QCM MATLAB Program Manual

The Shull Research Group
Version 1.0a and above

Chyi-Huey Joshua Yeh

04/06/2014

Contents

I. Quickstart Guide 2

1. Installation 2
1.1. Installing MyVNA program . . . 2
1.1.1. Step-by-step myVNA installation . . . 3
1.2. Installing AccessMyVNAv0.7 program . . . 3
1.3. Installing the QCM MATLAB Program . . . 4

2. How to take measurements 4
2.1. Starting up and closing down the programs . . . 4
2.2. Setting up parameters and measurement options . . . 5
2.2.1. Choosing which harmonic to track . . . 5
2.2.2. Spectra refresh resolution . . . 6
2.2.3. Reference time . . . 6
2.2.4. Time between harmonics . . . 6
2.2.5. Number of datapoints . . . 7
2.2.6. Fitting range factor (for version 1.0c and above) . . . 7
2.2.7. "Record data every _ s" option . . . 7
2.2.8. Fitting options . . . 8
2.2.9. Formatting output file . . . 9
2.2.10. Loading and saving settings . . . 11
2.3. Record scan . . . 11
2.3.1. The raw spectra panel . . . 11
2.3.2. Frequency/dissipation shifts plots . . . 12
2.4. Output files . . . 13
2.4.1. No designated output filename and file location . . . 13
2.4.2. Designated output filename and file location . . . 14
2.5. Troubleshooting . . . 15

3. Basic features of the QCM MATLAB Program 15
3.1. Status bar and the MATLAB command window . . . 15
3.2. Accessing global workspace and the QCM MATLAB GUI "handles" structure . . . 16
3.2.1. Global workspace . . . 16
3.2.2. QCM MATLAB GUI "handles" structure . . . 16

3.3. Peak centering . . . 18
3.3.1. Initial-guess algorithm for Lorentzian fitting . . . 18
3.3.2. Peak tracking algorithms . . . 19
3.3.3. For versions 1.0c and above . . . 19
3.4. Plotting . . . 19
3.5. Raw figure mode . . . 19
3.6. Lorentzian curve fitting . . . 20
3.6.1. Calculating goodness-of-fit, χ^2 . . . 20
3.7. Email notifications (versions 1.0b and above) . . . 21

4. Troubleshooting **21**

II. Introduction to the design overview **21**

5. Goal and purpose of the QCM MATLAB program **21**

6. Hardware and software components **21**
6.1. Hardware . . . 21
6.2. Software . . . 22
6.3. Design overview of how the programs work . . . 22

III. QCM MATLAB Program **25**

7. Record scan algorithm explained **25**

8. Brief description of each function in the QCM MATLAB GUI Program (not complete) **25**

IV. Aknowledgements **28**

Part I.
Quickstart Guide

1. Installation

1.1. Installing MyVNA program

The MyVNA program is a software, developed 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. **Install Visual Studio (2013 or higher).** This program is needed in order to manipulate the code for AccessMyVNA. Free licenses to this program are available to students. Visit https://www.dreamspark.com/ for more information.

    a) If manipulating code for AccessMyVNA is not necessary, simply run either *vcredit_x64(1).exe* (for Windows 64-but operating systems) or v*credit_x32(1).exe* (for Windows 32-bit operating systems). Both files can be found in the *Visual C++ redistributable packages for visual basic 2013* folder. These execution files can be found online. This process simply adds a required .dll file to run the AccessMyVNA program.

2. **Open up the folder name, "myVNA setup files".**

3. **Run myVNA.msi and follow the directions.** This installs the myVNA program that directly communicates and controls the N2PK Vector Network Analyzer (imepdance analyzer).

4. **After installing myVNA, double click on myVNA_setup.reg.** This adds registry values associated with the myVNA program.

5. **Go to the folder in which the myVNA program is installed.** The path usually looks something like this: C:\Program Files (x86)\G8KBB\myVNA.

6. **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 as the 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.

7. **In the myVNA 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).

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 crreates 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.

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.

1.3. Installing the QCM MATLAB Program

The QCM MATLAB Program does not require any formal installation. Make sure that the following files:

1. "QCM_v001a.m"
2. "QCM_v001a.fig"
3. "deleted_data.mat"
4. "fg_values.mat"
5. "raw_spectras.mat"
6. "other.mat"

are in the<C:\Program Files (x86)\QCM MATLAB Program\> path directory.

To run the QCM MATLAB Program, double click on the "QCM_v001a.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

2.1. Starting up and closing down the programs

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 1). 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.

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.

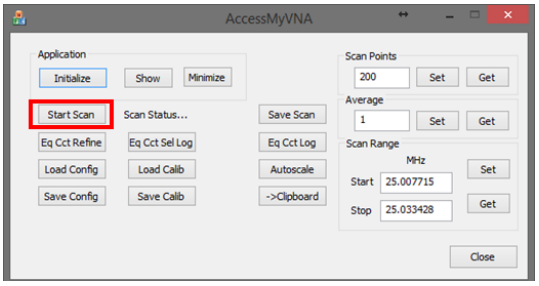


Figure 1: Click on the "Start Scan" button in the AccessMyVNA program to begin scanning.

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 2).

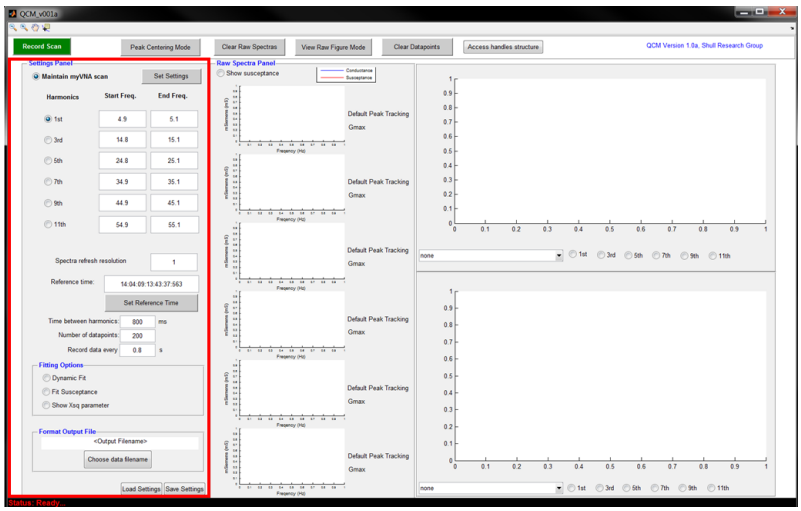


Figure 2: The settings panel contains information on the parameters, settings, and options related to the collection of the raw data.

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.

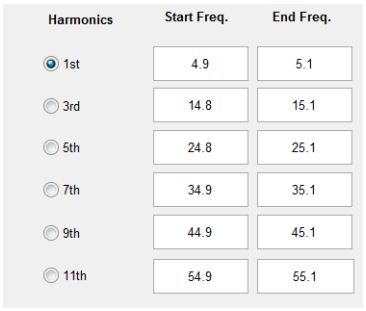


Figure 3: Multiple harmonics can be tracked. Choose which harmonics to track by checking/unchecking the radio dials under the "Harmonics" column. The start and end frequencies of the scan can be set for each harmonic.

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 "Set Reference Time" button after editing the reference time. Clicking on the "Set 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 4.

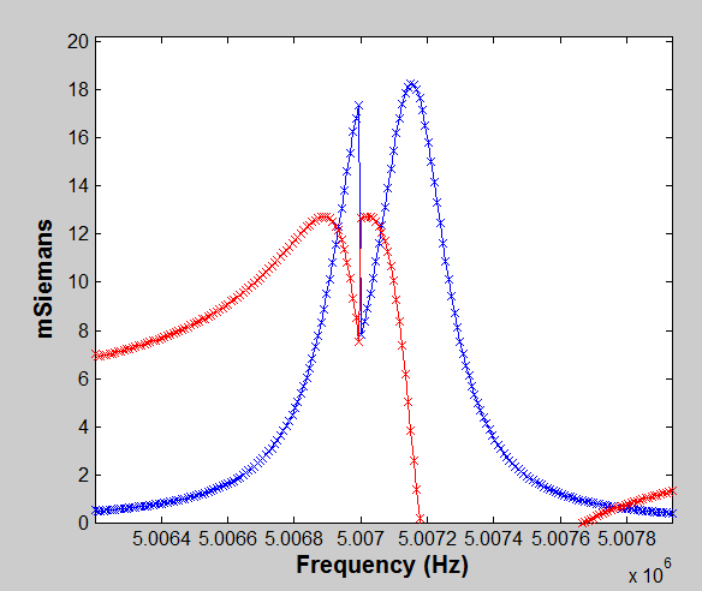


Figure 4: An unsynchronized resonance peak (5MHz) due to insufficient amount of time in between harmonics. Increase the time in between each harmonic measurement to solve this problem.

2.2.5. Number of datapoints

The number of datapoints in each frequency scan can be adjusted. The default value is 200 datapoints. Note that increasing the number of datapoints will require more time in between harmonics. Thus, the time in between harmonics need to be adjusted accordingly. **A good rule of thumb is to allow 4-6 ms for each datapoint.**

NOTE: For versions 1.0c and above, this function has been removed. To change the number of data points, go to the figure that is produced from the "Peak Tracking Mode". The number of datapoints can be set for each individual harmonic. For example, it is possible to scan 400 data points for harmonic 1, 200 data points for harmonic 3, and 800 data points for harmonic 5.

2.2.6. Fitting range factor (for version 1.0c and above)

An option to change the fitting range factor is replacing the number of datapoints feature. This factor determines the range in which the Lorentz fitting will take places. Speccifically, the range is determined by taking the fitting range factor and multiplying by one half of the width of the resonance peak. A typical value for the fitting range factor is 3-4. Note that if the fitting range factor is large, the fit will be affected by data points that do not contribute to the Lorentzian resonance peak. This can adversely bias your calculated frequency and dissipation shift.

2.2.7. "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 5.

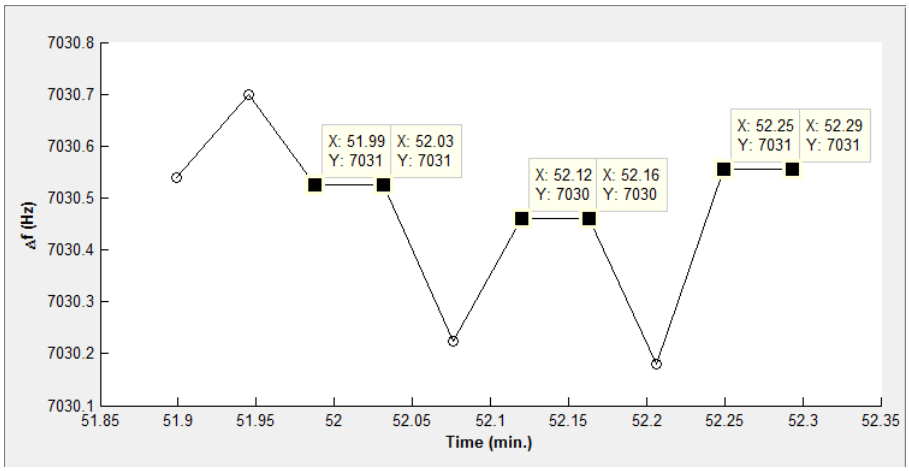


Figure 5: If the recording time interval is less than the time in between harmonics, duplicates of a datapoint will occur. This is evident if the freqeuncy shift plot is observed during the measurement process.

2.2.8. Fitting options

There are three radiodials in the "Fitting Options" panel (see Figure 6): "Dynamic Fit", "Fit Susceptance", and "Show Xsq parameter". 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 discussed in later sections.

For versions 1.0b there are new features on how the raw spectra data is displayed in the "Raw Spectra Panel". In particular, the susceptance can be plotted on a double axes plot and the spectra data can ploted in polar coordinates (susceptance versus conductance).

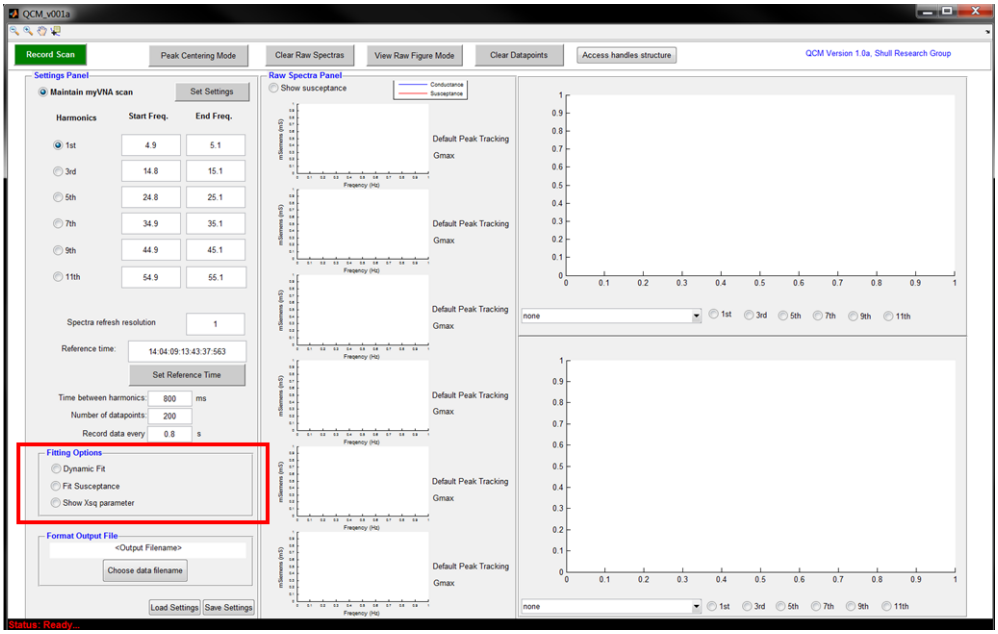


Figure 6: Options related to the fitting process during the measurements can be adjusted in the "Fitting Options" section located in the "Settings" panel.

2.2.9. 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). From this dialogue box, the name and location of the output data can be chosen. **If a pre-existing file is chosen, any collected absolute frequency/disspation and frequency/dissipation shift data will be overwritten unless the settings associated with the output datafile is also loaded.** If the frequency/dissipation data were 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.** See the section on "Loading and saving settings" for more informationon how the data is stored in the output datafile.

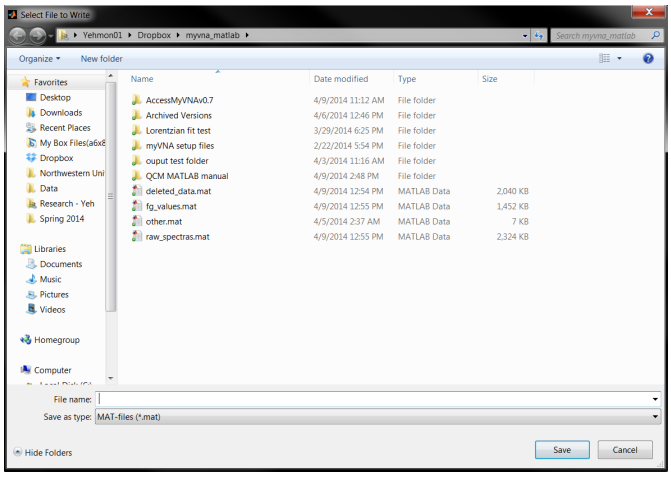
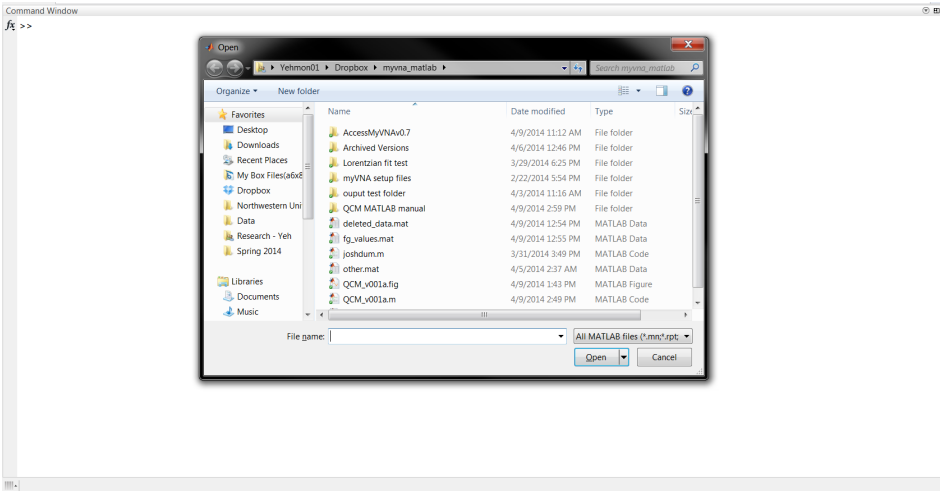


Figure 7: File-output explorer dialogue box.

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:

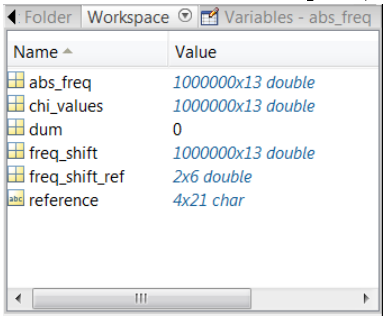
1. Click on the "Access handles structure"button.

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 the file-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



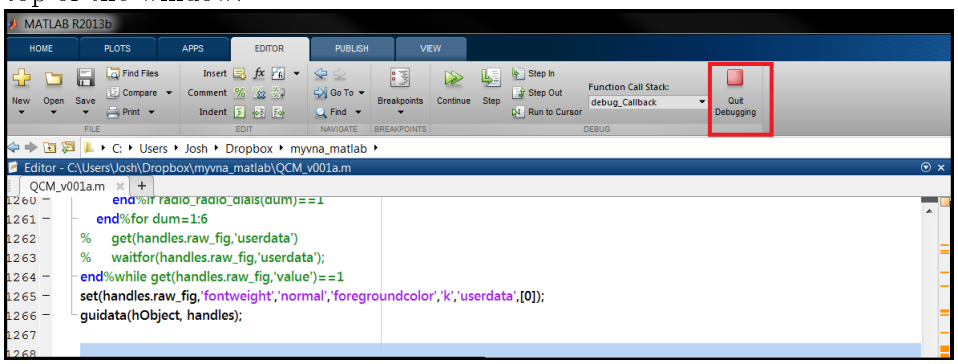
5. This will open up the "Variables" dialogue box containing the data stored in the variable that was double-clicked. Find the row number associated with the last datapoint that was stored.

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. In the MATLAB Workspace, double-click on reference. Copy the first line of characters. This line of string should be in the same format as the required format for the string input in the reference text box in the "Settings Panel". Paste this string of characters into the reference text box in the "Settings Panel". **This will ensure that the recorded timepoints are in reference to the initial reference time from the previous dataset.**

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



2.2.10. 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 associated with the output datafile 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". This usually means that the file that was selected is not a "settings file". 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.

2.3.1. The raw spectra panel

During the recording process, the raw spectra for each harmonic will be shown in the "Raw Spectra Panel". To view both the conductance and susceptance raw spectra, check the "Show susceptance" radio dial in the "Fitting and Display Options panel. Also, the raw spectra can be plotted on a polar plot (susceptance versus coductance). Next to each raw spectra plots, the

peak tracking algorithm and the initial-guess fitting algorithm is shown. As suggested from the set-up the peak tracking algorithm and the initial-guess algorithm can be customized for each harmonic. If the "Show Xsq parameter" radio dial has been selected, the χ^2 parameter will be shown next to each raw spectra plot

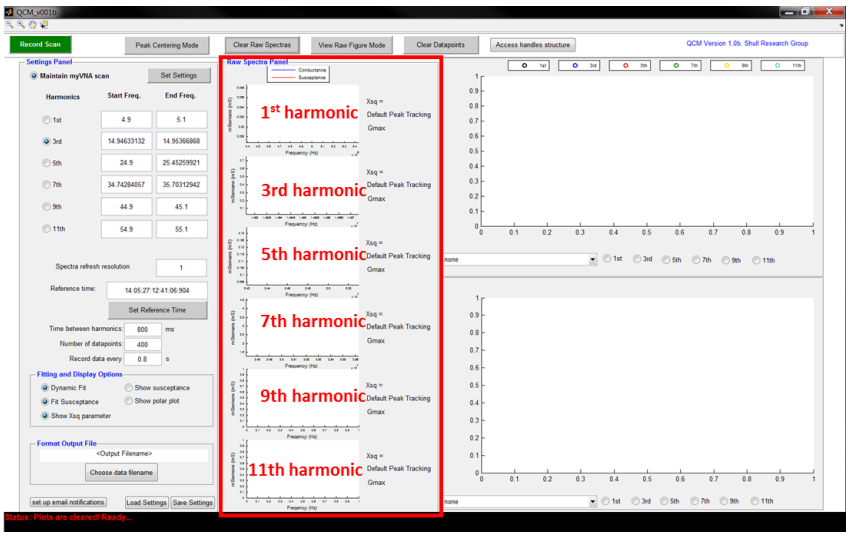


Figure 8: The Raw Spectra Panel

For versions 1.0b and above, new raw spectra features have been added to aid with the measurement process. Specifically, for each raw spectra plot that are being fitted with a Lorentzian curve (aka, the "Dynamic Fit" radio dial has been selected), the range in which the Lorentzian curve is being fit, the fitted frequency and width of the peak will be plotted. This ensures that the fitting algorithm is properly fitting the raw spectra data during the measurement process.

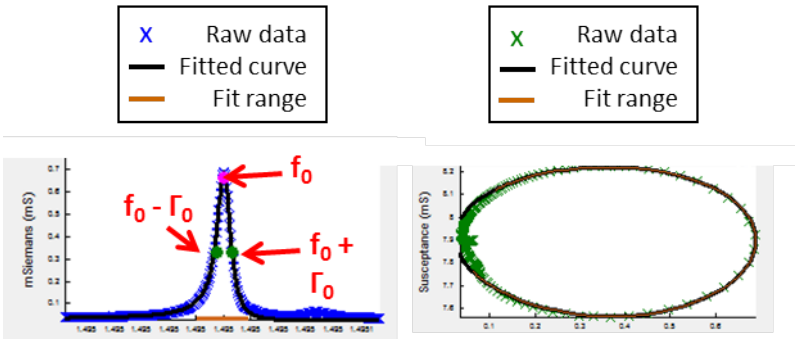


Figure 9: Plotting features in the raw spectra plots.

2.3.2. Frequency/dissipation shifts plots

During the measurement process, the fitted frequency shifts and dissipation shifts can be plotted. Multiple harmonics can be plotted at the same time and the plot type can be chosen (Δf versus *time*, $\Delta f/n$ versus *time*, $\Delta\Gamma$ versus *time*, or $\Delta\Gamma/n$ versus *time*).

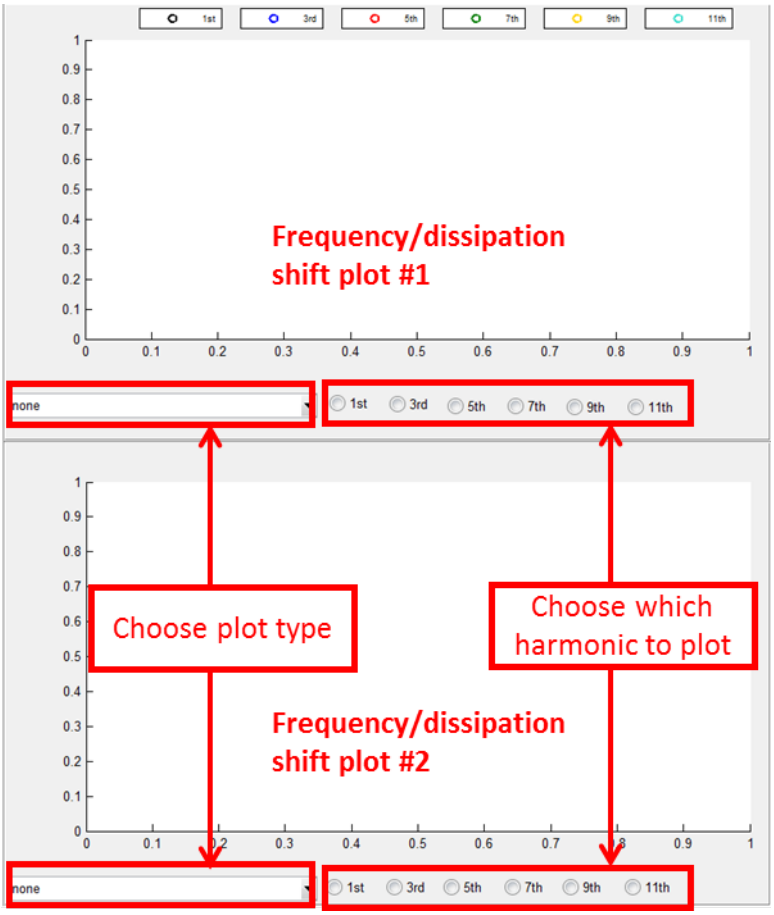


Figure 10: Frequency/dissipation shifts plots

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.

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>.

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.
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 access the QCM MATLAB GUI handles structure is by pushing the "Access handles structure" button. This allows the user to access the current GUI handles state.

3. Basic features of the QCM MATLAB Program

3.1. Status bar and the MATLAB command window

The status bar is a useful feature that provides updates on what the GUI is currently doing. Also, this stats bar will provide error and warning messages. It is good practice to key an eye on this status bar. This feature is not perfect, but it will definitely give you clues to sources of error when the MATLAB program is malfunctioning.

3.2. Accessing global workspace and the QCM MATLAB GUI "handles" structure

3.2.1. Global workspace

There are several variables that are exported into the global workspace while the MATLAB program collects data. They are listed below

1. "freq": This variable contains the frequency range from the most recent collected raw spectra data.

2. "conductance": Thisis variable contains the conductance values from the most recent collected raw spectra data.

3. "susceptance": This variable contains the susceptance values from the most recent collected rae spectra data.

4. "error_log": When MATLAB encounters an error, it attempts to continue on with the measurement (as opposed to prematurely terminating the measurement process). When this happens, it records what type of error it encountered. This error log is then exported into the global workspace as "error_log". For more information on common errors the MATLAB program can encounter, please refer to section on "Troubleshooting".

5. "fid1": This variable will appear in the global workspace when the MATLAB program fails to read the raw spectra data file from the AccessMyVNA program. Specifically, this variable is the "file identifier" associated with opening the raw datafile using the fopen function. For more detailed information regarding to file identifiers, refer to the MATLAB help-documentation on fopen.

6. "handles": When the "Access handles structure" button is pressed, the MATLAB program will automaticall export the handles strucutre of the GUI into the global workspace. This functionality was added for convenience if one needed to debug/access/modify the MATLAB program.

3.2.2. QCM MATLAB GUI "handles" structure

The handles strucutre contains all the information that is constantly updated and saved throughout the whole process when the QCM MATLAB GUI is running. To access the handles structure, simply push the "Access handles structure" button. This will put MATLAB in a "debugging" mode. To exit out of the "debugging" mode, just press the "Quit Debugging" button on the Editor toolbar or type in "return" in the command window.

The way in which all of the data are being stored in the handles structure can be overwhelming (but it is actually quite simple!). A diagram is shown below, which displays an overview of how the data are organized.

A description of what types of data is stored in each of the variables in the handles.din structure is shown below.

1. "freq_range": This variable contains the lower and upper bound frequencies associated with each harmonic. This variable is important in checking to make sure that the user does not choose to scan a frequency range that is oustside of what is typically exceptable frequency range of the associated harmonic. For example, the MATLAB program will throw back an error if the user decides to scan from 7-8 MHz for the first harmonic, since a typical exceptable frequency range associated with the first harmonic is between 4 and 6 MHz. The lower and upper bound can be modified for special experimental setup.

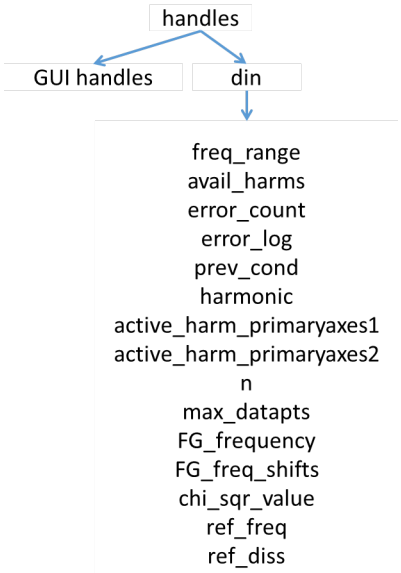


Figure 11: A diagram of how the handles structure is organized

2. "avail_harms": This variable lists out the available harmonics the MATLAB program can support. Currently, it is the 1st, 3rd, 5th, 7th, 9th, and 11th harmonic (assuming the quartz crystal that is being used is a 5 MHz crystal).

3. "error_count": This variable contains the number of errors the program encountered. This variable has a default value of 1 (although, no errors are encountered by default).

4. "error_log": This variable contains a log of errors that the MATLAB program has encountered.

5. "prev_cond": This variable contains the conductance values from the previously scanned raw spectra.

6. "harmonic": This variable contains the current harmonic that is being scanned.

7. "active_harm_primaryaxes1": This variable contains the number of harmonics that will be plotted in the handles.primaryaxes1 axes.

8. "axtice_harm_primaryaxes2": This variable contains the number of harmonics that will be plotted in the handles.primaryaxes2 axes.

9. "n": This variable containes the number of recorded datapoints + 1.

10. "max_datapts": This variable contains the maximum number of datapoints that can be stored by the MATLAB program.

11. "FG_frequency": This variable contains the fitted absolute frequency and dissipations values.

12. "FG_freq_shifts": This variable contains the fitted frequency and dissipation shifts.

13. "chi_sqr_value": This variable contains information on the goodness-of-fit of the Lorentzian fit.

14. "ref_freq": This variable contains the absolute reference frequency values in which the frequency shifts are calculated from.

15. “ref_diss”: This variable contains the absolute reference dissipation values in which the dissipation shifts are calcualted from.

There are other variables that pertains to the Lorentzian fitting parameters and how the resonant peaks are being tracked. For the purposes of creating a “simple” manual, details to what these variables represent will not be discussed. Careful examination of the code will provide the context of the purpose of these variables. Also, feel free to email CHJoshuaYeh@u.northwestern.edu for details on these variables.

3.3. Peak centering

This feature was included with the purposes of locating harmonic peaks before the measurement recording process. To access “Peak centering” mode, push the “Peak centering mode” button. Then, in the “Settings Panel”, choose which harmonic to locate. A figure window will pop up that shows a plot of the raw spectra.

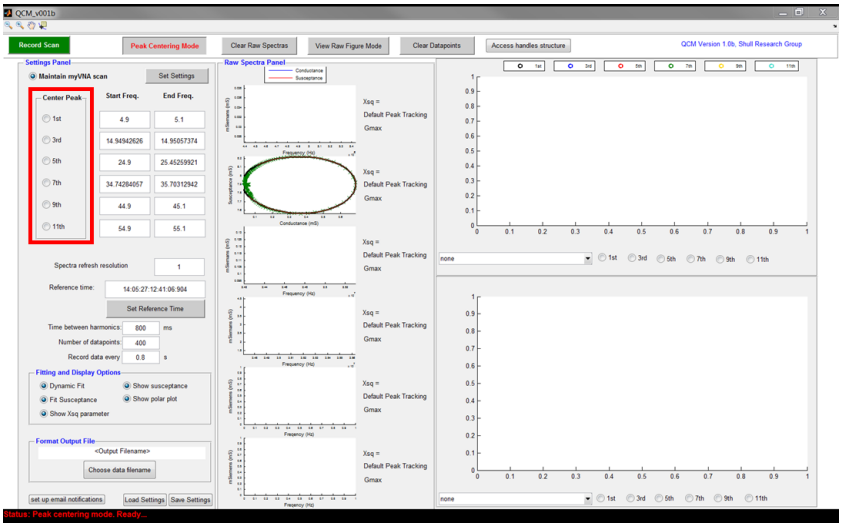


Figure 12: Choose which harmonic to locate by selecting the desired radio dials in the “Settings Panel”.

To locate the harmonic peak, increase or decrease span of the window or input a desired span in the “Span” panel in the figure window. The pan tool and/or zoom tool can also be used to locate the peak. To center a peak, use the datacursor tool to choose the datapoint that will be designated as the “center” of the peak. Then, click again on axes to refresh the plot (Note: the plot will not refresh if the second click is on a plot line or marker).

While the plot is refreshing during this process, a waitbar will appear, counting down the amount of time that is left for the plot to completely refresh. It is advised to wait for the countdown to finish before manipulating the plot window again. Otherwise, there can be synchronization issues. If the plot looks suspicious, plot can be refreshed manually by clicking on the “refresh” button (see the figure below).

3.3.1. Initial-guess algorithm for Lorentzian fitting

Once a peak has been located, a preliminary fit can be performed by pushing the “Fit” button. The initial-guess algorithm can be chosen in the popup menu below the “Span” panel. The three initial-guess algorithms are:

1. Gmax: This method uses the maximum conductance from the conductance spectra as the initial guess value for the fitting algorithm.

2. Derivative: This method takes the "peak" value from the derivative of the polar plot (susceptance versus conductance) as the initial guess value for the fitting algorithm.

3. Previous values: This method takes the previous fitted values (if any) as the initial guess values for the fitting algorithm.

3.3.2. Peak tracking algorithms

The algorithm in which the MATLAB program will track the resonance peak can be selected from this figure. There are three radiodials that can be selected (Fix span, Fix center, and Custom) to set the peak tracking behavior.

1. Fix span: The span of the frequency range will be held fixed throughout the measurement process. However, the MATLAB program will adjust the start and end frequencies in order to track the peak.

2. Fix center: The center of the frequenc range will be held fixed throughout the measurement process. However, the MATLAB program will adjust the span of the frequency range in order to track the peak.

3. Fix span and fix center (aka both the "Fix span" and the "Fix center" radio dials are selected): The center and the span of the frequency range will be held fixed throughout the measurement process. When this option is set, a warning message will pop up as areminder that the MATLAB program will not be able to track the peak if the peak moves outsize the frequency range. This option can potentiall lead to unwanted systematic errors. However, there are special situations in which this option can be useful during the measurement process.

4. Custom: Currently, there are no code written in this section. This option allows the user to define their own custom peak tracking algorthm in the MATLAB script. To do this, simply go into the "smart_peak_tracker" function in the MATLAB GUI script. At the end of the function code block, there is a comment section labelled as, "%%%%%%%CUSTOM, USER-DEFINED". Edit this block of code to define a custom peak tracking algorithm. Note that this option is designed for advanced users who want more control over how the resonance peaks are being tracked during the measurement process.

3.3.3. For versions 1.0c and above

The number of datapoints to scan for the harmonic can be set for each harmonic. The option to set the number of datapoints from the main GUI figure has been removed. See Section 2.2.5 for more details.

3.4. Plotting

Refer to Section 2.3.2.

3.5. Raw figure mode

This feature is still much under construction (enable this feature during recording at your own risk!). However, this feature currently allows the user to see the raw spectras in individual windows. To enable the "Raw figure mode" simply push the "Raw figure mode" toggle button.

3.6. Lorentzian curve fitting

The following equations are used to fit the resonance peaks to a Lorentzian function:

$$G_{fit} = G_{max}\left[\frac{f^2(2\Gamma)^2}{(f_0^2 - f^2)^2 + (2\Gamma)^2 f^2}cos(\theta) - \frac{f(f_0^2 - f^2)(2\Gamma)}{(f_0^2 - f^2)^2 + (2\Gamma)^2 f^2}sin(\theta)\right] + G_{off} \quad (1)$$

$$B_{fit} = G_{max}\left[\frac{f(f_0^2 - f^2)(2\Gamma)}{(f_0^2 - f^2)^2 + (2\Gamma)^2 f^2}cos(\theta) + \frac{f^2(2\Gamma)^2}{(f_0^2 - f^2)^2 + (2\Gamma)^2 f^2}sin(\theta)\right] + B_{off} \quad (2)$$

G_{max}: conductance at the peak of the Lorentz peak
f_0: frequency associated with G_{max}
Γ: half-max-half-width of the Lorentz peak
θ: rotation angle of circular plot in polar coordinates (dependent on the calibration state of the VNA)
G_{off}: vertical offset on the conductance curve (dependent on the calibration state of the VNA)
B_{off}: vertical offset on the susceptance curve (dependent on the calibration state of the VNA and the capacitance between the 2 electrodes deposited on the quartz crystal)

Note that is the VNA is properly calibrated, θ should be close to 0° and G_{off} should be close to 0. A graphical representation of the fitting parameters is shown below.

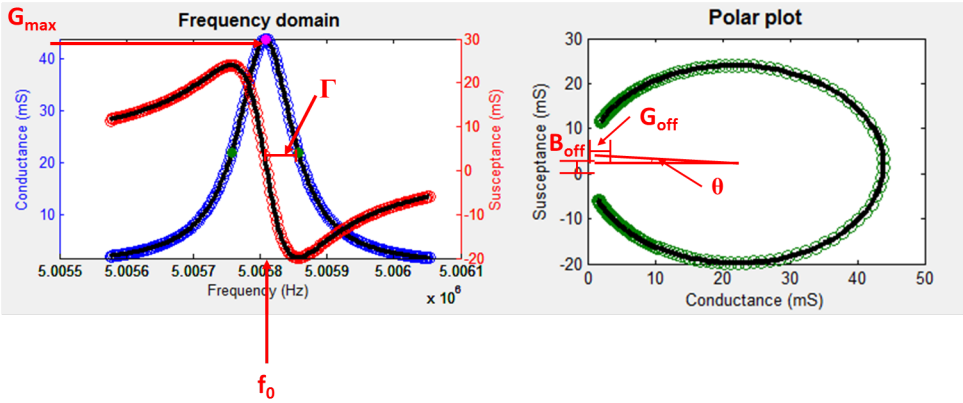


Figure 13: Graphical representation of the fitting parameters

3.6.1. Calculating goodness-of-fit, χ^2

Strictly speaking, the χ^2 statistic is defined as $\Sigma\frac{(y_{exp}-y_{fit})^2}{\sigma^2_{error}}/(total\ number\ of\ datapoints - 1)$. Thus, for a "good" fit will have a χ^2 value that is close to unity. Note that this definition of χ^2 imply that a statstic less than 1 indicates that the fitted values are fitting noise in the data. Currently, the calculation for the "χ^2" statistic is not actually the true χ^2 statistic, since the experimental error in the raw data is unclear (currently in the works). The way the "χ^2" value is calculated now is (strictly-speaking) the reduced-least-square value, $\frac{\Sigma(y_{exp}-y_{fit})^2}{(total\ number\ of\ datapoints-1)}$. Thus, a good fit will have a statistic close to 0. Note that this definition cannot differentiate whether or not the fit is fitting noise, since the statistic is also close to 0. In most cases, this is not an issue, since the raw spectral peaks are much larger than the baseline noise.

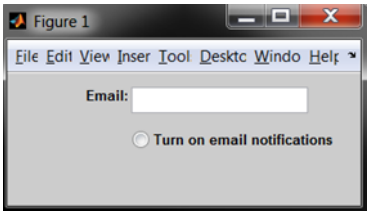


Figure 14: To set up email notifications, simply type in your email address and click on the "Turn of email notifications" radio dial. Then close out of the figure window.

3.7. Email notifications (versions 1.0b and above)

For version 1.0b and above, email notifications can be set up. This feature is useful when an experiment needs to be conducted overnight or for long periods of time. Email notifications will provide a diary of the command window about once every hour.

4. Troubleshooting

Part II.
Introduction to the 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 15.

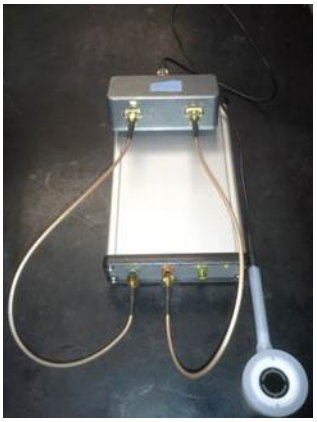


Figure 15: An image of the overall hardware setup.

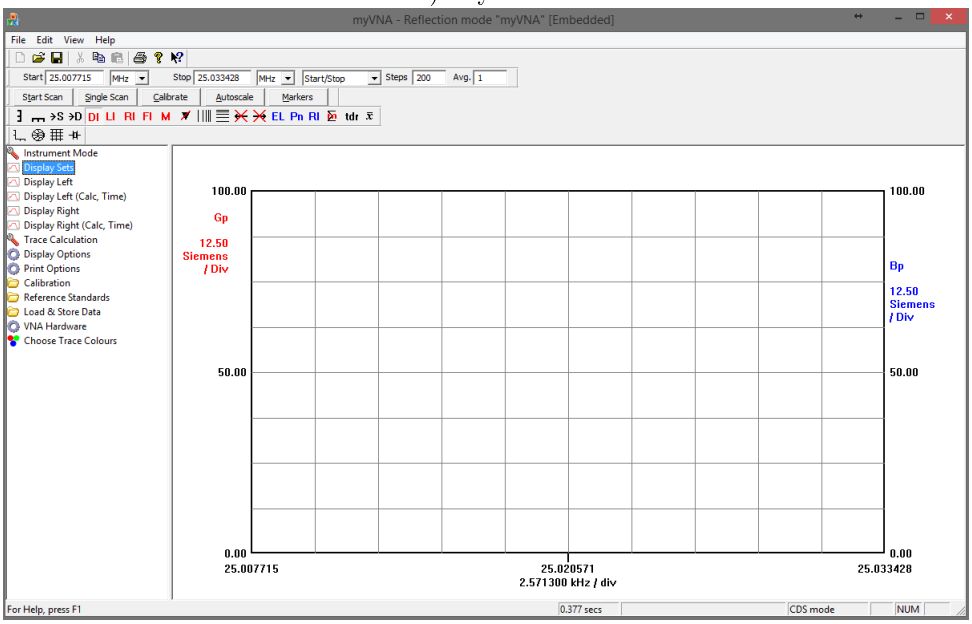
6.2. Software

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.

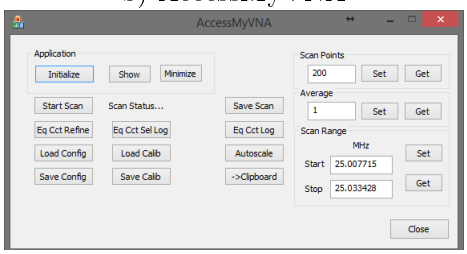
6.3. Design overview of how the programs work

A general overview of how all the programs communicate with each is shown in flow chart in Figure 17.

a) MyVNA



b) AccessMyVNA



c) QCM MATLAB Program

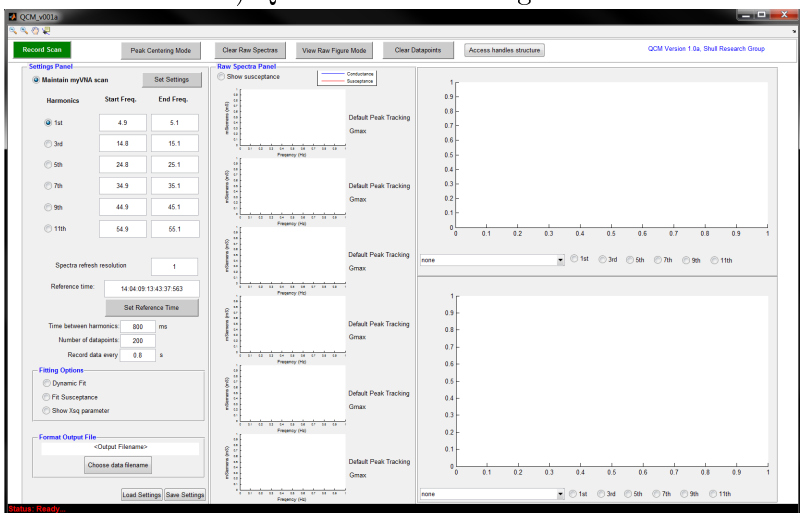


Figure 16: A screen shot of each program: a) MyVNA, b) AccessMyVNA, c) QCM MATLAB Program

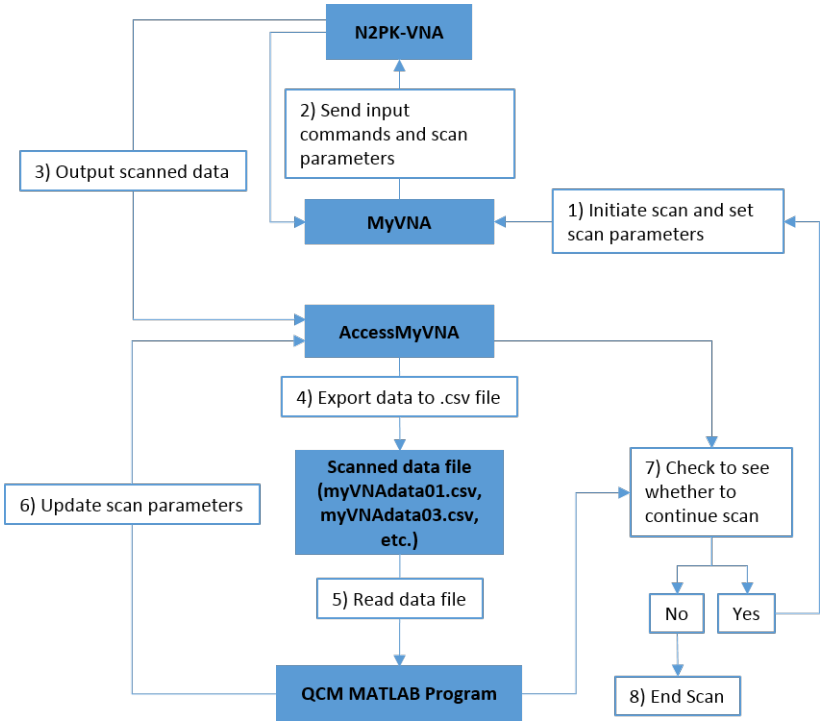


Figure 17: A flow chart of how the three programs communicate with each other.

As shown in Figure 17, 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 Figure18.

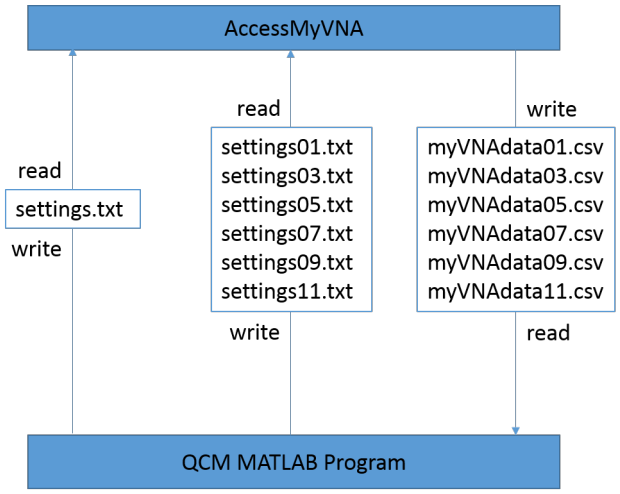


Figure 18: A more detailed flowchart how the QCM MATLAB program communicates with AccessMyVNA.

Part III.
QCM MATLAB Program

7. Record scan algorithm explained

(will include soon!)

8. Brief description of each function in the QCM MATLAB GUI Program (not complete)

Here is a list of functions in the code. A brief description of what each function does is described here (in alphabetical order).

1. QCM_vxxxx
2. center_peak_function
3. **check_freq_range:** Check whether or not the start/end frequency of the harmonic falls within range of acceptable limits.
4. **cla_raw_Callback:** This clear the axes figures displaying the raw spectras. It also clear the axes displaying the frequency shifts and dissipation shifts.
5. clear_datapoints_Callback
6. custom_peak_track_flag
7. **debug_Callback:** This will take the user into "debugging" mode via the MATLAB function, "keyboard".
8. dynamic_fit_Callback
9. email_func
10. email_notifications_Callback
11. eail_send
12. end_f11_Callback
13. end_f1_Callback
14. end_f3_Callback
15. end_f5_Callback
16. end_f7_Callback
17. end_f9_Callback
18. find_num_harms
19. fit_B_radio_Callback
20. fit_factor_Callback

21. fit_spectra
22. fit_spectra_sus
23. harm11_Callback
24. harm1_Callback
25. harm3_Callback
26. harm5_Callback
27. harm7_Callback
28. harm9_Callback
29. lfunc4c
30. lfun4s
31. load_settings_Callback
32. Lorentzian_dynamic_fit
33. maintain_myVNA_Callback
34. manual_set_span
35. my_buttons
36. my_closereq
37. myL_fit
38. myupdatefcn
39. myzoomfcn
40. num_datapoints_Callback
41. pause_func
42. pause_func1
43. peak_center_SelectionChangeFcn
44. peak_centering_Callback
45. peak_tracking_flag
46. plot1_choice_Callback
47. plot2_11_Callback
48. plot2_1_Callback
49. plot2_3_Callback
50. plot2_5_Callback
51. plot2_7_Callback

52. plot2_9_Callback
53. plot2_choice_Callback
54. plot_11_Callback
55. plot_1_Callback
56. plot_3_Callback
57. plot_5_Callback
58. plot_7_Callback
59. plot_9_Callback
60. plot_primaryaxes1
61. plot_primaryaxes2
62. polar_plot_Callback
63. primary1_CloseRequestFcn
64. primaryaxes2_harm
65. primaryaxes_harm
66. QCM_vxxxx_OpeningFcn
67. QCM_vxxxx_OutputFcn
68. radio_chi_Callback
69. raw_fig_Callback
70. read_scan
71. record_time_increment_Callback
72. refresh_button
73. refresh_button2
74. refreshing
75. save_data_Callback
76. **save_settings_Callback:** Save the settings and other metadat information into a designated .mat file.
77. **set_reference_time_Callback:** Set the reference time to the current time.
78. set_settings_Callback
79. show_susceptance_Callback
80. smart_peak_tracker
81. span_adjust
82. start_Callback

83. start_f11_Callback
84. start_f1_Callback
85. start_f3_Callback
86. start_f5_Callback
87. start_f7_Callback
88. start_f9_Callback
89. store_guess_options
90. store_num_data
91. toggle_func
92. **ukfunc:** Classified!
93. **ukfunc_ButtonDownFcn:** Classified!
94. **ukfunc_Callback:** Classified!
95. waitclose
96. **write_settings:** Write the settings for the scanning procedures into the appropriate text files.

Part IV. Aknowledgements

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