_1=0}$ and $S_{21} = (b_2/a_1)_{a_2=0}$ (transmission coefficients).



Fig. 1. Input and output signals for a 2-port network.

The design of measurement schemes will require the LCR values of the circuit components, including the pickup coils. They can be accurately measured using the matched RF probes and a Vector Network Analyser (VNA). Figure 2 shows two possible probe designs based on: (a) coaxial connector and (b) piece of coaxial cable with loose wires at its end. Other designs can be also proposed.

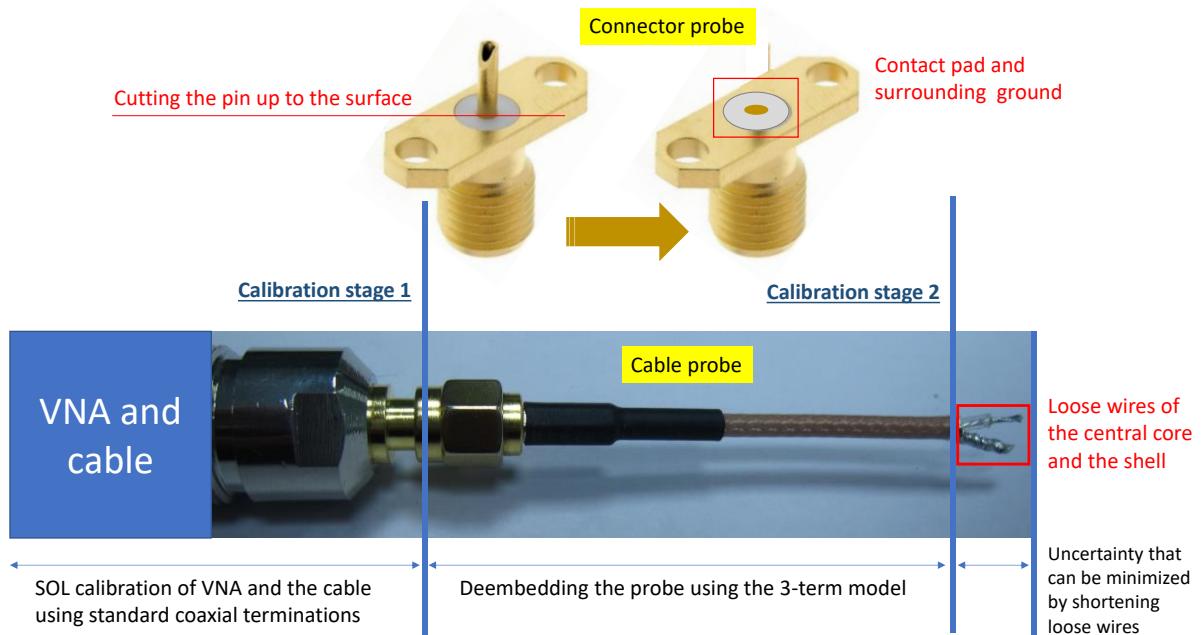


Fig. 2. Examples of the RF probes (connector- and cable-based) for measuring LCR parameters of electronic components.

For correct measurements, VNA must be calibrated to extend the reference plane as close as possible to the device under test (capacitor, inductor, or resistor for our applications). VNA and the cable connecting it to the probe can be calibrated together using a coaxial calibration kit and the standard 1-port SOL (3-term model with SHORT, OPEN, and LOAD terminations) or 2-port SOLT (12-term model with the additional THRU standard) error correction procedure. It will constitute the calibration stage 1 as shown in Fig. 2. At the calibration stage 2, the reference plane is extended further up to the probe output. The stage 2 is realized by means of virtual deembedding of the probe using its full S-parameter model. It can be recovered by measuring S_{11} -parameters from the probe terminated in sequence with some well-defined non-coaxial SOL standards as shown schematically in Fig. 3. Note that the probe's S_{11} -parameters are measured with VNA and the cable already calibrated at the stage 1.

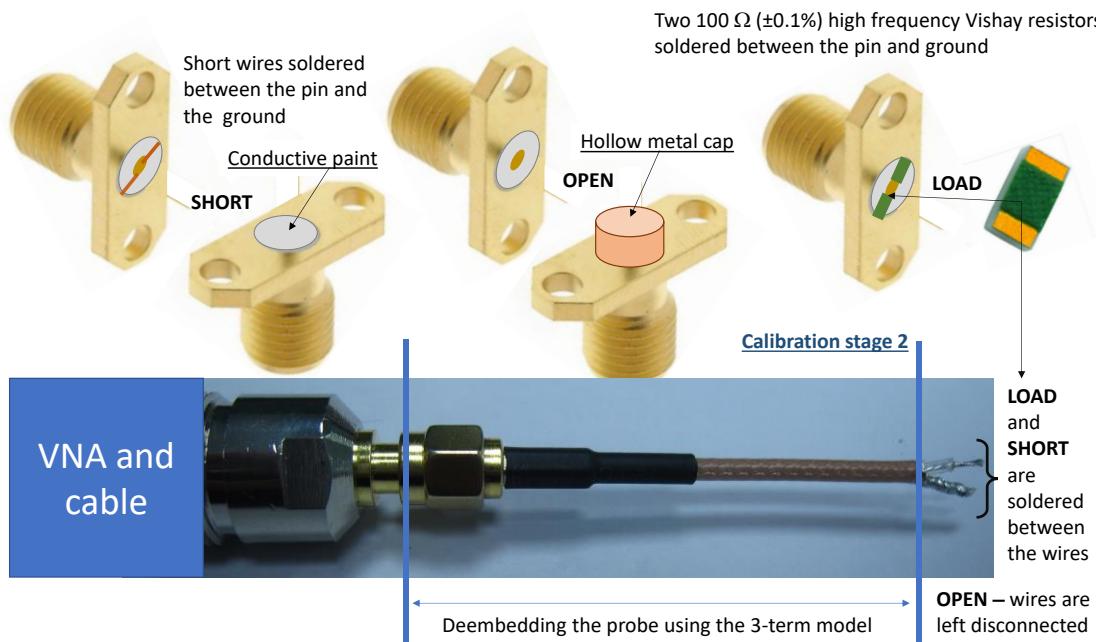


Fig. 3. Connector- and cable-based probes terminated with non-coaxial SOL standards for measuring S_{11} -parameters required for the 3-term model.

The SOL standards can be implemented by several methods:

Connector-based probe

- **SHORT.** One or several copper wires connected between the central pin and the ground. Alternatively, the entire cross-section of the connector can be covered with a conductive paint thus connecting the central pin with the peripheral ground. Up to several hundred MHz, such terminations should demonstrate almost ideal reflection coefficient $S_{11S} = -1$. At higher frequencies, a short connection to ground will inevitably introduce some stray inductance. Also, the skin-effect in the short connection may need to be considered at GHz frequencies that makes the resistance and reactance frequency dependent. In addition, a stray capacitance may become noticeable. Before the LC resonance, the reactance from L will prevail. If the probe output is not sealed with the total ground (e.g. a layer of conductive paint), the open capacitor will become radiative at higher GHz frequencies thus contributing to losses. In the theory of error corrections in microwave networks, the effects described above are modelled by lumped LC-parameters that depend polynomially on frequency (usually polynomials up to the third power are used in VNAs). Creating such models is a separate task. CST studio may prove to be an effective tool for modelling physical properties of the SOLT terminations.
- **OPEN.** The probe output is left open. Up to several hundred MHz, such termination should demonstrate almost ideal reflection coefficient $S_{11O} = 1$. For higher frequencies, OPEN can be characterized by a lumped capacitance. Radiation losses will introduce some effective resistance in

series. Radiation losses could be excluded by placing a hollow metal cap on the output, but it may increase the effective capacitance or introduce some cavity resonances.

- LOAD. High frequency surface mount resistors can be used, for example, provided by VISHAY company. Up to several hundred MHz, such termination should demonstrate almost ideal $S_{11L} = 0$. To bridge the gap between the central conductor and the ground in our connector probes, we need a resistor with the length slightly larger than 1.5 mm. VISHAY case size 0603 with the length of 1.626 mm would be appropriate:

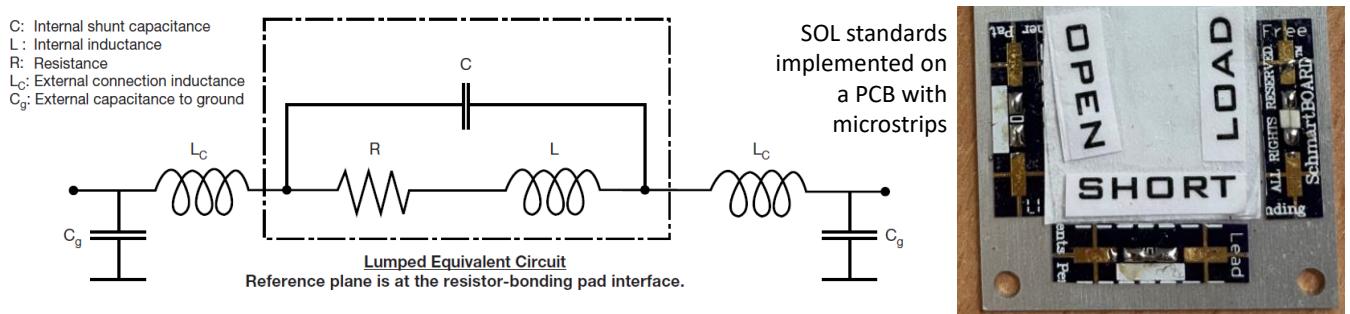
<https://www.mouser.co.uk/ProductDetail/Vishay-Thin-Film/FC0603E1000BTBST1?qs=DyUWGjl%252BcVvADnInzbYSg%3D%3D>

VISHAY provides accurate LCR models for their high frequency resistors, see Fig. 4. To further reduce the stray inductance, we should use two 100Ω resistors connected in parallel to ground. The probe opening will still introduce some stray capacitance and radiate at higher frequencies. However, these effects can be neglected in the MHz range.

Cable-based probe

- SHORT. The probe's loose wires (which must be kept as short as possible) are soldered together. Up to several hundred MHz, such terminations should demonstrate almost ideal reflection coefficient $S_{11S} = -1$. At higher frequencies, a short loop will inevitably introduce some stray inductance. Also, the skin-effect may need to be considered at GHz frequencies that makes the resistance and reactance frequency dependent.
- OPEN. The probe's loose wires are left disconnected. Up to several hundred MHz, such termination should demonstrate almost ideal reflection coefficient $S_{11O} = 1$. For higher frequencies, OPEN can be characterized by a lumped capacitance.
- LOAD. Two VISHAY resistors can be soldered in parallel to the loose wires.

For convenience, the standards can be mounted on a PCB, as shown in Fig. 4. The strips should be kept as short as possible to minimize parasitic parameters. We have implemented this approach.



The lumped circuit above was used to model the data at the bonding pad-resistor reference plane. High frequency testing was performed by Modelithics, Inc. on parts mounted to quartz test boards. Quartz test boards were chosen to minimize the contribution of the board effects at high frequencies.

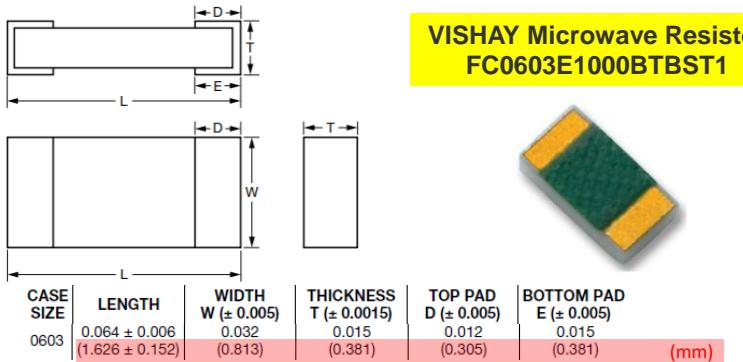


Fig. 4. Dimensions of the VISHAY case size 0603 high frequency resistor and its LCR model.

To virtually deembed a fixture attached (through a cable or directly) to an already calibrated VNA, it is necessary to create its full S-parameter model. Let us recall the 3-term error model, shown in Fig. 5, which is used to recover all S-parameters of the probe. The model comprises three error parameters: E_{DF} (directivity forward), E_{RF} (reflection tracking forward), and E_{SF} (source mismatch forward). Here, we use the terminology accepted for the analysis of errors in VNAs. The interpretation of the error parameters in terms of the S-parameters of a 2-port network is shown in Fig. 5. As we will see, in the 3-term model S_{21} and S_{12} are met only as the product $S_{21} \times S_{12}$. That is why $E_{RF} = S_{21} \times S_{12}$ is used instead of S_{12} , while S_{21} is assumed to be 1.

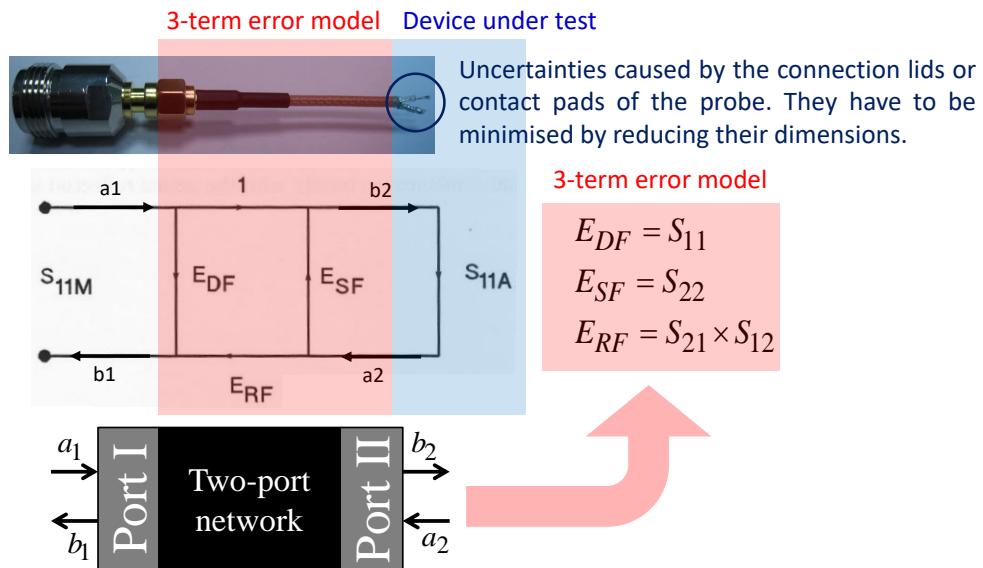


Fig. 5. 3-term error model used for the 1-port deembedding.

The actual reflection S_{11A} produced by the device under test (DUT) is buried into S_{11M} measured from the probe. By measuring three additional reflections when the probe is being consequently connected to SHORT, OPEN, and LOAD, it becomes possible to find E_{DF} , E_{RF} , and E_{SF} , and then recover S_{11A} . The signals in the 3-term model, see Fig. 5, satisfy the following system of linear equations:

$$\begin{cases} b_1 = a_1 E_{DF} + a_2 E_{RF} \\ b_2 = a_1 + a_2 E_{SF} \\ a_2 = b_2 S_{11A} \end{cases} \quad (3)$$

Excluding a_2 and b_2 , we obtain:

$$S_{11M} = \frac{b_1}{a_1} = E_{DF} + \frac{E_{RF} S_{11A}}{1 - S_{11A} E_{SF}} \quad (4)$$

Using (4), S_{11A} can be expressed through S_{11M} :

$$S_{11A} = \frac{S_{11M} - E_{DF}}{E_{RF} + E_{SF}(S_{11M} - E_{DF})} \quad (5)$$

Eq. (5) allows one to recover the actual reflection coefficient from the measured one. In (5), we have three unknowns E_{DF} , E_{RF} , and E_{SF} which can be found from S_{11M} measured from three known termination standards SHORT (S_{11S}), OPEN (S_{11O}), and LOAD (S_{11L}):

$$\begin{cases} S_{11MS} = E_{DF} + \frac{E_{RF} S_{11S}}{1 - S_{11S} E_{SF}} \\ S_{11MO} = E_{DF} + \frac{E_{RF} S_{11O}}{1 - S_{11O} E_{SF}} \\ S_{11ML} = E_{DF} + \frac{E_{RF} S_{11L}}{1 - S_{11L} E_{SF}} \end{cases} \quad (6)$$

The system of equations (6) can be rewritten in the following form:

$$\begin{cases} E_{DF} + S_{11MS}S_{11S}E_{SF} + S_{11S}(E_{RF} - E_{DF}E_{SF}) = S_{11MS} \\ E_{DF} + S_{11MO}S_{11O}E_{SF} + S_{11O}(E_{RF} - E_{DF}E_{SF}) = S_{11MO} \\ E_{DF} + S_{11ML}S_{11L}E_{SF} + S_{11L}(E_{RF} - E_{DF}E_{SF}) = S_{11ML} \end{cases} \quad (7)$$

which is a system of linear equations with respect to the three unknowns $x = E_{DF}$, $y = E_{SF}$, and $z = (E_{RF} - E_{DF}E_{SF})$.

Let us write (7) in a matrix form:

$$\begin{pmatrix} 1 & S_{11MS}S_{11S} & S_{11S} \\ 1 & S_{11MO}S_{11O} & S_{11O} \\ 1 & S_{11ML}S_{11L} & S_{11L} \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} S_{11MS} \\ S_{11MO} \\ S_{11ML} \end{pmatrix} \quad (8)$$

Solving (8) numerically, we will find the probe's complex S-parameters $S_{11P} = E_{DF} = x$, $S_{22P} = E_{SF} = y$, and $S_{21P} = S_{12P} = \sqrt{E_{RF}} = \sqrt{z + E_{DF}E_{SF}}$. The latter square root is calculated as $\sqrt{|z + E_{DF}E_{SF}|} \times \exp(i\varphi/2)$, where φ is the unwrapped phase $\text{atan}2(\text{Im}(z + E_{DF}E_{SF}), \text{Re}(z + E_{DF}E_{SF})) \in]-\pi, \pi[$. This phase, when unwrapped, must have a negative slope. The actual reflection coefficient S_{11A} from the device under test can be recovered using (5) with already calculated error parameters. For further details, referrer to the module **Deembeding.py** file in the algorithm folder on [GitHub](#).

Up to several hundred MHz (our case in the present project), the reflections from the terminations can be assumed to be ideal and dispersionless: $S_{11S} \equiv -1$, $S_{11O} \equiv 1$, and $S_{11L} \equiv 0$. Already in the lower GHz range, the dispersion may become noticeable, and hence these coefficients must be considered as the functions of frequency. The terminations in commercial calibration kits are usually supplied with neat LCR models where capacitance and inductance versus frequency are fitted by third degree polynomials. These polynomial dependences are caused by the skin-effect (in inductive elements) and growing radiation losses (in capacitive elements). A delay time may be also taken into account, especially for the THRU standard.

In addition to this analytical description, in modern VNAs it is possible to set the reflection coefficients of the standard terminations through matrices consisting of three columns: discrete frequency values covering a wide range, and the real and imaginary parts of the reflection coefficient for each frequency. Such matrices are saved in the text files of the special format called the Touchstone data: http://www.ibis.org/touchstone_ver2.0/touchstone_ver2_0.pdf. If required, intermediate values can be calculated using a spline interpolation.

Touchstone data files consist of an “option line” followed by network parameters taken at specific frequencies. The option line specifies (among other things) the kind of n-port parameters the file contains (S-parameter, Z-parameter, etc.) and the format of the network data values (magnitude-phase, real-imaginary, etc.). Data is arranged into groups of n-port parameters preceded by the frequency at which the data was taken or derived. For each frequency, data for a 1-port or 2-port network is contained on a single data line. Option

line parameters are separated by one or more whitespace; the option line itself is terminated with a line termination sequence or character. If a parameter is missing, the default value is assumed.

With the exception of the opening # (hash mark) symbol and the value following “R”, option line parameters may appear in any order:

- For 1-port files (S1P): # [Hz|kHz|MHz|GHz] [S|Y|Z] [DB|MA|RI] [R n]
- For 2-port files (S2P): # [Hz|kHz|MHz|GHz] [S|Y|Z|G|H] [DB|MA|RI] [R n]

Here, [Hz|kHz|MHz|GHz] are the frequency units, [S|Y|Z] are parameters, [DB|MA|RI] are the data formats (dB, linear magnitude and angle, real and imaginary parts), [R n] specifies the reference resistance in ohms, where n is a real, positive number of ohms. The default reference resistance is 50 ohms. Minimum required option line example, using all default values:

```
# Hz S RI R 50
```

where frequency in Hz, S-parameters in real-imaginary format, referenced to 50 ohms. For the THRU standard, all four S-parameters must be provided.

In modern VNAs, a fixture attached to a port cable after its standard coaxial calibration can be virtually deembedded if the full S-parameter model of the fixture is known. Let us explain this procedure. Fig. 6 shows 2-port linear networks cascaded together. Linear networks connected in series can be represented as a single “effective” network, however its S-matrix will not be a product of S-matrices of the constituting networks.

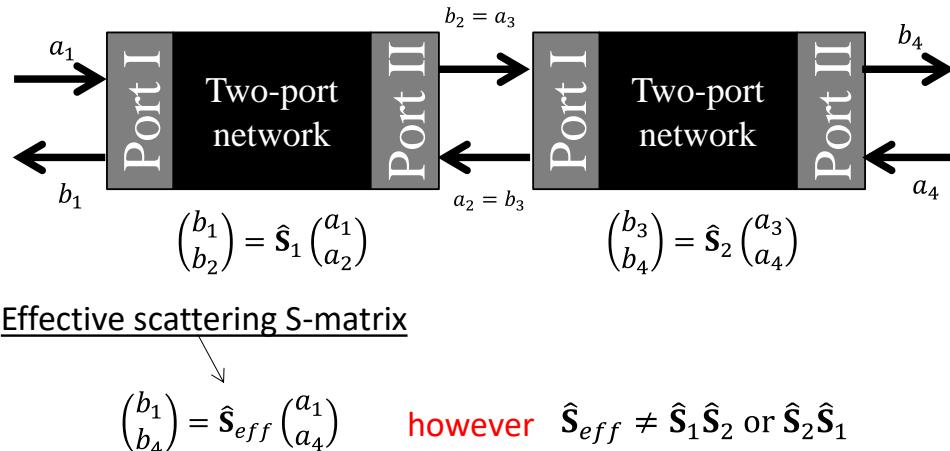


Fig. 6. Two 2-port networks are cascaded to form a new 2-port network. However, S-matrices do not possess the cascading property.

However, there is a matrix characteristic, called the propagation P-matrix, which possesses the cascading property:

$$\begin{pmatrix} b_2 \\ a_2 \end{pmatrix} = \widehat{\mathbf{P}} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \quad (9)$$

where the matrix components can be expressed through the S-parameters

$$\widehat{\mathbf{P}} = \begin{pmatrix} \frac{S_{12}S_{21} - S_{22}S_{11}}{S_{12}} & \frac{S_{22}}{S_{12}} \\ -\frac{S_{11}}{S_{12}} & \frac{1}{S_{12}} \end{pmatrix} = \frac{1}{S_{12}} \begin{pmatrix} S_{12}S_{21} - S_{22}S_{11} & S_{22} \\ -S_{11} & 1 \end{pmatrix} \quad (10)$$

Let us note that in (9) as the result of action of the operator $\widehat{\mathbf{P}}$ on the vector $\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$ we obtain $\begin{pmatrix} b_2 \\ a_2 \end{pmatrix}$, and not $\begin{pmatrix} a_2 \\ b_2 \end{pmatrix}$.

For cascading networks, the matrix of the effective network is the product of matrices of the individual networks constituting it, as shown in Fig. 7.

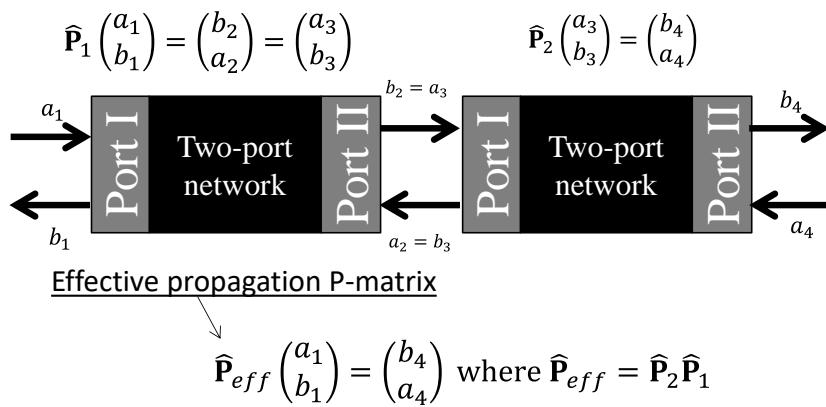


Fig. 7. Cascaded 2-port networks and their total P-matrix.

S-matrix can be expressed through the components of P-matrix:

$$\widehat{\mathbf{S}} = \begin{pmatrix} -\frac{P_{21}}{P_{22}} & \frac{1}{P_{22}} \\ P_{11} - \frac{P_{12}P_{21}}{P_{22}} & \frac{P_{12}}{P_{22}} \end{pmatrix} \quad (11)$$

Along with the transmission “from the left to the right” in (9), P-matrix for the transmission from “the right to the left” can be introduced as well:

$$\widehat{\mathbf{P}}^{-1} \begin{pmatrix} b_2 \\ a_2 \end{pmatrix} = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \quad (12)$$

$$\widehat{\mathbf{P}}^{-1} = \begin{pmatrix} \frac{1}{S_{21}} & -\frac{S_{22}}{S_{21}} \\ \frac{S_{11}}{S_{21}} & \frac{S_{12}S_{21} - S_{22}S_{11}}{S_{21}} \end{pmatrix} \quad (13)$$

Such definition of the transmission matrix appears useful if $S_{12} = 0$, e.g. in the systems where signals can propagate only in one direction.

For one port measurements with a fixture (probe), the reflection coefficient of the device under test S_{11A} (“actual”) can be found from (9) and Fig. 5:

$$\left\{ \begin{array}{l} \begin{pmatrix} b_2 \\ a_2 \end{pmatrix} = \hat{\mathbf{P}} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \Rightarrow \\ \Rightarrow \begin{cases} S_{12P}b_2 = (S_{12P}S_{21P} - S_{22P}S_{11P})a_1 + S_{22P}b_1 = \begin{cases} S_{12P}b_2 = (E_{RF} - E_{SF}E_{DF})a_1 + E_{SF}b_1 \Rightarrow \\ S_{12P}a_2 = -S_{11P}a_1 + b_1 \end{cases} \\ S_{12P}(b_2/a_1) = (E_{RF} - E_{SF}E_{DF}) + E_{SF}S_{11M} \\ S_{12P}(a_2/a_1) = -E_{DF} + S_{11M} \end{cases} \Rightarrow \frac{a_2}{b_2} = S_{11A} = \frac{-E_{DF} + S_{11M}}{(E_{RF} - E_{SF}E_{DF}) + E_{SF}S_{11M}} \end{array} \right. \quad (14)$$

The obtained equation for S_{11A} coincides with (5). This is what a VNA does when deembedding a fixture for the 1-port measurements. In the case of 2-port measurements with fixtures, modern VNAs use cascading P-matrices, as shown in Fig. 8, for deembedding:

- Finish a standard coaxial calibration of VNA, including the cables before the fixtures.
- Provide the S-parameter models for the fixtures. These models must be saved as S2P text files (see above) and uploaded to VNA.
- VNA transforms S-matrices of the port fixtures into the corresponding P-matrices (see (10)).
- VNA measures all S-parameters of the combined network, including the device under test connected between the fixtures.
- VNA transforms the measured S-matrix into P-matrix.
- VNA conducts calculations for the cascaded networks and finds P-matrix of the device under test.
- VNA transforms P-matrix of the device under test into the corresponding S-matrix.

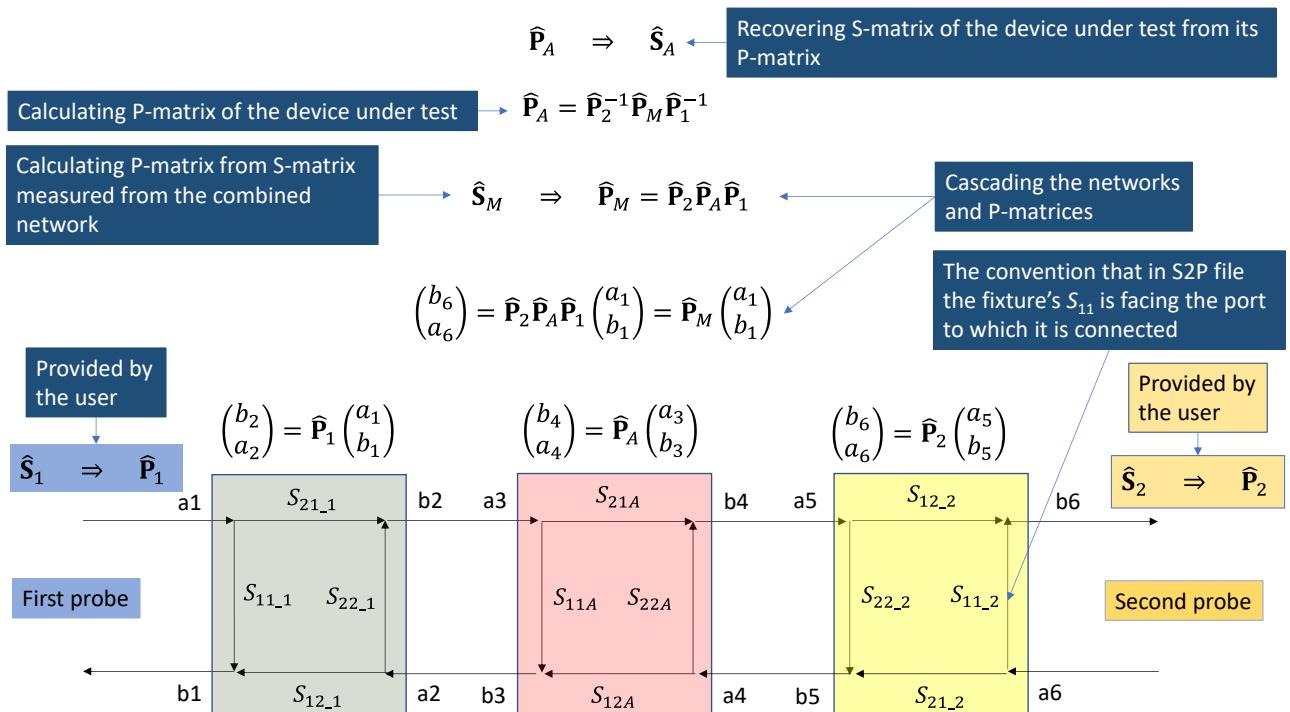


Fig. 8. Deembedding procedure for the 2-port measurements with two probes which may be different.

In Fig. 8:

$$\widehat{\mathbf{P}}_M = \frac{1}{S_{12M}} \begin{pmatrix} S_{12M} \times S_{21M} - S_{22M} \times S_{11M} & S_{22M} \\ -S_{11M} & 1 \end{pmatrix} \quad (15)$$

$$\widehat{\mathbf{P}}_1 = \frac{1}{S_{12_1}} \begin{pmatrix} S_{12_1} \times S_{21_1} - S_{22_1} \times S_{11_1} & S_{22_1} \\ -S_{11_1} & 1 \end{pmatrix} \quad (16)$$

$$\widehat{\mathbf{P}}_1^{-1} = \frac{1}{S_{21_1}} \begin{pmatrix} 1 & -S_{22_1} \\ S_{11_1} & S_{12_1} \times S_{21_1} - S_{22_1} \times S_{11_1} \end{pmatrix} \quad (17)$$

$$\widehat{\mathbf{P}}_2 = \frac{1}{S_{21_2}} \begin{pmatrix} S_{12_2} \times S_{21_2} - S_{22_2} \times S_{11_2} & S_{11_2} \\ -S_{22_2} & 1 \end{pmatrix} \quad (18)$$

$$\widehat{\mathbf{P}}_2^{-1} = \frac{1}{S_{12_2}} \begin{pmatrix} 1 & -S_{11_2} \\ S_{22_2} & S_{12_2} \times S_{21_2} - S_{22_2} \times S_{11_2} \end{pmatrix} \quad (19)$$

$$\widehat{\mathbf{P}}_A = \begin{pmatrix} P_{11A} & P_{12A} \\ P_{21A} & P_{22A} \end{pmatrix} = \begin{pmatrix} \frac{S_{12A} \times S_{21A} - S_{22A} \times S_{11A}}{S_{12A}} & \frac{S_{22A}}{S_{12A}} \\ \frac{S_{12A}}{-S_{11A}} & \frac{1}{S_{12A}} \end{pmatrix} \quad (20)$$

$$\widehat{\mathbf{S}}_A = \begin{pmatrix} & -\frac{P_{21A}}{P_{22A}} & \frac{1}{P_{22A}} \\ \frac{P_{11A}}{P_{22A}} - \frac{P_{12A} \times P_{21A}}{P_{22A}} & \frac{P_{12A}}{P_{22A}} & \end{pmatrix} \quad (21)$$

Notice how S_{11} and S_{22} are swapped in $\widehat{\mathbf{P}}_2$ as compared to $\widehat{\mathbf{P}}_1$ according to the definition in Fig. 8. This is due to the convention (by default in a R&S VNA) that in a S2P file the fixture's S_{11} is facing the port to which it is connected. The matrices (15) – (21) are used in the numerical algorithm in the module **Deembeding.py**.

With a 1-port measurement (VNA's Port I or II), after performing SOL calibration of VNA together with the cable and then deembedding the probe, the complex impedance of the device under test can be determined without distortions:

$$Z_A = 50 \times \frac{1+S_{11A}}{1-S_{11A}} [\Omega] \quad (22)$$

When a 2-port device is connected in series between two probes, as shown in Fig. 9, its impedance can be determined from S_{21A} :

$$Z_A = 100 \times \frac{(1-S_{21A})}{S_{21A}} [\Omega] \quad (23)$$

The transmission coefficient S_{21A} must be calculated according to the 2-port deembedding procedure in Fig. 8. This will require the measurement of all S_M -parameters, although in (23) only S_{21A} is used. Strictly speaking, recalculating S_{21A} into Z_A only makes physical sense if the transmission coefficient is not related to any wave propagation. For a larger device at higher frequencies, the internal wave propagation may be involved making the interpretation in terms of impedance less straightforward. Consider, for example, a ferromagnetic wire sample placed between two microstrips on the top of a PCB with a continuous copper ground on the opposite side. Such a structure constitutes a waveguide that may introduce a noticeable delay time at higher frequencies. Therefore, it cannot be interpreted as a lumped impedance element. However, the purely impedance properties of the wire sample can be extracted by compensating for the phase effect in S_{21A} using the so-called phase

unwrapping technique. Such corrected S_{21A} can be already used in (23) for calculating the wire impedance that includes the internal resistance and inductance. This method has been proposed in Ref. [2] to extract the wire surface impedance used in the antenna equation.

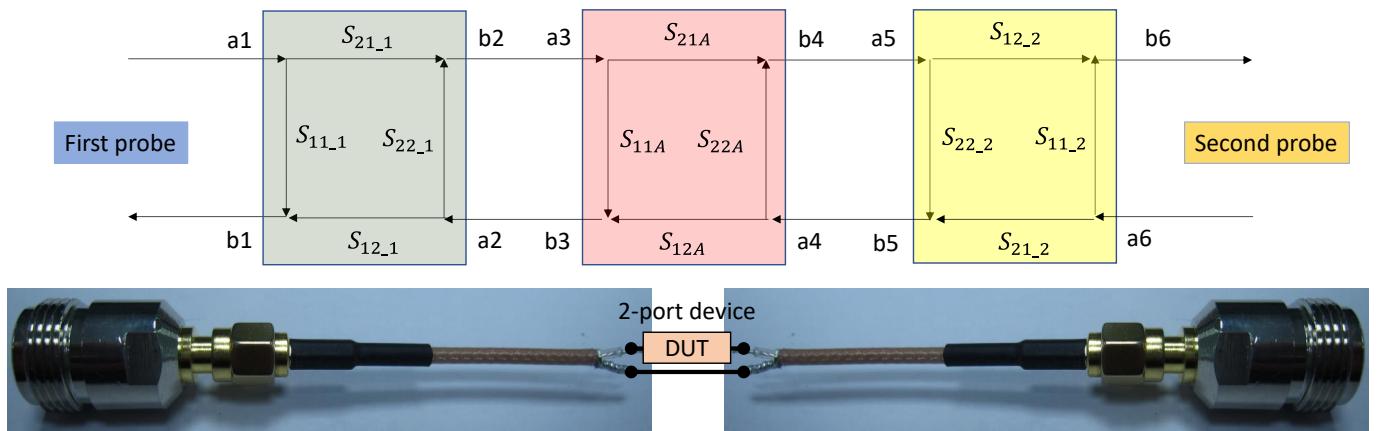


Fig. 9. Connecting a 2-port device between two probes, and their joint S-parameter model.

In the diagram in Fig. 10, we have summarized all the necessary procedures required for measurements with one or two probes.

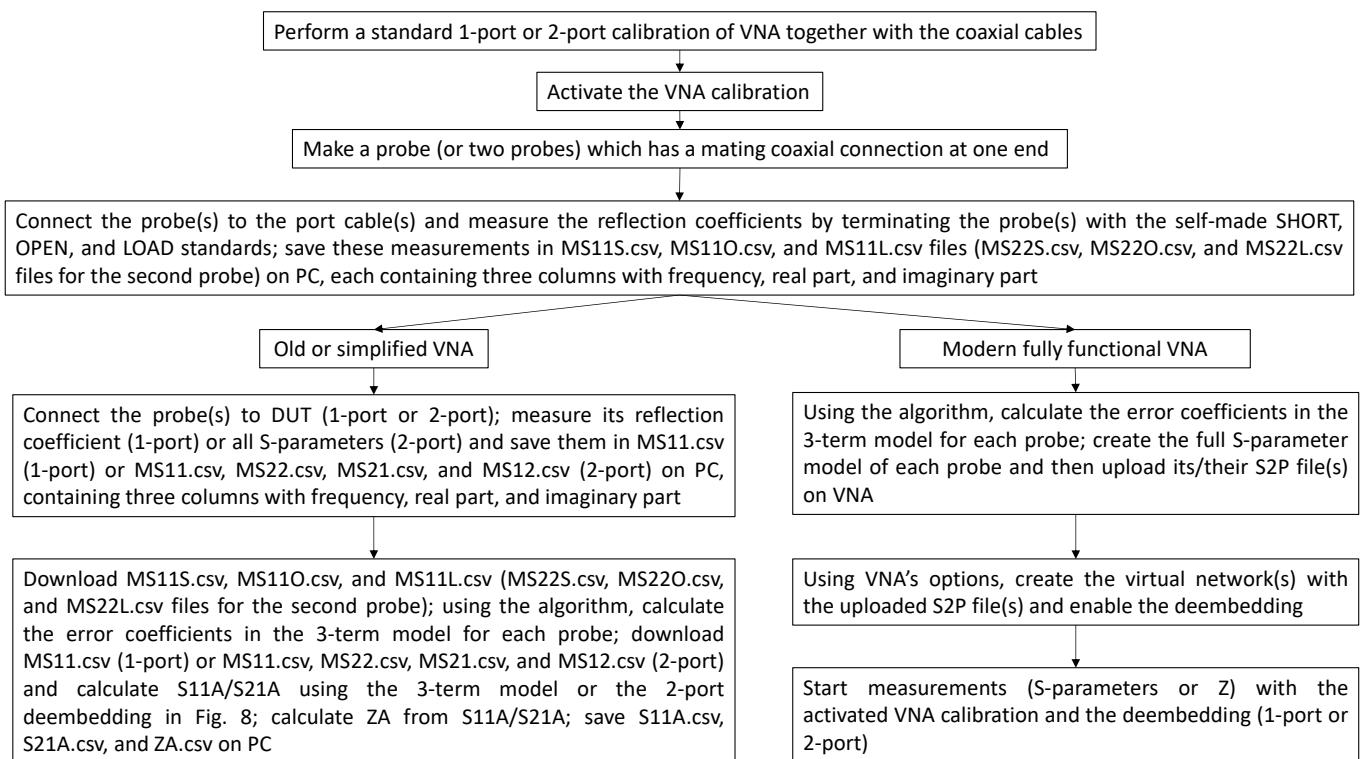
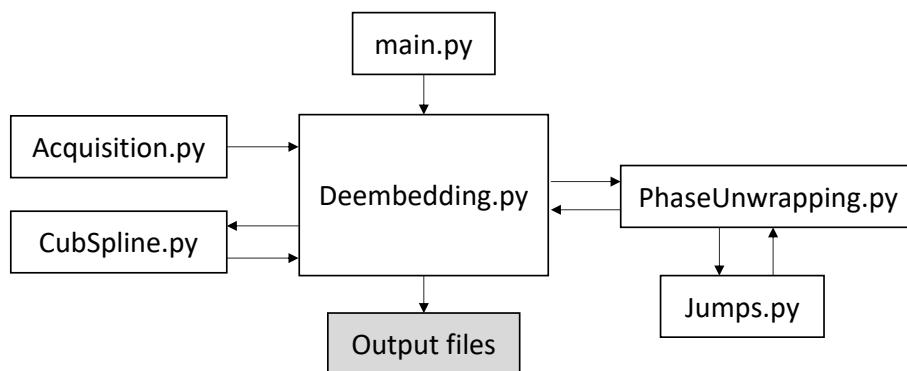


Fig. 10. Probe deembedding procedure for different types of VNA and measurement procedures.

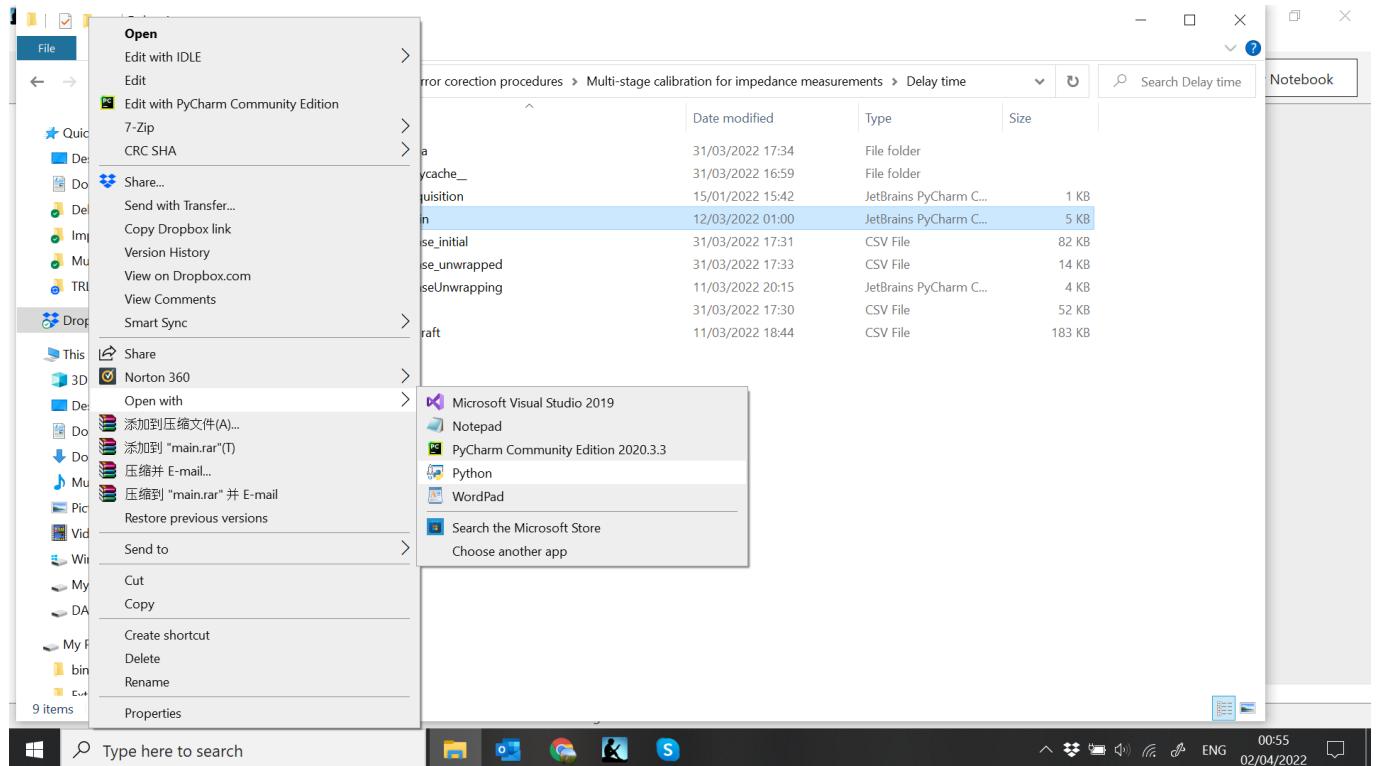
When run [the program](#), the head module **main.py** will show the format of the files to be measured and saved in a user folder. Read this description carefully. These experimental files will be used for calculations (deembedding and impedance calculation). Then, the module **main.py** calls **Deembedding.py**, which implements the 1-port and 2-port deembedding algorithms (see Figs. 5 and 8). The files are read by **Acquisition.py** which also forms complex arrays from the real columns in the files. In general, the frequency points at which S-parameters of the probes were measured do not need to coincide with the points at which the device is measured. Therefore, **Deembedding.py** requests **CubSpline.py** to recalculate S-parameters of the probes at the device's frequency points using cubic spline interpolation. The probes are deembedded by means of the 3-term model where a phase unwrapping for the propagation coefficients, the module **PhaseUnwrapping.py**, is used. An auxiliary module **Jumps.py** is used to calculate the number of phase jumps in complex S-parameters.



The execution of the program produces the files with the dispersion of S_A , Z_A , and also S2P files, which can be uploaded on a modern VNA for the 1-port or 2-port deembedding.

Appendix 2: Python IDEs and libraries

First of all, install Python on your computer.[5] A program can be run even without IDE by right-clicking **main.py** and selecting “Open with → Python”.



PyCharm IDE [6] is a very advanced tool for writing, debugging, and running Python codes. You can also use a simpler IDE, for example, **Thonny** (<https://thonny.org/>), which is preinstalled on all Raspberry Pi. In our algorithms we use the following Python libraries:

- numpy
- matplotlib.pyplot
- scipy
- scipy.interpolate

Installing Python packages: <https://packaging.python.org/en/latest/tutorials/installing-packages/>