QCM MATLAB Program Manual

The Shull Research Group
Version 2.0e "Eurynomos" and beyond

Chyi-Huey Joshua Yeh

Updated on 07/07/16

Contents

Part I. Quickstart Guide

1. Installation

1.1. Installing MyVNA program

The MyVNA program is a software that was devloped by the original harware designers of the N2PK Vector Network Analyzer (N2PK-VNA). This program directly manipulates the N2PK-VNA. For more information on the MyVNA program and the N2PK-VNA hardware setup, visit http://g8kbb.co.uk/html/myvna.html. Also, there is a small active community of radio-enthusiasts that constantly improve the software and harware design and capabilites of the N2PK-VNA. For more information on this community, visit https://groups.yahoo.com/neo/groups/N2PK-VNA/info.

1.1.1. Step-by-step myVNA installation

1. **Open up the folder name, "myVNA setup files".**
2. **Run myVNA.msi and follow the directions.** This installs the myVNA program that directly communicates and controls the N2PK Vector Network Analyzer (imepdance analyzer).
3. **After installing myVNA, double click on myVNA_setup.reg.** This adds registry values associated with the myVNA program.
4. **Go to the folder in which the myVNA program is installed.** The path usually looks something like this: C:\Program Files (x86)\G8KBB\myVNA.
5. **In this folder, make sure that there is a file name, "InitGenericUsb.dll".** If not, go to the "USB configure and Drivers" folder, located in the same path directory asthe myVNA program and copy the file, "InitGenericUsb.dll" into the myVNA path directory. This file is reponsible for communicating with the N2PK Vector Network Analyzer via a usb cord.
6. **In the myVNA program path directory, search for the file name, "RegServers.exe", right click on it, and click on "Run as administrator".** It is important that you run that execution file with administrator rights, otherwise, the execution file will not run properly. This file is responsible for registering the myVNA program with OLE capabilites. In other words, this file allows the MyVNA program to be controlled remotely with a 3rd-party program (in this case, AccessMyVNA).
7. **On the right hand panel in the MyVNA software, click on "VNA Hardware". Double click on "Choose CDS/Harmonic mode". A window will appear; make sure "Basic Mode" is selected.** Doing this will significantly increase the scan speed of the Network Analyzer. To the best of knowledge, differences between the "Basic Mode" and the "Harmonic Suppression Mode" is negligible (besides the difference in scan speed).
8. **When myVNA program has been successfully installed, connect the N2PK VNA to the computer.** The computer will automatically start scanning for the appropriate drivers to download and install, in order

to interface with the N2PK VNA. It is not necessary for the computer to go through this process (canceling this process will not affect the driver installation), since the associated drived will be installed manually. Go to Control Panel>View devices and printers. Scroll down and right click "Unknown device". From there, the driver can be manually installed by directing Windows to look into the folder in which the myVNA program was installed. Make sure "Check subfolder" option is checked.

9. **When running myVNA for the first time, a message will display, saying that the "Calibration" file was not found.** Click "OK"; the GUI will initialize. The first thing to do is to calibrate the N2PK VNA. Go to "Calibration Options" (left panel) and turn on the "Unguided Calibration" option. On the top toolbar, set the start and end frequency to 4 MHz and 56 MHz. Make sure the units are set to "MHz" and that the "Start/Stop" setting is selected. Also, change the number of setps to 400 and the average to 3. Click on the "Calibrate" button, located at the top toolbar.

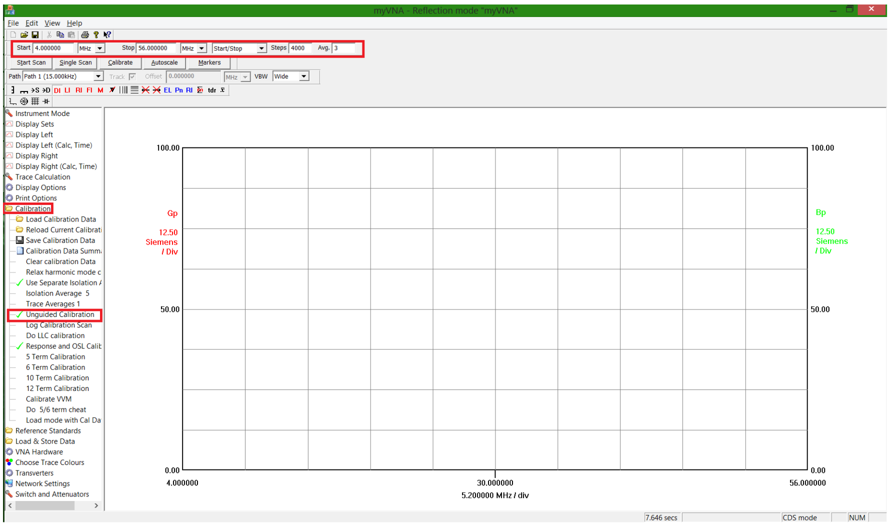


Figure 1: Set up te calibration process.

10. Make sure that the N2PK VNA is in an open-circuit state. Click on the "Open" button. The program will perform a scan from 4 MHz to 56 MHz with 4000 datapoints in between. Once this is finished, connect the "short" to the N2PK VNA, and click on the "Short" button. Repeat this step with the 50 Ω resistor and click on the "Load" button. Once the calibration is finished, click on the the "Finished" button and save the calibration file.

11. The myVNA program should fully functioning. To test it, simply click on the "Single Scan" button on the top toolbar. The "Autoscale" button might need to be pressed in order to adjust the plot to show the measurement. Once everything has been tested and properly set-up, exit out of myVNA.

Note: It is good practice re-calibrate before an experiment. This will reduce any unneccessay complication associated with the Lorentzian fitting process QCM measurement.

1.2. Installing AccessMyVNAv0.7 program

Setting up this program is easy. All that is required is to ensure that the folder, "AccessMyVNAv0.7", is in the same directory as the QCM MATLAB Program. For example, if the path directory to the QCM MATLAB Program is <C:\Program Files (x86)\QCM MATLAB Program>, make sure that <C:\Program Files (x86)\QCM MATLAB Program\AccessMyVNAv0.7> exists. This folder contains the program, AccessMyVNA, that is important for the QCM MATLAB Program to (indirectly) manipulate the MyVNA program. Currently, the QCM MATLAB Program automatically executes the AccessMyVNA each time the QCM MATLAB Program initiates and creates the GUI figure. However, to manually execute the AccessMyVNA program, the "AccessMyVNA.exe" file can be found in the "release" folder in the AccessMyVNAv0.7 folder.

Make sure to install the necessary .dll files. To do this, simply run execution file located in the "Visual C++ redistributable packages for visual basic 2013" folder. If the installation is being performed on a 32-bit computer, run the "vcredist_x86(1).exe" file. If the installation is being performed on a 64-bit computer, run the "vcredist_x64(1).exe" file. If an error message appears, (when executing the AccessMyVNA.exe file) stating that a .dll file is missing, try running both the 32-bit and 64-bit execution files.

For those running the "AccessMyVNAv0.7" on Windows 8 or higher, **make sure to change the compatability mode of the "AccessMyVNAv0.7.exe" to Windows 7**. Otherwise, the program will unexpectedly crash or malfunction.

1.2.1. Note for code developers

Within the AccessMyVNAv0.7 folder, there are many other folders and files. These "extra" folders and files are needed if there is a need to edit the AccessMyVNAv0.7 program code. To access this code, Visual Basic C++ (2013 or higher) must be installed. The file, "AccessMyVNA.sln", can be opened with Visual Basic C++ and the file, AccessMyVNADlg.cpp, can be accessed in the solution explorer panel. It is within the AccessMyVNADlg.cpp in which most of the important features and functions of the AccessMyVNAv0.7 is defined. Ideally, this program should be rewritten. However, rewriting this program requires advanced knowledge and experience in working with Visual Basic C++. In particular, this program relies on "MFC" (Microsoft Foundation Class) libraries, which are only supported in Visual Basic Professional 2013 or higher (not the "Express" version!). A free student license of Visual Basic Professional 2013 (or higher) can be obtained from Microsoft's Dreamspark program.

1.3. Installing the QCM MATLAB Program

The QCM MATLAB Program does not require any formal installation. However, make sure the program is performed by MATLAB 2014b or higher. Make sure that the following files:

1. "QCM_v002e_Eurynomos.m"
2. "QCM_v002e_Eurynomos.fig"
3. "fg_values.mat"
4. "raw_spectras.mat"

are in the <C:\Program Files (x86)\QCM MATLAB Program\> path directory. Also, make sure the following folders are located in the <C:\Program Files (x86)\QCM MATLAB Program\> path directory:

1. "QCM MATLAB manual"
2. "email files"
3. "AccessMyVNAv0.7"
4. "deleted data"

Note that the filenames and folder names are case sensitive.To run the QCM MATLAB Program, double click on the "QCM_v002b_bigfoot.m" file and run the script. As aforementioned, the AccessMyVNA program should also start with the creation of the MATLAB GUI figure. If it did not, an error should be reflected in the MATLAB command window.

2. How to take measurements (Quickstart guide)

2.1. Starting up and closing down the programs

NOTE: The GUI window associated with the "AccessMyVNA" will no longer appear for versions 2.0c and higher. Instead, this program runs in the background. A toggle button, , runs and closes the program using the Windows system cmd commands. The execution process, "AccessMyVNA.exe", can be located in the windows task manager. The native program, "MyVNA" will automatically run and be minimized on the desktop toolbar. Changes associated with the number of average scans should be adjusted in the "MyVNA" native program.

<THIS SECTION IS DEPRECATED FOR VERSION 002e AND HIGHER> To start up the QCM MATLAB program (and the other programs, a.k.a. MyVNA and AccessMyVNA), run the "QCM_v001a.m" script. An instance of the AccessMyVNA program should appear along with the creation of the MATLAB GUI figure. In the AccessMyVNA window, click on the button, "Start Scan" (see Figure 2). This will automatically start up the MyVNA program and begin (continuous) scans. Note that when the AccessMyVNA program is continuously scanning, the program cannnot be minimized and may seem unresponsive. As long the AccessMyVNA program is telling the myVNA program to continue scanning, the AccessMyVNA program is still running correctly.
<THIS SECTION IS DEPRECATED FOR VERSION 002e AND HIGHER>To have AccessMyVNA program to stop scanning, simply go to the radio dial labelled, "Maintain myVNA scan", and uncheck the radio dial. This will communicate to the AccessMyVNA program to stop scanning (this process may take a few seconds to complete). If the program does not stop scanning, then the AccessMyVNA program is probably malfunctioning. In that case, the AccessMyVNA program may have to be terminated in the windows task manager. **To begin the continous scanning again, make sure that the radiodial, "Maintain myVNA scan" is checked and click on the "Start Scan" button in the AccessMyVNA window.** Also, note that when the MATLAB GUI figure is closed, the QCM MATLAB code will communicate the AccessMyVNA to stop scanning. However, the **AccessMyVNA program and the MyVNA program needs to be manually closed**. If the AccessMyVNA program is not manually closed, reinitiating the QCM MATLAB GUI figure will create a new instance of AccessMyVNA; thus, there will be multiple instances of AccessMyVNA running at the same time, which can cause fatal errors.

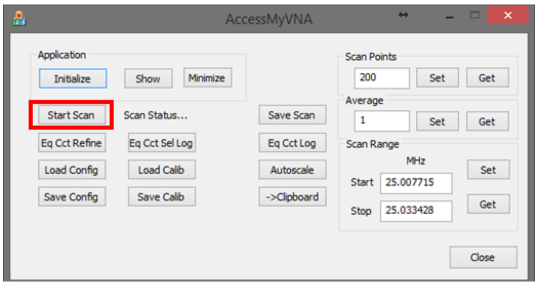


Figure 2: Click on the "Start Scan" button in the AccessMyVNA program to begin scanning.

2.2. Setting up parameters and measurement options

Parameters, settings, and options related to the collecting scans from the N2PK-VNA impedance analyzer can be controlled from the settings panel (see Figure 3).

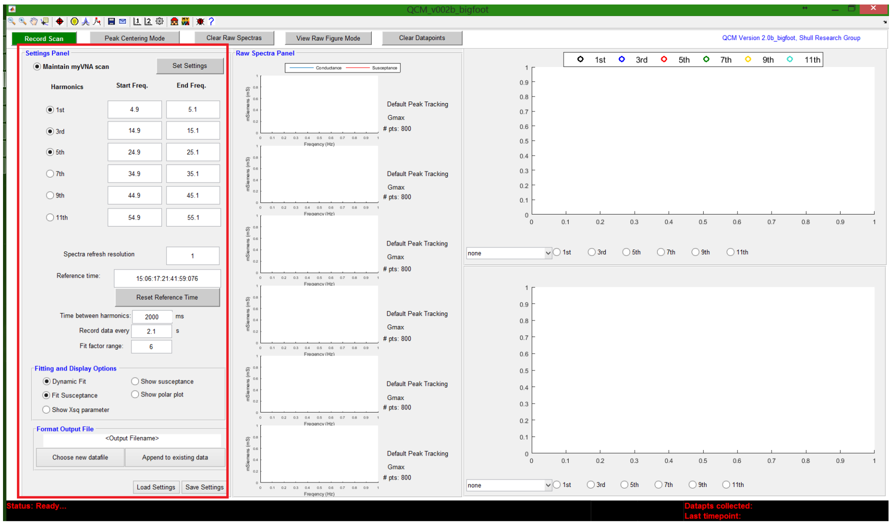


Figure 3: The settings panel contains information on the parameters, settings, and options related to the collection of the raw data.

2.2.1. Choosing which harmonic to track

The N2PK-VNA impedance analyzer is capable of tracking up to 6 harmonics at the same time (1st, 3rd, 5th, 7th, 9th, 11th harmonics). To choose which harmonics to track, check/uncheck the radio dials on the "Harmonics" column (see Figure). The start and end frequencies of the scan for each of the harmonic can be set manually by editing the numbers in the corresponding start/end frequencies. Also, the start and end frequencies can be set by initiating the "Peak Centering Mode". For more details in regards to the Peak Centering Mode, refer to Part 1, Section 3.1.

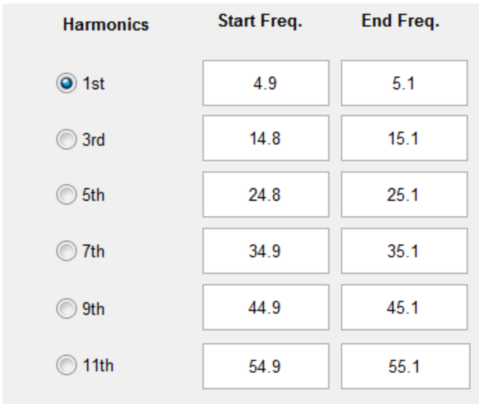


Figure 4: Multiple harmonics can be tracked. Choose which harmonics to track by checking/unchecking the radio dials under the "Harmonics" column. The start and end frequencies of the scan can be set for each harmonic.

2.2.2. Spectra refresh resolution

This option in the "Settings Panel" controls how often the QCM MATLAB Program refreshes the raw conductance and/or susceptance spectras. Its default value is 1, which means that the MATLAB program will refresh the spectras each time it reads in data from the datafile (see Part 2 for a more thorough explanation on how the QCM MATLAB Program reads data from the impedance analyzer). Changing this number to a higher integer-value can potentially increase the time resolution of the measurements significantly, since it takes a finite amount of time for MATLAB to constantly refresh the spectral plots. For example, a spectra refresh resolution of 10 means that the MATLAB prgram will refresh the plots every 10th measurement it collects.

2.2.3. Reference time

Since this programs collects frequency and dissipation shifts as a function of time, having a reference time is important, especially if new data needs to be appended onto another dataset from a previous experiment. The default reference time is set to the time in which the QCM MATLAB GUI figure was initiated. The reference time can also be changed manually by editting the values in the text box. Make sure that the values are in the correct format, yy:mm:dd:HH:MM:SS:FFF, where yy, mm, dd, MM, SS, and FFF, represent the last two digits of the year, the month, the day, the hour (24-hour format), the minutes, the seconds, and the milliseconds, respectively. **Do not hit the "Reset Reference Time" button after editing the reference time. Clicking on the "Reset Reference Time" will set the reference time to the current time! Simply hit the "Enter" key after inputting your desired reference time.**

2.2.4. Time between harmonics

This option controls the amount of time (in milliseconds) in between each harmonic measurements. In general, it is not neccesary to change this value. **A short time can lead to synchronization and systematic errors.** If the resonance peaks in the spectra looks odd or abnormal, increasing the amount of time in between each harmonic measurement might solve the problem. Otherwise, it might be due to other errors (see the Troubleshooting section). An example of what a resonance peak looks like if there is not sufficient amount of time in between harmonics is shown in Figure 5.

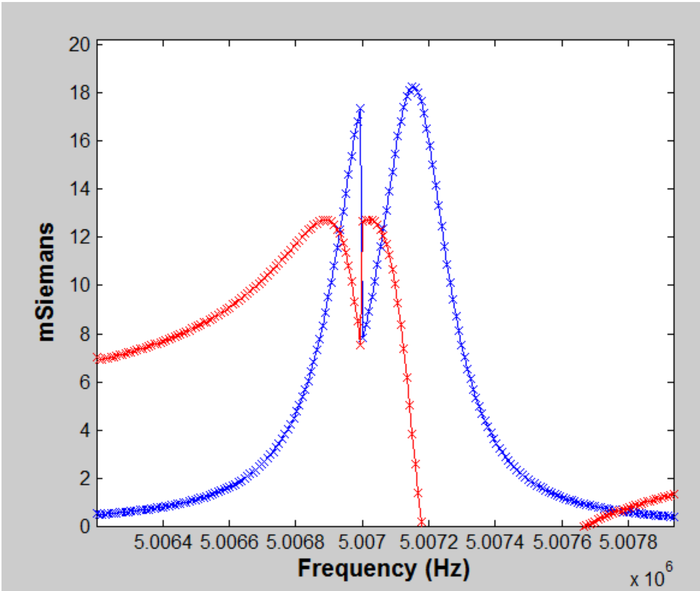


Figure 5: An abnormal resonance peak (5MHz) due to insufficient amount of time in between harmonics. Increase the time in between each harmonic measurement to solve this problem.

2.2.5. Number of datapoints

The number of datapoints in each frequency scan can be adjusted. The default value is 300 datapoints. Note that increasing the number of datapoints will require more time in between harmonics. Thus, the time in between haronics need to be adjusted accordingly. **A good rule of thumb is to allow 4-6 ms for each datapoint.**

2.2.6. "Record data every _ s" option

The time interval in which the QCM MATLAB Program records the spectra measure can be set in this option. For experiments that last for days to weeks, it is more practical to increase the time interval in which the MATLAB program records the data. Currently, the MATLAB program allows for 1 million timepoints to be recorded. If more than one million datapoints are collected, an error will appear and terminate the data collection process. The maximum number of datapoints to keep can be adjusted in the MATLAB code. However, it should be noted that the MATLAB program will progressively slow down as more data is being collected since it will take longer for MATLAB to append new values and variables into the output ".mat" files. For all practical purposes, this should not be an issue. If it is an issue, the output files should be split up into multiple files to mitigate this effect.
For experiments that are on a short time scale, it is important to remember that the number of datapoints and time in between harmonics need to be adjusted accordingly before decreasing the recording time interval. Otherwise, duplicates of the same spectra will be recorded. **In other words, if the recording time interval is less than the time in between harmonics, duplicates of the same spectra will be recorded.** An example of this systematic error is shown in Figure 6.

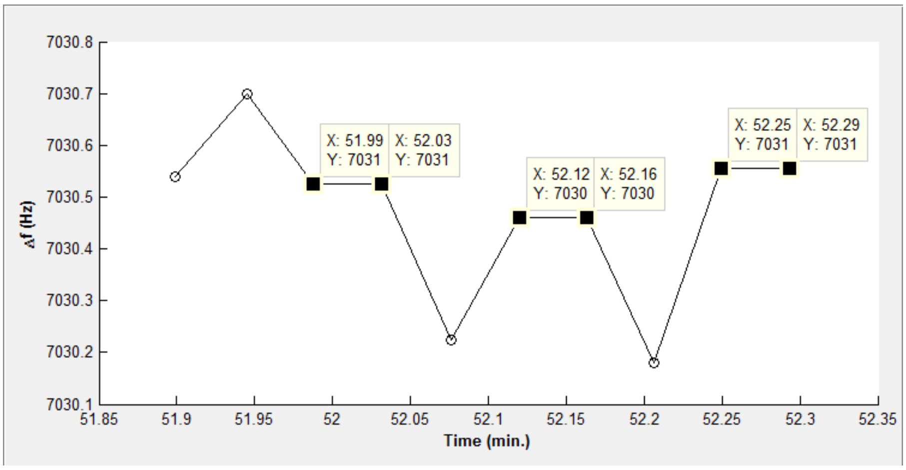


Figure 6: If the recording time interval is less than the time in between harmonics, duplicates of datapoints will occur. This is evident if the freqeuncy shift plot is observed during the measurement process.

2.2.7. Fitting options

There are five radio-dials in the "Fitting Options" panel (see Figure 7): "Dynamic Fit", "Fit Susceptance", "Show Xsq parameter", "Show susceptance", and "Show polar plot". If the "Dynamic Fit" radiodial is checked, a Lorentzian function curve fitting algorithm will be used to fit the resonance conductance peaks each time a spectra is collected. Unchecking this option significantly decreases the time required to record a measured spectra, which may be of importance if recording fast scans are required in an experiment. If the "Fit Susceptance" radiodial is checked along with the "Dynamic Fit" radio dial, both the conductance and susceptance resonance curves will be fitted. Note that the additional step of fitting the susceptance curve will require more time to record a measured spectra. If the "Show Xsq parameter" radiodial is checked along with the "Dynamic Fit" radiodial, the χ^2 parameter for each fit will be calculated and recorded. Details in regards to calculating goodness-of-fit for the curve fitting process is discuseed in later sections.

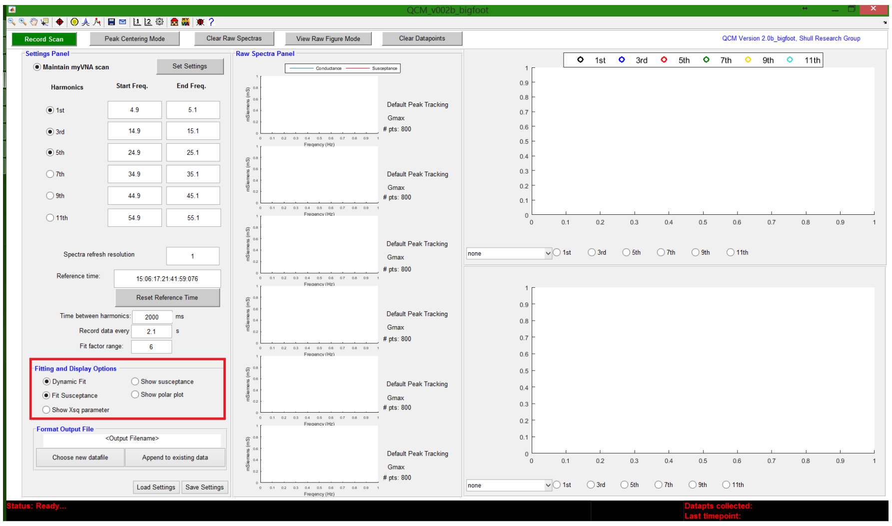


Figure 7: Options related to the fitting process during the measurements can be adjusted in the "Fitting Options" section located in the "Settings" panel.

2.2.8. Formatting output file

In the "Format Output File" panel, there is a button called, "Choose data filename". This button will open a file-output explorer dialogue box (see Figure 8). From this dialogue box, the name and location of the output data can be chosen. **If a pre-existing file is chosen, any absolute frequency/disspation and frequency/dissipation shift data that is collected will be overwritten unless the settings associated with the output datafile is also loaded.** If the frequency/dissipation data was not recorded (in other words, a Lorentzian function was not fitted to the raw spectra data), the saved raw spectra curves will be added into the .mat file containing the raw spectras. **However, be sure to set the correct reference time associated with the original output datafile. Otherwise, the timestamp**

associated with the raw spectras will be incorrect. Generally, choosing a pre-exisiting file is not recommended. Appending new data to a pre-exisiting file can be better accomplished by clicking on the "Append exisiting data" button. This allows the user to load pre-existing data sets. Data collection will added to the pre-exisiting datafiles. For more information on how the data is stored in the output data file, see the section on "Loading and saving settings".

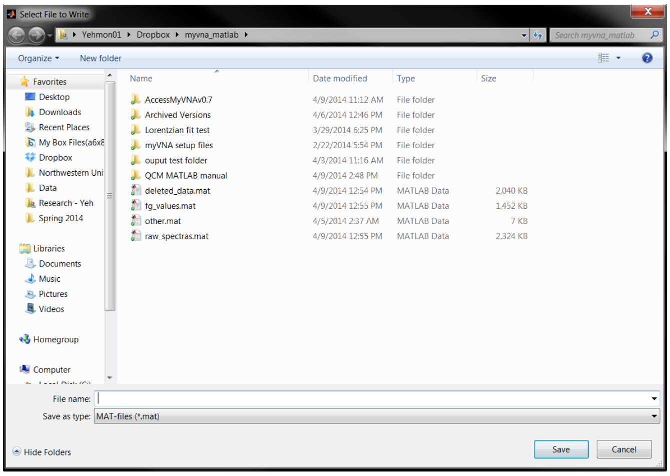
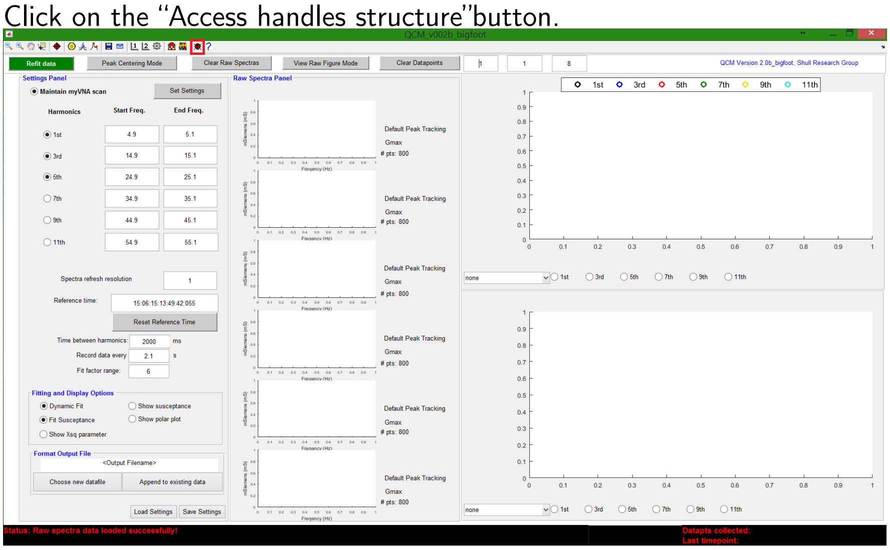


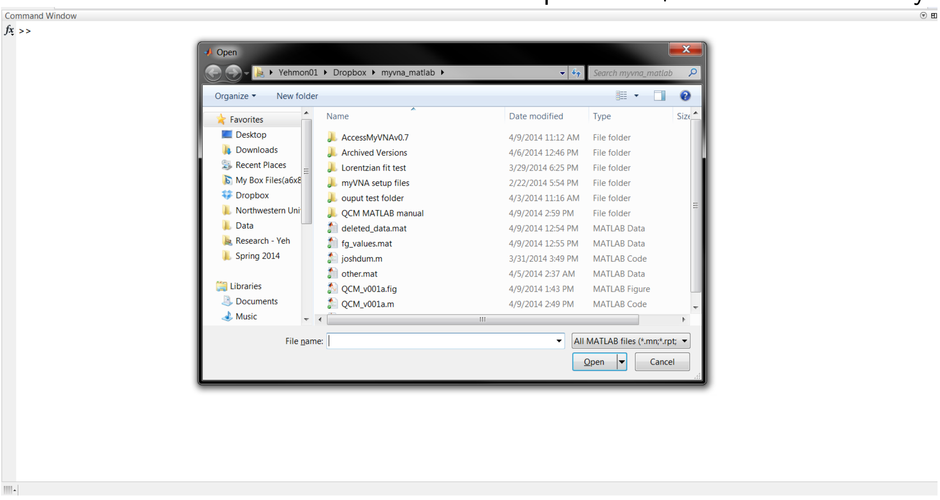
Figure 8: File-output explorer dialogue box.

If the "settings" file associated with the output datafile does not exist, the frequency/dissipation data can still be manually appended. To do this, follow the steps listed below:

1. Click on the "Access handles structure"button.



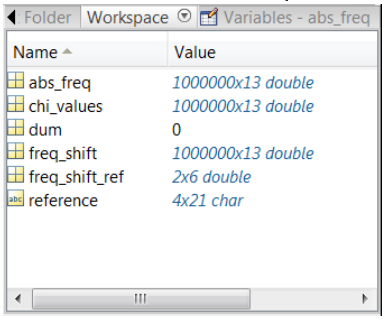
2. Go to the MATLAB Command Window and push Crtl+o buttons on the keyboard.



3. Open the output datafile that was chosen previously from thefile-output explorer dialogue. This will load the

variables stored in the .mat file into the MATLAB Workspace.

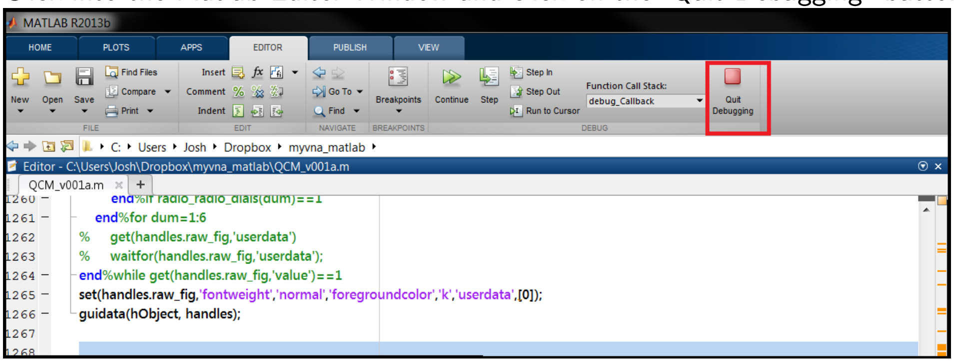
4. In the MATLAB Workspace, double-click on abs_freq or freq_shift



5. This will open up the "Variables" dialogue box containing the data stored in the variable that was double-clicked. Find the row number associated with the last datapoint that was stored.

6. Go back to the Command Window and type in the following command: "handles.din.n=<row number associated with the last datapoint that was stored>".
This will tell Matlab where (based on the row number) to begin appending the new frequency and dissipation data.

7. Click into the Matlab Editor Window and click on the "Quit Debugging" button near the top of the window.



2.2.9. Loading and saving settings

The options and setting parameters can be saved. By clicking the "Save Settings" button, a settings file will be saved in the same path location as the output datafiles. If no output datafile was selected to begin with, the the current setting options and parameters will be saved in the same path directory as the QCM MATLAB Program.
Clicking on the "Load Settings" button will automatically load the settings file associated with output datafile (which is programmed to be named as "<Output Filename>_.settings.mat"). If the automattically-loaded settings file was not the desired settings to be loaded, the "Load Settings" button can be clicked again to manally load the desired settings file. If the settings file cannot be found, an open file dialogue box will pop-up and the desired settings file can be loaded.
When the QCM MATLAB Program runs into any errors while loading or saving the setting options and parameters, the program will go into "Debugging mode". To exit out of the debugging mode, simply push the button that says, "Quit debugging", or type "return" in the MATLAB Command Window.

2.3. Record scan

To begin recording the measurements, simply click on the "Record scan" button 

. Details on how the measurment is recorded can be seen in the QCM MATLAB code. That secion of the code is thoroughly commented. Also, refer to later sections on the summary of how the MATLAB program records and saves the measurement scans.

2.4. Output files

2.4.1. No designated output filename and file location

When no output file is designated in the "Format Output File" panel in the "Settings" section, the recorded data will be stored in the default output files:

1. “fg_values.mat”: This MATLAB .mat file contains frequency and dissipation information that was collected during the recording process.

   Variables

   a) **“abs_freq”:** (1e6)x13 double array containing the absolute frequency and dissipation data. The first column contains the timepoints. Columns 2, 4, 6, 8, 10, and 12 contain the absolute frequency data. Columns 3, 5, 7, 9, 11, and 13 contain the absolute dissipation data.

   b) **“freq_shift”:** (1e6)x13 double array containing the frequency and dissipation shift data. The first column contains the timepoints. Columns 2, 4, 6, 8, 10, and 12 contain the frequency shift data. Columns 3, 5, 7, 9, 11, and 13 contain the dissipation shift data.

   c) **“chi_values”:** (1e6)x3 double array containing the χ^2curve fitting statistics. The first column contains the timepoints. Columns 2 and 3 contain the χ^2curve fitting statistics for the conductance and susceptance data.

   d) **“freq_shift_ref”:** contains the reference frequency (row 1) and dissipation (row 2) values for the frequency and dissipation shift data. Columns 1, 2, 3, 4, 5, and 6 contain reference values for harmonics 1, 3, 5, 7, 9. and 11.

   e) **“reference”:** 4x1 cell array containing (in order of row index) the reference timestamp (yy:mm:dd:HH:MM:SS:FFF), the time in between harmonic measurements in ms, the number of datapoints collected in the raw spectra data, and the time interval in which the data was recorded in s.

   f) **“version”:** string variable containing information on which version of the QCM program was used to record the data

   g) For versions 002e and higher, the estimated standard deviations of the Lorentz fit (a measure of the error in the Lorentz fits based on the results of the nlparci MATLAB function) are included. The first column contains the timepoints. Columns 2, 4, 6, 8, 10, and 12 contain the standard deviation of the resonance frequency. Columns 3, 5, 7, 9, 11, and 13 conatin the standard deviation of the dissipation data.

2. “raw_spectras.mat”: This MATLAB .mat file contains raw spectra information that was collected during the scan.

   Variables

   a) **“reference”:** 4x1 cell array containing (in order of row index) the reference timestamp (yy:mm:dd:HH:MM:SS:FFF), the time in between harmonic measurements in ms, the number of datapoints collected in the raw spectra data, and the time interval in which the data was recorded in s.

   b) **filename_t_<min>dot<fractional min>_iq_1_ih_<harmonic order>:** contains the frequency (first column), conductance (second column), and susceptance data (third column) at time <min>.<fractional min>.[1]

   c) For versions 002e and higher, the way in which the raw spectras are saved has been updated. Raw spectras are saved in a 4-column cell format: “raw_spectra_<harmonic mode>”. The first, second, third, and fourth column contain information of the timepoints, base filename, the harmonic order, and a 7-column raw spectra matrix, respectively. The first, second, and third columns, of the raw spectra matrix represent the frequency, raw conductance, and raw susceptance.

Both files are located in the same path directory as the QCM MATLAB Program. It is important to note that these two .mat files may or may not be cleared before the data collection process. Thus, any data stores may or may not be convoluted from previous stored data in the .mat files. These two files is just a safeguard from losing data that was collected. If data is extracted from these two files, it is important to clear the data that was stored, so that future data stored in the files will not be convoluted with old data. To do this, follow these steps:

1. Go to the MATLAB Command Window and type, “clear all”. Hit the “Enter” key.
2. In the MATLAB Command Window, type “dum=0”. Hit the “Enter” key.
3. Click into the MATLAB Workspace and push “Crtl+o” buttons on the keyboard.

[1] This format is deprecated for v002e and higher. However, this format can still be used by selecting the radio dial for “Save/read spectra using legacy format” in the “Set Preferences” figure window. See Section 3.1.10 for details.

4. This will create a save prompt window. Save and replace both “fg_values.mat” and “raw_spectras.mat”.

If the settings are saved with no designated output file , a “default_settings.mat” file will be saved in the same path directory as the QCM MATLAB Program.

2.4.2. Designated output filename and file location

When an output filename and location are designated, the recorded data will be stored in the following output files:

1. “<user-designated filename>.mat”: This MATLAB .mat file contains frequency and dissipation information that was collected during the recording process.

   Variables
   a) “abs_freq”: (1e6)x13 double array containing the absolute frequency and dissipation data. The first column contains the timepoints. Columns 2, 4, 6, 8, 10, and 12 contain the absolute frequency data. Columns 3, 5, 7, 9, 11, and 13 contain the absolute dissipation data.
      i. “freq_shift”: (1e6)x13 double array containing the frequency and dissipation shift data. The first column contains the timepoints. Columns 2, 4, 6, 8, 10, and 12 contain the frequency shift data. Columns 3, 5, 7, 9, 11, and 13 contain the dissipation shift data.
      ii. “chi_values”: (1e6)x3 double array containing the χ^2curve fitting statistics. The first column contains the timepoints. Columns 2 and 3 contain the χ^2curve fitting statistics for the conductance and susceptance data.
      iii. “freq_shift_ref”: contains the reference frequency (row 1) and dissipation (row 2) values for the frequency and dissipation shift data. Columns 1, 2, 3, 4, 5, and 6 contain reference values for harmonics 1, 3, 5, 7, 9. and 11.
      iv. “reference”: 4x1 cell array containing (in order of row index) the reference timestamp (yy:mm:dd:HH:MM:SS:FFF), the time in between harmonic measurements in ms, the number of datapoints collected in the raw spectra data, and the time interval in which the data was recorded in s.

2. “<user_designated filename>_raw_spectras.mat”: This MATLAB .mat file contains raw spectra information that was collected during the scan.

   Variables
   a) “reference”: 4x1 cell array containing (in order of row index) the reference timestamp (yy:mm:dd:HH:MM:SS:FFF), the time in between harmonic measurements in ms, the number of datapoints collected in the raw spectra data, and the time interval in which the data was recorded in s.
   b) filename_t_<min>dot<fractional min>_iq_1_ih_<harmonic order>: contains the frequency (first column), conductance (second column), and susceptance data (third column) at time <min>.<fractional min>.

3. If a settings file was saved, “<user-designated filename>_settings.mat”

2.5. Troubleshooting

The QCM MATLAB Program code is thoroughly commented and tested. If errors occur, the code can be examined and debugged. The easiest way to access the QCM MATLAB GUI handles structure is by pushing the  button.

Part II.
Comprehensive guide to the QCM MATLAB Program

3. Features of the QCM MATLAB Program

This section provides a more comprehensive overview of all the features available in the MATLAB program. The basic sections of the GUI are shown in .

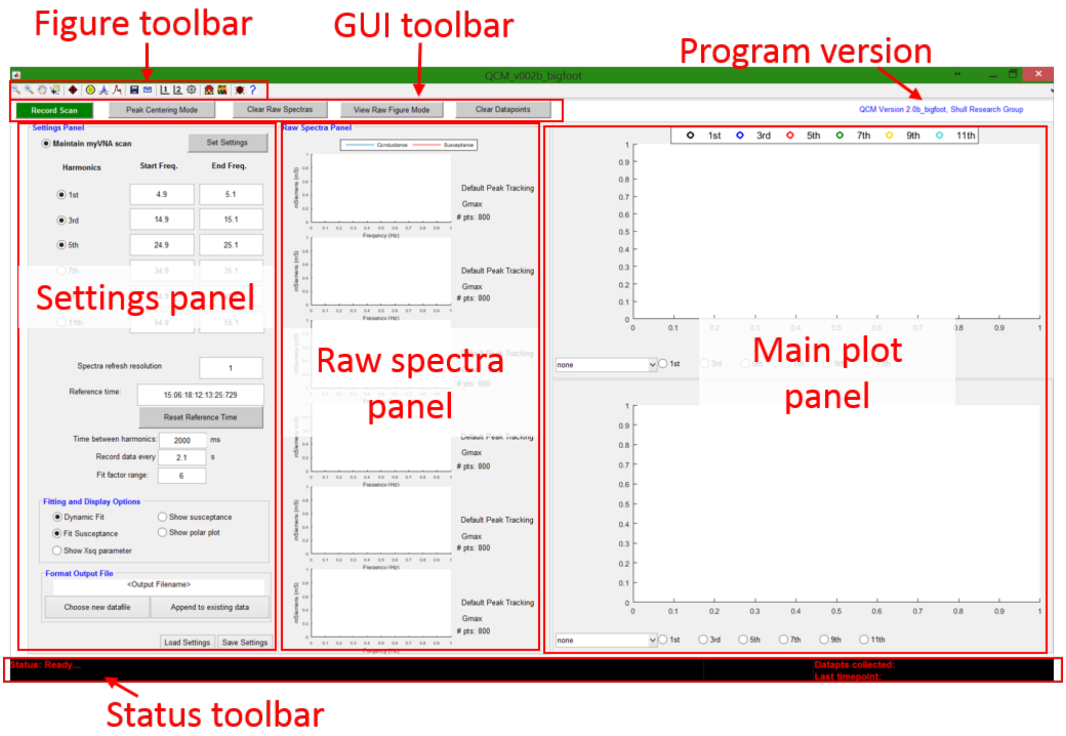


Figure 9: GUI Overview

3.1. Figure toolbar

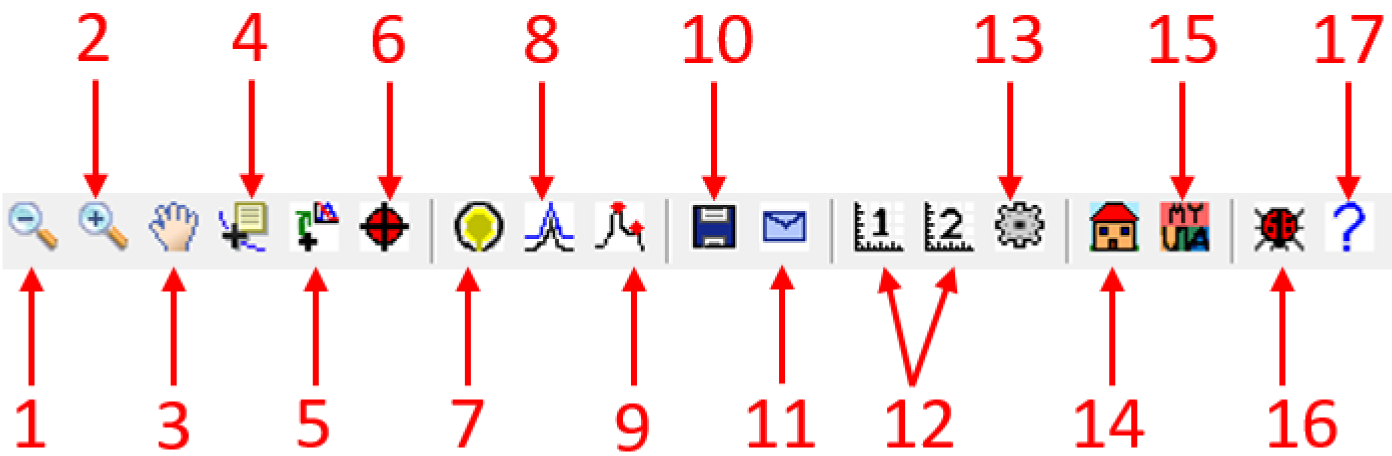


Figure 10: A list of all of the button on the figure toolbar

#	name	description
1	zoom-in	zoom-in into an axes plot
2	zoom-out	zoom-out from an axes plot
3	pan tool	pan axes tool
4	data cursor	select datapoints in an axes
5	select raw spectra (new in v002e)	view raw spectra corresponding to selected datapoints
6	delete datapoints mode	enable "delete datapoints mode"
7	load bare crystal data	load reference frequencies
8	refit pre-existing raw spectras	this will enable the "refit mode"
9	multi-peak fitting options	set multi-peak functions
10	save frequency shift data	save the frequency shift data
11	email notifications	set-up email notifications
12	plot options	plot options for primaryaxes1 and primaryaxes2
13	set preferences	set-up GUI preferences
14	home	revert GUI state back to the default state
15	MyVNA	execute AccessMyVNA.exe
16	debugging mode	enter into debugging mode (for advanced users only!)
17	help	opens the pdf manual

Table 1: Short description of each button on the figure toolbar

3.1.1. Zoom in/out, pan, and data cursor tool 

These tools have the same functionality as the default zoom in/out, pan, and data cursor tools on a standard figure window.

3.1.2. Select raw spectra (new in v002e Eurynomos) 

Clicking on this button will activate the data cursor mode with a customized callback function (both the "data cursor tool" and the "select raw spectra" button will be enabled: ). When a datapoint is selected in the primaryaxes1 and primaryaxes2, the program will attempt to extract and display the associated raw spectra. The user will be able to refit the spectra based on the user-selected guess values by pushing the "Fit" push button (see Figure 11). The fitting process is the same as the fitting process introduced in the "peak centering mode" (see for more details). The user is also given the option to choose the range in which the fit will be performed by selecting/deslecting the "Only fit within axes span" radio button. When the fit is complete, information comparing the old and new Lorentz fit will be shown as an annotated textbox (see Figure 12). The user can choose to accept the new fit by pressing the "Accept" push button; this will update the selected datapoint to reflect the newly accepted Lorentz fit. However, the removal of the datapoint will not be reflected in the saved <base_filename>.mat file, where the fitted f and Γ values are stored. To permanently save the changes to the <base_filename>.mat file, the user must push the save button,  (see 3.1.7 for details).
Alternatively, if the raw spectra is not useable, the user can choose to delete the datapoint by pushing the "Remove" push button. This will remove the datapoint from the data arrays stored in the handles structure of the GUI program; the removal of the datapoint will not be reflected in the saved <base_filename>.mat file, where the fitted f and Γ values are stored. To permanently save the changes to the <base_filename>.mat file, the user must push the save button,  (see 3.1.7 for details).

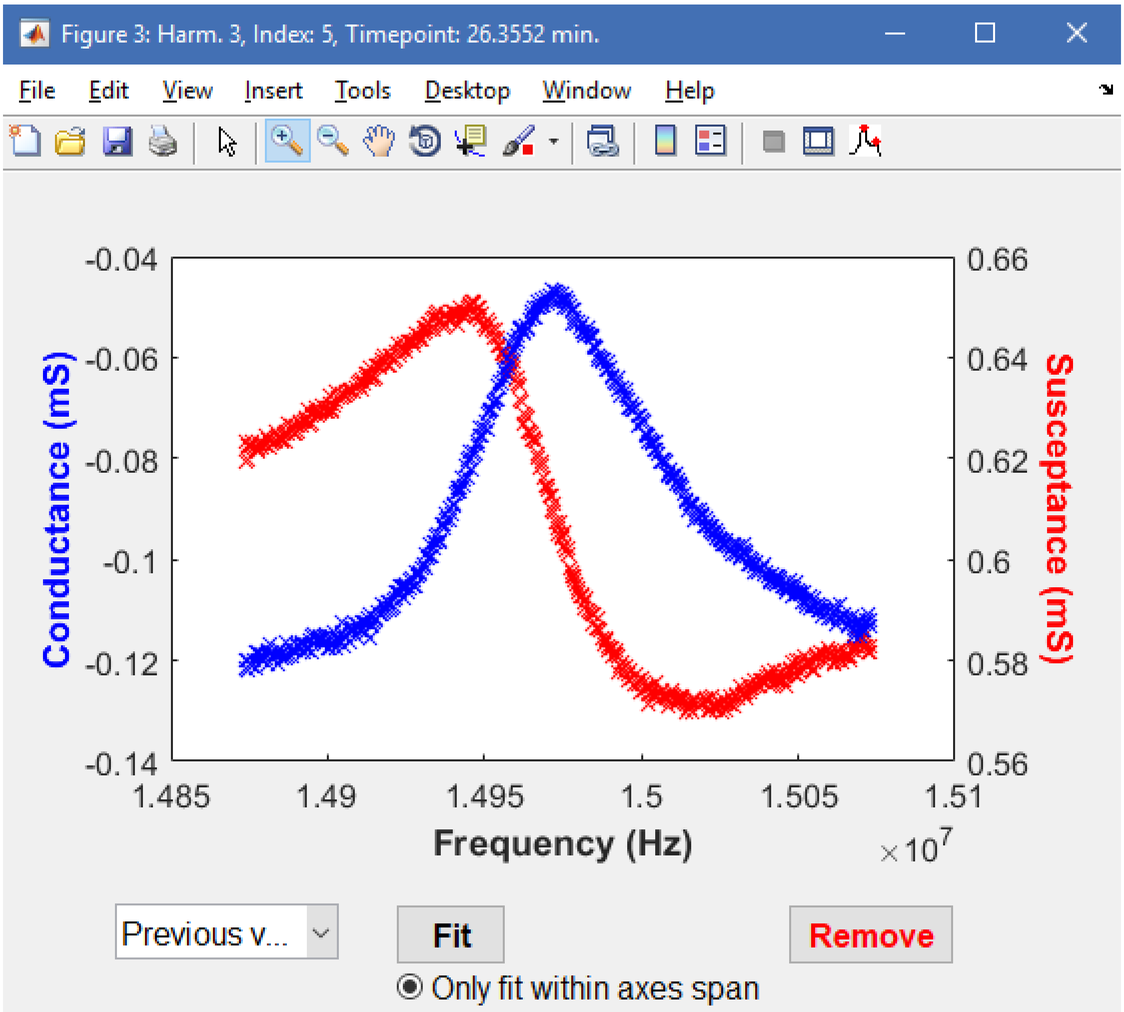


Figure 11: The refitting window that allows the user to refit the selected raw spectra.

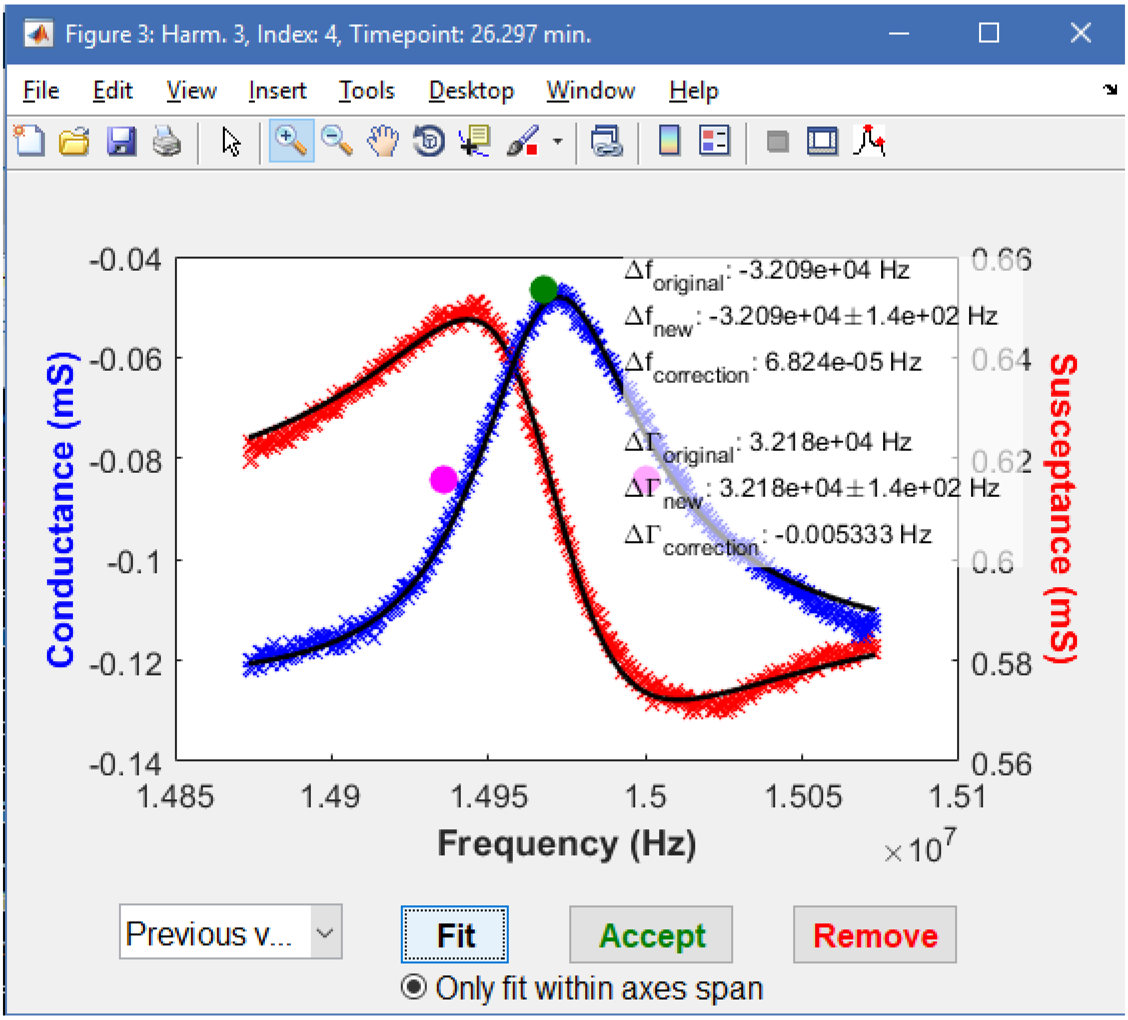


Figure 12: An example of what the refitting window looks like after a new Lorentz fit of the selected datapoint.

3.1.3. Delete datapoints mode 

Clicking on this toggle button will put the user in the "Delete datapoints" mode. You will have the ability to select a datapoint graphically from the main plots showing the frequency and dissipation shifts. The selected datapoints can then be deleted by puching the "delete" button on the keyboard. This will temporarily remove the datapoint from the plot. Additional datapoints can be deleted from the plot. Refreshing the plot by clicking on one of the hamonic radial dials in the main plot panel (the radial dials that allow you to choose which harmonic mode to plot) will undo all deletion of datapoints.

When the "Delete datapoints" button is toggled on the "on" state, an additional button, , will appear. Clicking on this "trash can" icon will permanently remove any datapoints that were deleted from the main plot panel. However, the raw spectra data associated with the datapoints will not be deleted. If a datapoint was accidentally deleted, that datapoint can be recovered by refitting the raw spectra, which is stored in the "<filename>_raw_spectra.mat" file. This functionality is particularly useful when an outlier (due to poor fitting, etc.) causes the frequency and/or dissipation plots to zoom out, making graphical interpretation of the measurement proccess difficult. This function is NOT to be used for data manipulation!

3.1.4. Load bare crystal data 

Since frequency and dissipation shifts are relative to a reference value, the reference values, which are typically values measured using a bare crystal, can be loaded. The default reference values correspond to theoretical resonance frequencies (5, 15, 25, etc. MHz) and an arbitrary dissipation value of 100 Hz. Clicking on this button will bring up a windows file explorer dialogue, asking to load the reference crystal data file, "<reference filename>.mat". This data file has the same format and data structure as a typical QCM file (see Section 2.4.2 for details). The program reads in the reference crystal data file and computes the average of the entire frequency and dissipation dataset for each harmonic. The average values are then used to replace the default reference values. The frequency and dissipation shifts measured will be adjusted accordingly. The plots in the main plot panel might need to be refreshed in order to see the changes made to the reference values.

Note that clicking on the "Home" button (, see Section 3.1.11) will reset the reference shift values back to the default values.

3.1.5. Refit pre-existing raw spectras 

This button will place the user in the "Refitting" mode. This button allows the user to load raw spectra data files and refit the data. A window file explorer dialogue will appear asking for a raw spectra file, "<file-name>_raw_spectra.mat". The program effectively treats the raw spectras as measurements obtained from the MyVNA program. Thus, all fitting options related to multi-peak fitting and different guess values will be avaialble. This is useful when frequency and dissipation shifts are lost and need to be recovered. Also, for situations where the measurement process used a sub-optimal single-peak fitting algorithm to characterizing overlapping resonance peaks, this functionality allows the user to post-fit the collected data in order to obtain a more accurate and precise values for the frequency and dissipation values.

3.1.6. Multi-peak fitting options 

This button brings up a table containing parameters related to the options for identifying peaks in the resonance spectra. The first column contains the harmonic order the parameters are associated. The second column contains the peak prominence sensitivity factor associated with the peak identification prcoess (a lower number corresponds to higher peak sensitivity); the values must be between 0 and 1. The third column contains parameters associated with the minimum peak height requirement relative to the datapoint with the highest y-value in the spectra (a lower number corresponds to higher peak sensitivity); the values must be between 0 and 1. The fourth column contains the maximum number of peaks the program will search in the resonance spectra; values must be between 1 and 3 (the maximum number of peaks is set to 3).

3.1.7. Save frequency shift data 

This button saves the current state of the frequency and dissipation shift data stored in the handles structure of the GUI. If no filename has been designated, the data will be saved according to the default filenames (see Section 2.4.1 for details).

3.1.8. Email notifications 

Clicking on this button will bring up a window asking the user to input an email address and an option to turn on/off the email notifications.

For versions 002c and below:

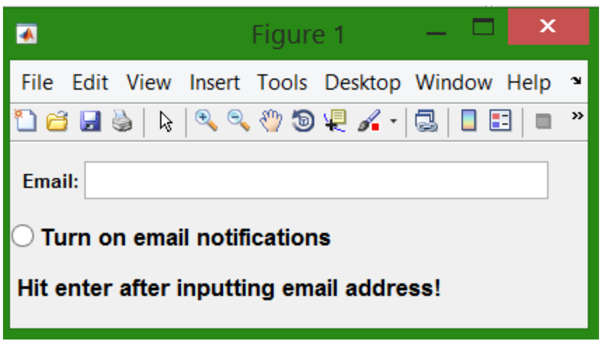


Figure 13: Email notification window for versions 002c and below.

This function only stores the email information and the toggle state of the email notifications in the handle structure. When the notifications are turned on, an email will be sent (using the "email_send" function) during the measurement process. The "email_send" function utilizes undocumented Javascript code. A screenshot of the GUI (.jpg) will be sent along with a Command Window log text file, QCM_diary.txt (see "Diary" function in the MATLAB help

document). Note that the QCM_diary.txt file will continue to grow larger as more points are being recorded (since there are more "stuff" in the command window). Thus, it is advisable to clear out the QCM_diary.txt file ever once in awhile during a long measurement process with short measurement intervals. Also, if the "email_send" functions runs into an error, an error message will appear in the command window, but the measurement will continue (the main code is written in a try/catch-block).
*Note: The email notification window was revamped in version 002d_Drakon. The outgoing email server is no longer hard-coded into the MATLAB script. This change was made due to the public release of this program onto GitHub. The outgoing server, host email, and host email password must be designated in order for this option to work properly. Also, a "test connection" button was added to ensure succesful email notification.

For versions 002d and above:
For versions 002d and above, the email host information is no longer hard-coded into the MATLAB program. Versions 002d have been released publicly; thus, for security reasons, email usernames and passwords were explicitely removed. The user now has to input the email and associated password that he/she wishes to use to send out email notifications. The email outgoing host server also needs to be provided.
A "Test connection" button has been added to test that the email communication between the host and recipient email had been succesfully established; a "test" email is sent between the host and recipient servers.

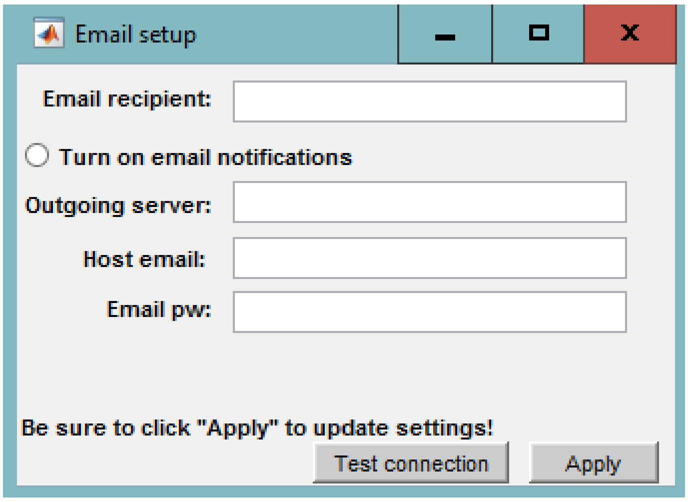


Figure 14: Email notification window for versions 002d and above.

3.1.9. Plot options 

This button allows the user to adjust the properties for "primaryaxes1" (top right axes) and for "primaryaxes2" (bottom right axes). The callback function uses the "inspect" function to provide the user access to the axes properties. For more information on what each axes property does, refer to the MATLAB help document on the "inspect" function.

3.1.10. Set preferences 

Clicking on this button will bring up a window with a list of preferences and options associated with the data collection process.

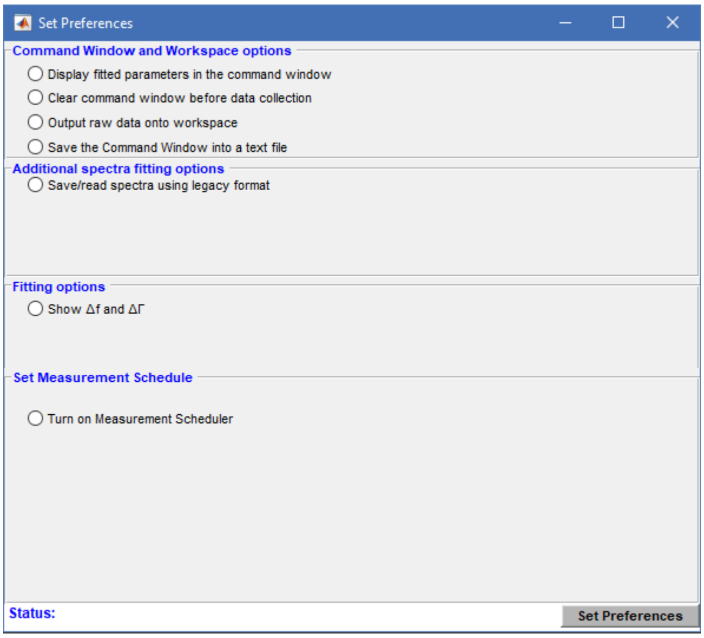


Figure 15: User preference window.

For versions 002c and above, the preference window is divided into four panels:

1. Command Window and Workspace options
2. Additional spectra fitting options
3. Fitting options
4. Set Measurement Schedule

Details of each option in the preferences can be found in associated tooltip string (hover the mouse pointer over the option).
The measurement scheduler is a useful tool that is designed for measurements that require minimal monitoring. A start and end time can be designated. Also, the measurement recording increment time can be changed; this allows for the user to change the measurement frequency that is appropriate for the time scale in which the experiment takes place. Be sure that the "Set Preferences" button is pushed before closing the Preferences window. Otherwise, the settings will not be saved!
Note: Changing preferenecs must be performed when the measurement is NOT in progress. Otherwise, the changes in the preferences will not be set properly.

3.1.11. Home 

This button resets the entire GUI back into its default state. All data that was collected will be deleted from the handle structure associated with the figure window.

3.1.12. MyVNA 

NOTE: For versions 2.0c and higher, this button is a toggle button. When the button is in the "on" state, both AccessMyVNA and MyVNA programs will execute. However, only the MyVNA GUI will appear (minimized in the desktop toolbar). The AccessMyVNA GUI will no longer appear. Instead, the execution process (AccessMyVNA.exe) can be located in the Windows Task Manager. When the button is in the "off" state, both the AccessMyVNA and MyVNA program will be terminated (forced termination using the cmd.exe commands).

3.1.13. Debugging mode 

This takes the user into debugging mode. The callback function uses the "keyboard" function and also exports the handle structure of the GUI into the "base" workspace. Note that any changes that are made will not necessarily be saved. For example, changes to variables (not handle objects) will not be saved in the handles structure unless the "guidata" function has been initiated in the command window. However, direct changes to handle objects (using the "set" function), will be immediately applied and saved. For more information on the details on how information is saved, see the "guidata" in the MATLAB general help files.

3.1.14. Help 

This button will direct the user to the pdf manual. Also, feel free to contact me through email (CHJoshuaYeh@u.northwestern.edu).

3.2. GUI toolbar

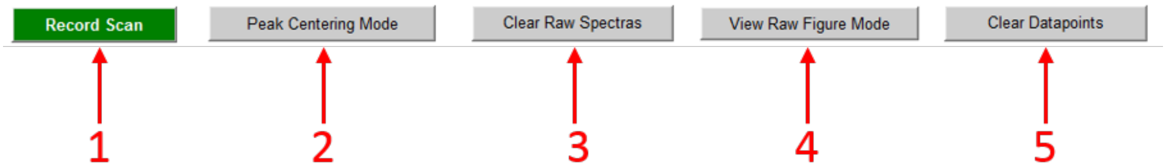


Figure 16: GUI toolbar

#	name	description
1	Record Scan	Start the measurement process
2	Peak Centering Mode	Allows for finding the appropriate harmonics
3	Clear Raw Spectras	Clear all of the spectral plots
4	View Raw Figure Mode	View each spectra in individual plots
5	Clear Datapoints	Delete all frequency shift datapoints

Table 2: Short description of each button on the GUI toolbar

3.2.1. Record Scan

Pushing this button starts the data collection process. Note that when the data acquisition process is ongoing, some of the features are disabled; this is to prevent unwanted errors that can lead to loss of data.

3.2.2. Peak Centering Mode

Pushing this button sets the GUI into a "peak centering" mode, which allows the user to set the span for the program to scan for each odd harmonic resonance. When this mode is active, a "Center Peak" panel will display within the "Settings Panel" (see Figure 17). Selecting the odd harmonic radio dial will allow the user to view the current live reading of the raw spectra, defined by the start and end frequency. A new figure window will appear, displaying the conductance and susceptance across the set frequency range (see Figure 18).

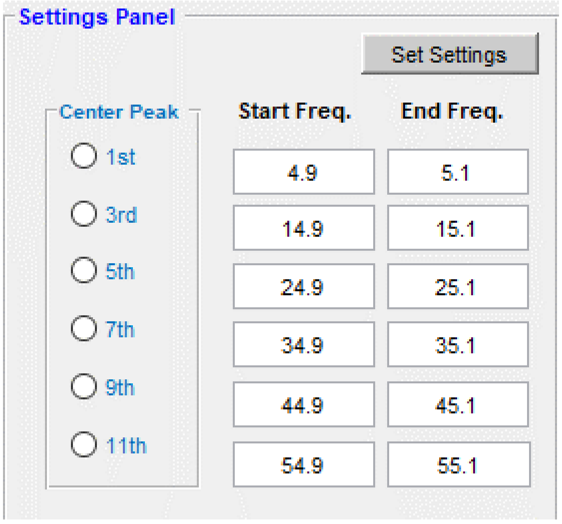


Figure 17: The "Center Peak" panel appears within the "Settings Panel" when "Peak Centering Mode" is active.

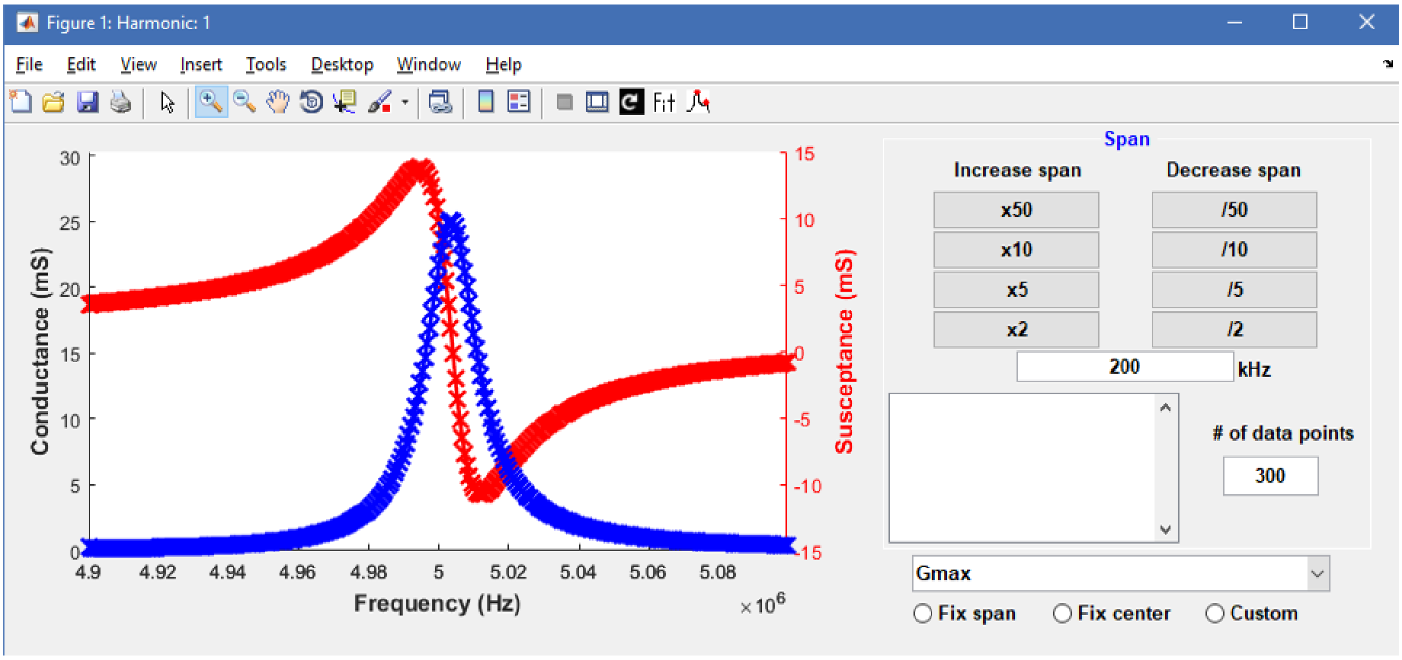


Figure 18: A figure window appears showing the electrical conductance and susceptance scan for the selected harmonic resonance.

On the right side of the figure window, as shown in Figure 18, the "Span" panel allows the user to adjust the start and end frequency of the electrical admittance scan. This can be done either by pushing the increase/decrease span push buttons or manually inputting the width of the span in kHz (where the center of the span remains unchanged). Alternatively, the user can use the zoom in/out, data cursor[2], and pan tools to change the frequency scan range.

Due to the nature of how the MATLAB GUI communicates with myVNA, a refresh toolbar button ( ) was added in order to manually reload the electrical admittance frequency scan. If the spectra looks suspiciously incorrect, it is recommended to refresh the spectra manually. In addition to changing the start and end frequencies of the scan, the number of datapoints measured in the spectra can also be changed. The number of datapoints should at least be 100 (default is set to 300); otherwise, the Lorentz fitting will likely be inaccurate.

Methods for determining guess parameters While in this mode, the different methods for determining an initial estimate of the Lorentz fitting parameters can be set. Currently there are 5 options: "Gmax", "Derivative", "Bmax", "Previous values", and "User-defined"

[2]The data cursor allows the user to center at the corresponding frequecy of the selected datapoint. Once the datapoint is selected, click off of the datatip (but within the plot axes) to refresh the spectra.

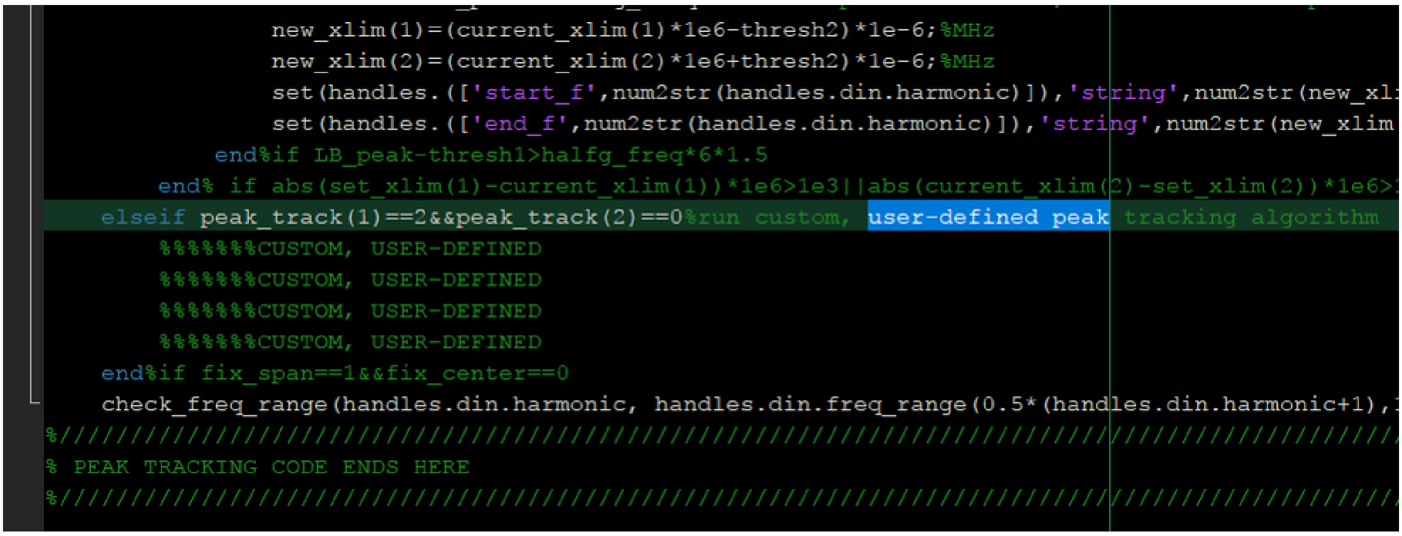
1. "Gmax": The guess parameters are based on examining the electrical conductance spectra, where the maximum of the conductance response represents an estimate of the resonance frequency.
2. "Derivative": The guess parameters are based on the derivative of the complex electrical admittance magnitude, where the maximum of the derivative is used as an estimate for the resonance frequnecy.
3. "Bmax": The guess parameters are determined from the electrical suscpetance spectra, where the maximum of the susceptance reponse represents an estimate of the resonance frequency. This method is analogous to the "Gmax" guess method. This is algorithm is useful when the VNA has a less-than perfect calibration, where the resonance response undergoes a rotation in the complex admittance polar plot.
4. "Previous values": Guess parameters will be based on previously fitted parameters. This is useful for most cases when the change in the harmonic resonances occurs slowly.
5. "User-defined": This prompts the user to manually set the guess parameters. Generally, this algorithm is reserved for fitting complex resonance responses, where multiple anharmonics are coupled with the primary harmonic mode.

Once guess parameters are set, the spectra is ready to be fitted with a Lorentz curve. To do this, push the "Fit" toolbar button (Fit). The MATLAB program will attempt to fit the spectra using the defined guess parameter method. If the Lorentz fit fails, the MATLAB program will attempt to fit the spectra again using another guess parameter method.

Multiple anharmonic modes To mitigate the effects of anahrmonic modes that can couple with the primary harmonic resonance, multiple Lorentz curves can be used to fit the primary and anharmonic resonances. To adjust how the MATLAB program detects the relevant resonance peaks, the "Peak finding options" toolbar button can be pushed (). See Section 3.1.6 for more details.

Peak tracking algorithm During the measurement process, the MATLAB GUI program relies on a peak tracking algorithm to measure the resonance responses. There are 3 radio dials that can be selected: "Fix span", "Fix center", "Custom".

1. Only "Fix span" is selected: The program will adjust the start and end frequencies by dynamically changing the span center of the scan. The range of the scan remains static.
2. Only "Fix center" is selected: The program will adjust the start and end frequencies by dynamically chaning the scan span. The span center remains static.
3. Both "Fix span" and "Fix center" are selected: The program will not adjust the start and end frequencies. Essentially, the program will not try to track the resonance peak.
4. Only "Custom" is selected: This option allows for future developers or users to define their own peak tracking algorithm. To do this, open the m-script of the MATLAB GUI program and search for (crtl+f) "user-defined peak tracking". This should lead to the section in which users can insert their own code.



3.2.3. Clear raw spectra

Clicking this will clear all of the axes in the GUI. Note that this does not delete any data; instead, it is useful when the GUI is in an akward state and the axes need to be refreshed.

3.2.4. View raw figure mode

The raw spectra of the selected harmonics will be displayed in separate figure windows. This is useful for quickly exporting the raw spectra.

3.2.5. Clear datapoints

This will clear all of the datapoints that was collected. Deleted datapoints can be manually recoved in the "deleted data" folder, which is located in the same directory as the MATLAB GUI program. The name of the deleted data is based on the timepoint of deletion: "<YYYYMMDD>T<hhmmss>_deleted_data.mat".

3.3. Settings panel

The settings panel contains user interfaces to manipulate the harmonic scanning range, the time resolution between each scan, and each set of harmonic measurements. Also, basic options related the fitting process can be set in this panel. Further, this is the panel where the filename is designated and/or appended.

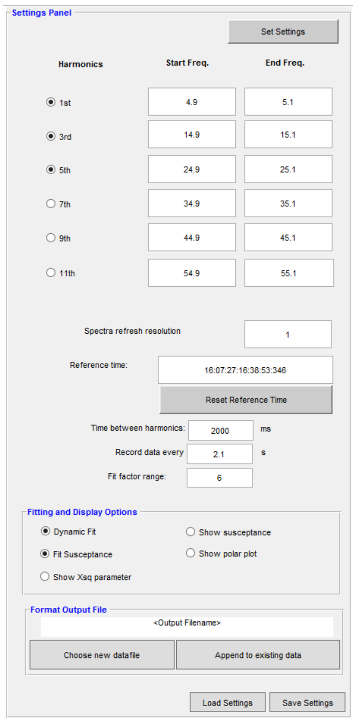


Figure 19: The settings panel.

It is often useful to save the "Settings" of an experiment, so that the user is not required to re-input all of the experimental parameters associated with the measurement process. Previously saved settings can be loaded by clicking on the "Load Settings" button. Note that only .mat files with the "_settings" appended at the end of the filename can be used to load in the settings: "<filename>_settings.mat"

3.4. Raw spectra panel

The raw spectra panel contains the conductance and susceptance plots of each harmonic. Other information such as the peak tracking algorithm, the initial fitting guess parameters, and data resolution are included. The brown-orange orizontal line represent the range of data used for the fitting process. The location and peak width of the resonance peak are also plotted for each harmonic.

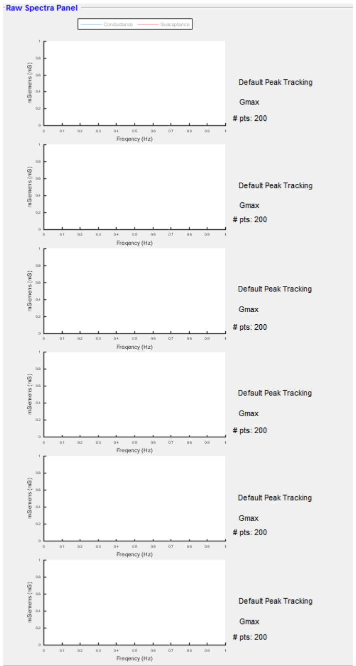


Figure 20: The raw spectra panel.

3.5. Main plot panel

The main plot panel contains data associated with the fitted frequency and dissipation shifts. The harmonic dataset can be selected by enablin/disabling the radio dials in each of the subpanels.

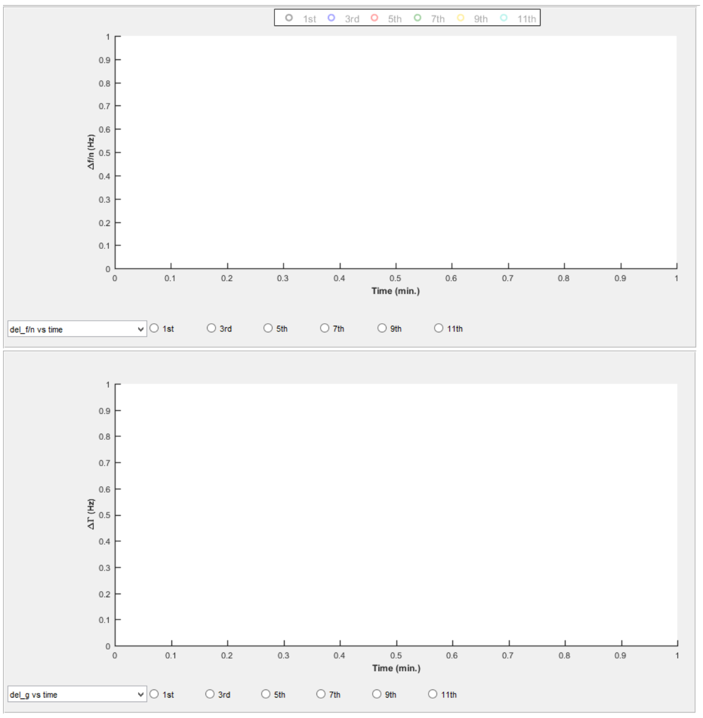


Figure 21: The main plot panel.

3.6. Status toolbar

The status toolbar can be found at the bottom of the main GUI window. This toolbar reflects the current state of the program. When everything is running properly, the toolbar foreground and background should be red and black, respectively. When an error occurs, the foreground and background colors are black and red, respectively. Any warning messages in the status bar will have a red and yellow color for the foreground and background color respectively. This toolbar will also have other useful information such as the total number of datapoints that have been collected, and the most recent timepoint that was recorded. Also, a wait bar has been added in between sets of harmonic measurements when the "Record data" time is greater than 10 s. Also, the status bar will display the total number of errors the program encounters during the measurement process.

4. Troubleshooting

4.1. MyVNA

Here are some things to try and check if an error associated with MyVNA occurs.

- Make sure that the "servers" have been properly registered (with administrator priviledges). This is important in establishing communication between the AccessMyVNA and MyVNA programs. See Section 1.1.1 for more details.
- If the calibration cannot be properly loaded, it is best to redo the calibration process.

4.2. AccessMyVNA

Here are some things to try and check if an error occurs in the AccessMyVNA program. Note that for version 002c and above, AccessMyVNA is hidden and runs in the background. Use the task manager to determine whether this program is actively running.

- If AccessMyVNA is running on Windows 8 or higher, ensure that compatability mode has been enable for Windows 7. See Section 1.1.1 for more details.
- Make sure that the "servers" has been properly registered. This is important in establishing communication between the AccessMyVNA and MyVNA programs. See Section 1.1.1 for more details.

4.3. MATLAB GUI

When an error appears stating that MATLAB is unable to close or delete the GUI window, make sure that the current MATLAB working directory is the same in which the .m and .fig file of the GUI is located. This error occurs because a custom close request function for exiting the program was written; thus, MATLAB needs to be able to access this close request function. Alternatvely, the program can be manually closed by using the "delete" function:

```
delete(<figure handle to GUI>)
```

Note that the figure handle to the GUI can be obtained with the "findall" function:

```
findall(0,'type','figure','tag','primary1)
```

Part III. Design overview

5. Goal and purpose of the QCM MATLAB program

The primary goal for the developement of this program is to provide a MATLAB interface in obtaining QCM measurements. Specifically, this program has the capability to measure the location and half-max-half-width (HMHW) of the resonance conductance peaks. Based on this information, analysis can be performed (in MATLAB) on these measurements to extract out the viscoelastic properties of a film deposited on the QCM crystal. Currently, error bars are not explicitly calculated (only χ^2values for the peak fitting is calculated); however, one can easily calculate error statistics since all of the raw conductance/susceptance spectra data is saved in a ".mat" .

6. Hardware and software components

6.1. Hardware

The hardware required to run the program is the N2PK Vector Network Analyzer (N2PK-VNA, an impedance analyzer custom built by Ivan Markarov), a crystal holder (Inficon, NY), and a QCM crystal (Inficon, NY). Further information on the N2PK-VNA hardware details can be found in http://g8kbb.co.uk/html/myvna.html. An image of the setup can be seen in Figure 22.

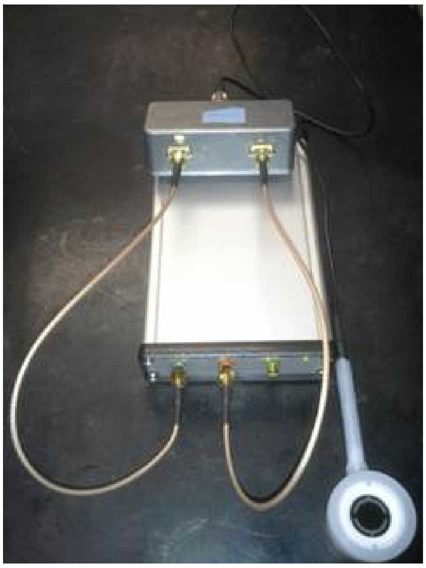


Figure 22: An image of the overall hardware setup.

6.2. Software

The N2PK-VNA came with a proprietary software called myVNA. This software allows for remote access, which provides a means to control the N2PK-VNA hardware with a custom program. Currently, the program called AccessMyVNA (written in VB C++) is used to remote-access the impedance analyzer. AccessMyVNA is designed as a "middleman" or a "gateway" program for the QCM MATLAB program to communicate with the N2PK-VNA hardware. Details on how the programs communicate with each other is discussed in subsequent sections. To summarize, in order to collect QCM measurements with the QCM MATLAB program, three programs must be runnning simultaneously: 1) MyVNA, 2) AccessMyVNA, 3) QCM MATLAB program. An image if each program is shown in Figure 23.

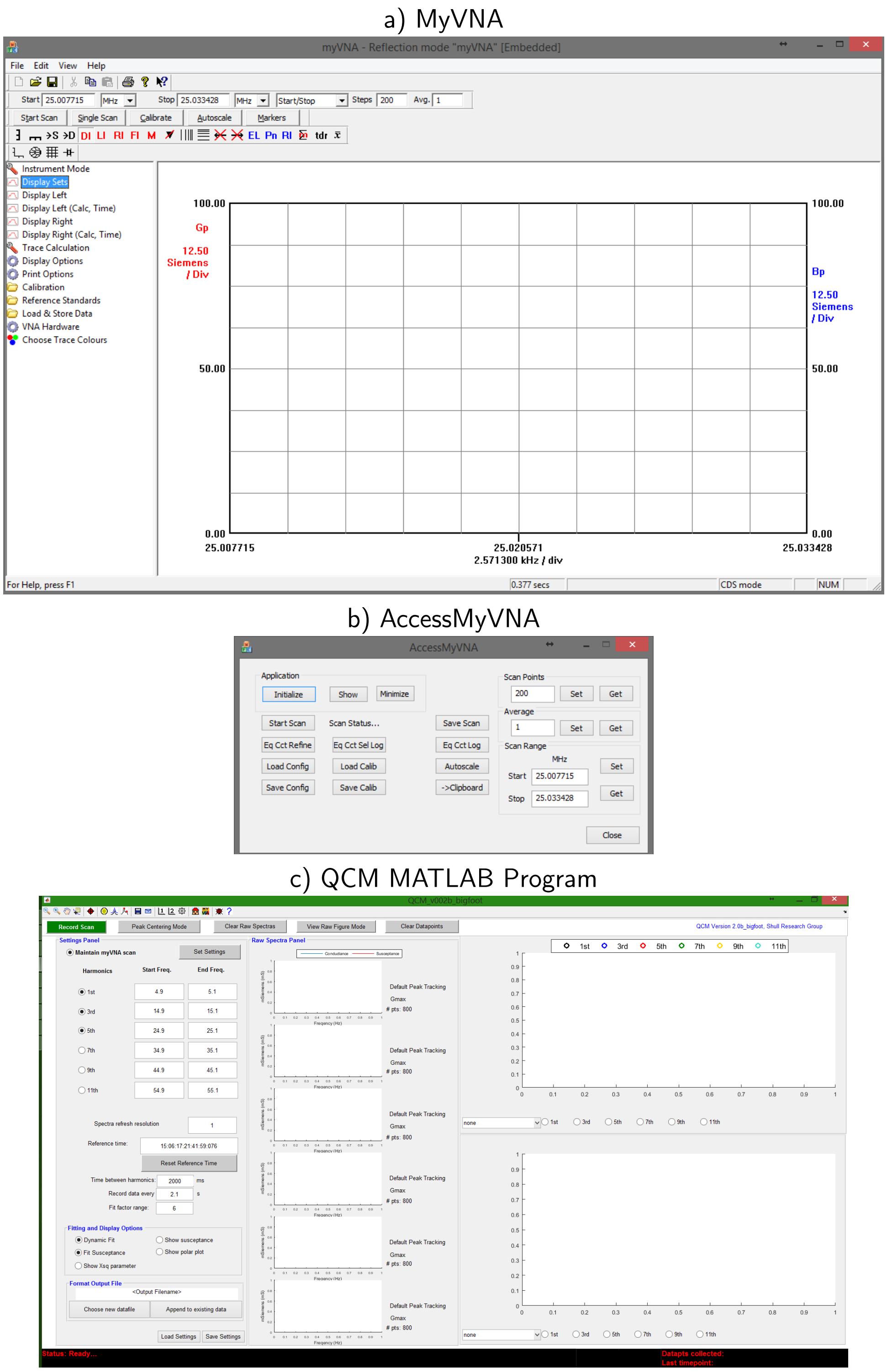


Figure 23: A screen shot of each program: a) MyVNA, b) AccessMyVNA, c) QCM MATLAB Program

6.3. Design overview of how the programs work

A general overview of how all the programs communicate with each is shown in flow chart in Figure 24.

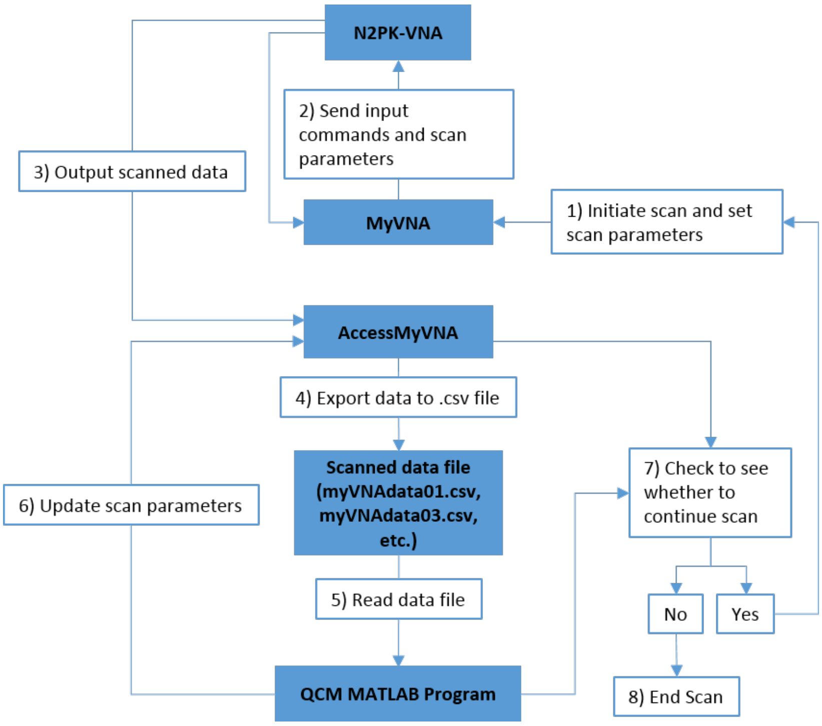


Figure 24: A flow chart of how the three programs communicate with each other.

As shown in Figure 24, the QCM MATLAB Program does not directly communicate with the impedance analyzer, instead the MATLAB program directly manipulates AccessMyVNA to control the impedance analyzer. A more detailed interaction between AccessMyVNA and the QCM MATLAB Program is shown in Figure 25.

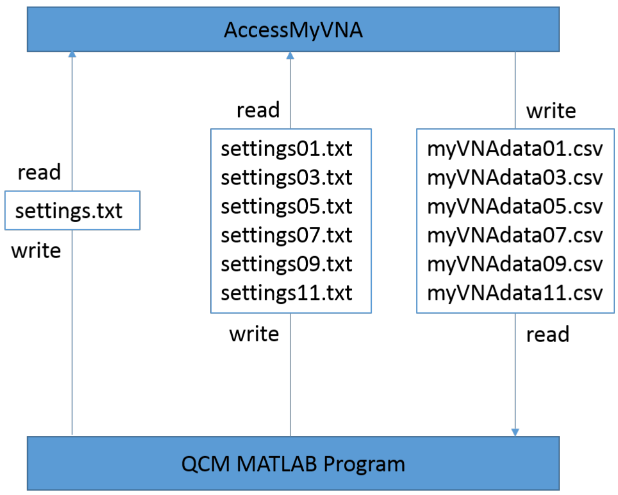


Figure 25: A more detailed flowchart how the QCM MATLAB program communicates with AccessMyVNA.

Part IV. MATLAB code overview

7. Record scan algorithm explained

Since the program heavily relies on the details on how the measurement process is run, a section has been dedicated in explaining how the MATLAB script was written. When an unforseen error or bug occurs, this section is a good place in finding the origin of the problem.

8. Index of functions

This section lists all of the functions that were written and used in the GUI code. The intent of this section is to provide a comprehensive list of each function. Function are group based on its type of function. Also note that this list corresponds to the functions used in version 002e.

8.1. Functions that are explicitly initialized during the GUI start-up process

These functions are always initialized and "pre-made" when a GUI script is created with the MATLAB GUIDE. Some modifications were made to these "prewritten" functions in order to initialize default parameters properly.

- varargout = QCM_v002d_Eurynomos(varargin)
- QCM_<Program version>_OpeningFcn(hObject, eventdata, handles, varargin)
- varargout = QCM_v002d_Eurynomos_OutputFcn(~, ~, handles)

8.2. Measurement and scanning processes

8.2.1. Settings

- set_settings_Callback(~, ~, handles)
- write_settings(handles,harm_num)
- save_settings_Callback(~, ~, handles)
- load_settings_Callback(hObject, ~, handles)

8.2.2. Data acquisition

- start_Callback(hObject, ~, handles)
- [freq,conductance,susceptance,handles]=read_scan(handles)
- handles=save_data_Callback(hObject, ~, handles)
- maintain_myVNA_Callback(~, ~, handles)

8.3. Plotting

- handles=raw_fig_Callback(hObject, ~, handles)
- plot1_choice_Callback(~, ~, handles)
- plot2_choice_Callback(~, ~, handles)
- plot_primaryaxes1(handles,FG_frequency,harm_tot,n)
- plot_primaryaxes2(handles,FG_frequency,harm_tot,n)
- plot_1_Callback(~, ~, handles)
- plot_3_Callback(~, ~, handles)
- plot_5_Callback(~, ~, handles)
- plot_7_Callback(~, ~, handles)
- plot_9_Callback(~, ~, handles)
- plot_11_Callback(~, ~, handles)
- plot2_1_Callback(~, ~, handles)
- plot2_3_Callback(~, ~, handles)

- plot2_5_Callback(~, ~, handles)
- plot2_7_Callback(~, ~, handles)
- plot2_9_Callback(~, ~, handles)
- plot2_11_Callback(~, ~, handles)
- num_harms=primaryaxes_harm(handles)
- num_harms=primaryaxes2_harm(handles)
- **uipushtool6_ClickedCallback**: this function uses the "inspect" function to allow manipulation of the primary axes plots 1
- **uipushtool7_ClickedCallback**: this function uses the "inspect" function to allow manipulation of the primary axes plots 2

8.4. Fitting procedure

- [G_fit,B_fit,G_l_sq,B_l_sq,combine_spectra,G_parameters,B_parameters,handles,I]= Lorentzian_dynamic_fit(handles,freq,conductance,susceptance,combine_spectra)

8.4.1. Initial guessing algorithms

- [guess,f0,gamma0]=G_guess(freq,conductance,susceptance,handles,ylab)
- [guess,freq_mod,modulus,f0,gamma0]=deriv_guess(freq,conductance,susceptance,handles)
- flag=preview_peak_identification(freq,ydata,index,ylabel_str,handles)

8.4.2. Lorentz fitting procedure

- [G_fit,G_parameters,G_l_sq,B_fit,B_parameters,B_l_sq]=fit_spectra(handles,raw_data,guess,I)
- [G_parameters,B_parameters]=par_check(G_parameters,B_parameters)
- [fitted_y,residual,parameters]=fit_spectra_con(x0,freq_data,y_data,I,show_GB,lb,ub)
- [fitted_y,residual,parameters]=fit_spectra_sus(x0,freq_data,susceptance_data,I,show_GB,lb,ub)
- [fitted_y,residual,parameters]=fit_spectra_both(x0,freq_data,conductance,susceptance,num_peaks, I,handles,lb,ub)
- F_conductance = lfun4c(p,x) (No longer included in v002e[3])
- F_susceptance = lfun4s(p,x) (No longer included in v002e[3])
- F_conductance = lfun4c_2(p,x) (No longer included in v002e[3])
- F_susceptance = lfun4s_2(p,x) (No longer included in v002e[3])
- F_conductance = lfun4c_3(p,x) (No longer included in v002e[3])
- F_susceptance = lfun4s_3(p,x) (No longer included in v002e[3])
- fcns=lfun4_both_1(p,x) (No longer included in v002e[3])
- fcns=lfun4_both_2(p,x) (No longer included in v002e[3])
- fcns=lfun4_both_3(p,x) (No longer included in v002e[3])

[3] Lorentz functions containing the code for the equations have been moved to another m-script as anonymous functions (for easy access to the codes). The functions can be accessed in the "AccessMyVNAv0.7\release\Lorentz_eqns.m" file.

8.4.3. Fitting options

- radio_chi_Callback(~, ~, handles)
- dynamic_fit_Callback(~, ~, handles)
- fit_B_radio_Callback(hObject, ~, handles)

8.4.4. Refitting algorithm

- refit_ClickedCallback(hObject, eventdata, handles)
- refit_start_Callback(hObject, ~, handles)
- refit_inc_Callback(~, ~, handles)
- refit_end_Callback(hObject, ~, handles)

8.4.5. Refitting selected datapoints

- output_txt=select_spectra_fcn(hObject,event_obj)
- refit_select_spectra(spectra,handles,harm,timepoint,index)
- guess_method_callback(hObject,~,harm,handles)
- rss_close(hObject,~)
- accept(hObject,~)
- select_spectra_fit(hObject,~,handles,f1,guess_method,spectra,harm)
- update_tick_display(~,~,handles)
- zoom_out_ClickedCallback(hObject, eventdata, handles)
- zoom_in_ClickedCallback(hObject, eventdata, handles)
- select_spectra_ClickedCallback(hObject, eventdata, handles)
- output_txt=default_dcm(hObject,event_obj)

8.5. Peak Centering/Tracking Functions

- peak_centering_Callback(hObject, ~, handles)
- center_peak_function(handles,harm,hObject)
- peak_center_SelectionChangeFcn(hObject, eventdata, handles)
- store_num_data(hObject,~,handles,p)
- span_adjust(~,~,handles,p,factor,set_span)
- manual_set_span(~,~,handles,p,set_span)
- peak_tracking_flag(~,~,handles,radio_handles,flag,p)
- custom_peak_track_flag(~,~,handles,radio_handles,p)
- store_guess_options(~,~,handles,guess_handle,p)
- [handles]=smart_peak_tracker(handles,freq,conductance,susceptance,G_parameters)
- handles=confirm_peak_finding(~, ~, handles)
- [handles]=peak_finding_ClickedCallback(~, ~, handles)
- fpo(hObject,~,handles)
- fp_close(hObject,~,handles,find_peak_options)

8.5.1. Harmonic radial dial selection

- center1_Callback(hObject, eventdata, handles)
- center3_Callback(hObject, eventdata, handles)
- center5_Callback(hObject, eventdata, handles)
- center7_Callback(hObject, eventdata, handles)
- center9_Callback(hObject, eventdata, handles)
- center11_Callback(hObject, eventdata, handles)

8.5.2. Toolbar buttons

- my_closereq(~,~,handles,freq,radio_handles,guess_values_options,f1,p)
- myL_fit(~,~,handles,p,statistics_txt)
- refresh_button(~,~,handles,p)
- myzoomfcn(~,~,handles,set_span,p)
- output_txt = myupdatefcn(~,event_obj,handles,set_span,p)

8.6. Other Callback functions

- primary1_CloseRequestFcn(hObject, ~, handles)

8.6.1. Settings callback functions

- harm1_Callback(~,~,handles)
- harm3_Callback(~,~,handles)
- harm5_Callback(~,~,handles)
- harm7_Callback(~,~,handles)
- harm9_Callback(~,~,handles)
- harm11_Callback(~,~,handles)
- start_f1_Callback(~, ~, handles)
- end_f1_Callback(~, ~, handles)
- start_f3_Callback(~, ~, handles)
- end_f3_Callback(~, ~, handles)
- start_f5_Callback(~, ~, handles)
- end_f5_Callback(~, ~, handles)
- start_f7_Callback(~, ~, handles)
- end_f7_Callback(~, ~, handles)
- start_f9_Callback(~, ~, handles)
- end_f9_Callback(~, ~, handles)
- start_f11_Callback(~, ~, handles)
- end_f11_Callback(~, ~, handles)

- set_reference_time_Callback(~, ~, handles)
- polar_plot_Callback(~, ~, handles)
- fit_factor_Callback(hObject, ~, handles)
- show_susceptance_Callback(~, ~, handles)
- wait_time_Callback(~, ~, handles)
- num_datapoints_Callback(~, ~, handles): this function is mostly defunct

8.6.2. GUI toolbar callback functions

- handles=cla_raw_Callback(hObject, ~, handles)
- clear_datapoints_Callback(hObject, ~, handles)
- load_bare_ClickedCallback(hObject, ~, handles)
- save_shifts_ClickedCallback(~, ~, handles)
- exe_vna_ClickedCallback(hObject, ~, handles)

8.6.3. Email functions

- email_notification_Callback(~, ~, handles)
- test_email(hObject,~,handles)
- set_email_options(hObject,~,handles,outserver,host_email,host_email_pw,txt,toggle_email,email)
- toggle_func(hObject,~,handles,txt)
- email_push_ClickedCallback(hObject, eventdata, handles)
- email_send(handles,message)
- handles=reset_fcn(hObject,eventdata,handles)
- home_push_ClickedCallback(hObject, eventdata, handles)
- del_mode_ClickedCallback(hObject, ~, handles)
- confirm_del_ClickedCallback(hObject, ~, handles)
- append_data_Callback(hObject, eventdata, handles)
- debug_ClickedCallback(~, ~, handles)
- uipushtool8_ClickedCallback(~, ~, ~)
- show_dfdg_callback(~,~,handles,show_dfdg,status)

8.6.4. Preference functions

- pref_ClickedCallback(hObject, ~, handles)
- schedule_table_callback(~,callbackdata,handles,schedule_table)
- reset_row_callback(~,~,handles,reset_row,schedule_table)
- del_row_callback(~,~,handles,del_row,schedule_table)
- add_row_callback(~,~,handles,add_row,schedule_table)
- scheduler_onoff(~,~,hObject,handles,radio_on_off_schedule,schedule_table,add_row, del_row,reset_row,status,flag)
- set_pref1(~,~,hObject,handles,set_pref,radio_GB_values,radio_clc_cw,radio_output_raw, radio_simul_peak,schedule_table,radio_on_off_schedule,show_dfdg,status,radio_diary)

8.7. Misc

- **pause_func(~,~,handles)**
- **harm_tot=find_num_harms(handles)**
- **check_freq_range(harm, min_range, max_range, handles)**
- **refreshing(handles,harm,flag)**
- **pause_func1(~,~,handles)**
- **refresh_button2(~,~,handles,p)**
- **num_peaks_check(~,~,num_peaks_edit)**
- **my_disp(msg,color)**
- **ins(hObject,~)**

9. List of versions released

Version control is managed by GitHub, under the username Shull-Research-Group.

1. version 001b
2. version001c
3. version002a
4. version002b_bigfoot
5. version002c_Cthulhu
6. version002d_Drakon
7. version002e_Eurynomos

Part V. Acknowledgements

This program is heavily influenced by the many features in the QTZ.exe written by Diethelm Johannsmann, Institute of Physical Chemistry, TU-Clausthal, Germany. The author of this MATLAB program is very thankful for his contribution to the QCM community and theory. Further, many thanks to the members of the Shull Research Group, Northwestern University, and Professor Kenneth Shull for his guidance and design into the developement of this QCM MATLAB program. Any comments or suggestions are welcome! CHJoshuaYeh@u.northwestern.edu