Philipp MALIN

Graphical User Interface and Data Handling in for a Nuclear Quadrupole Resonance Software Defined Radio Spectrometer

Bachelor's Thesis
to achieve the university degree of
Bachelor of Science
Bachelors's degree programme: Biomedical Engineering

submitted to

Graz University of Technology

Supervisor
Hermann Scharfetter Ao.Univ.-Prof. Dipl.-Ing. Dr. techn.

Institute of Biomedical Imaging
Head: Martin Uecker Univ.-Prof. Dipl.-Phys. Dr. rer. nat.

Graz, March 2022

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present bachelor's thesis.

Date Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Bachelorarbeit identisch.

Datum	Unterschrift

Abstract

This thesis describes, the control of an intuitive graphical user interface (GUI) to experiment on Nuclear Quadrupole Resonance (NQR) spectroscopy with an Software Defined Radio (SDR). Additional a automatic data handling system was implemented as it is essential for a efficient testing and measuring of the samples. A automated structured overview minimises the risk of experiment errors and the loss of important acquired data. Standard Sequences are predefined, furthermore a sequence generator has been implemented with the ability to dynamic select from one up to ten consecutive RF pulses consisting of timing and phase($\varphi_{min} = 0$ $\varphi_{max} = 2\pi$) orientation for individual customisation. Since one of the requirement was that the Software would be used for a lab exercise with students, it should include warning statements if destructive settings are saved to be run on the SDR. Since one of the requirements was that the software would be used for a lab courses with students, it should include warnings when destructive settings are saved and prevented to execution on the hardware. The settings of individual experiments can be saved in *.cfg file format from the system and reloaded back.

Contents

List of Figures

1 Introduction

Nuclear Quadrupole Resonance spectroscopy (NQR) is a technique to analyse samples in solid state with the absence of an external magnetic field as it is used in nuclear magnetic resonance (NMR). A radio frequency pulse is used to excite the samples. For this setup a software defined radio (SDR) [4] is used to generate the pulses. If the sample gets excited at its resonance frequency a gaugeable signal gets back induced into the coil. To mention is, that the received signal is a lot weaker than the excitation pulse to the simulation pulse. Based on an SDR based NMR system, leading developments and publications were made by Doll [1]. The system was further adopted by Kaltenleiter[2] at the institute of Biomedical Imaging at the TU Graz for NQR spectroscopy.

Single frequency measurements and broadband measurements are made in the very high frequency (VHF) range. [3]

The task of the this thesis was to implement an intuitive GUI for this existing setup with the SDR in an object oriented python script. Based on this graphical user interface (GUI), a user-friendly control of the SDR spectrometer should be achieved. Extending, also a sequence generator was implemented for a fast and easy selection of the sequence. All parameters for several standard sequences are predefined.In additionally for extended options, there is the option to set the own sequence in the GUI for designing a custom multiple pulse sequence. In the background, sending and receive data handling to and from the SDR is done automatically. Acquired results are immediately presented in the main window. Rounding the system off, a file handler for structured saving of all acquired data is implemented.

1.1 Preparations for future extensions

The software is structured in such a way that the software is later easy expandable. Special care has been made at predefining the automatic tune and matching unit so that it can be taken easily extended in the future. Further more thoughts the Software, options were made to implement the option of pulse shaping in the future.

2 Methods

All the software is available on the Git repository as in the download link below. Additionally there is also the description of the complete software in the wiki section. The GUI was written is Python 3.9.7 with the module Tkinter.
The software has been developed to run on a Linux machine, but a limited number of features also run on a windows system. This can be used, for example, to generate sequence files in advance or reload measurement data for reevaluation.

2.1 LimeNQR software Download link

Repository:
http://www.github.com/OE9NAT/bacharbeit

2.2 LimeNQR software Documentation links

User Wiki:
http://www.github.com/OE9NAT/bacharbeit/wiki

Developer documentation:
https://oe9nat.github.io/bacharbeit/

3 Results

3.1 What it can do

The Software has been designed to handle user inputs via a structured graphical user interface. A set of standard sequences are pre-configured and can be selected from the main window. There is also the possibility to generate individual own sequences. It will update the set sequence info to the main window. After selecting the data from the main window, if the resonant circuit is tuned, the measurement can be started with the specific sample.

Since the LimeSDR is a software defined radio with full duplex capability, the signal recording starts with the sequence. This can be achieved as it has two receivers and two transmitters implemented in the integrated circuit hardware. With the user set boundary conditions the recorded signal is cut to the desired size. The cut should be chosen in such a way that the re-induced signal will be completely enfolded in the recording. This information will be plotted and saved in the predefined file structure for further analyses.

3.2 Where are its limitations

Repetitive tuning and matching is done manually for measuring at every defined frequency. As this is a very tedious task and can be manually only done to a certain accuracy in a given time, an automatic system needs to be developed. As this feature is foreseen, arrangements have been made to easily implement this in the future.
On the other hand after the acquire, the information from the last run is shown in the plotter frame on the main window. This basic plotter provides a short preview of the raw signal, but for an additional analysis, cornerstone have bin set to developed a more detailed plotter.
For broad band frequency measurements safety precautions have to be made. For a new measurement for ascending frequency order, the old result will not be overwritten but will be appended until reaching the highest frequency. This type of acquire is special as this will fill in the plot from the starting frequency (lowest frequency) to the end frequency (highest frequency).

Error handling is only checked for plausibility. It is mainly done through structured logical entry of the parameters. Before saving all added and changed values, it will be checked that they are in the right format before saving them.

3.3 What can not be done yet

Fly to the moon.
It has the Potent to be more intuitive from setting the Parameters. While doing the experiment it can be more obvious, to put some settings and Parameters in different locations.
Error handling can be implemented for more cases. Such cases can be for example that Parameters are not checked for hardware specific limitations. In addition, there could also be an option to automatically correct Parameters which are set incorrect.

3.4 System evaluation

The test sequence was an FID with an offset of 3000 samples, pulse length of 3 µsec, start acquisition 22.5 µs, stop acquisition 42.5 µs. The Sample material was Triphenylbismut, $BiPh_3$. The manual tuning coil was used. The result are shown in the plot in the main window in fig 4.1.

4 Conclusion and Future Work

4.1 Main Window

In the main window all relevant data for the experiment can be presented and set. At the initial start of the system this window will be displayed. The sequence is also started from this main window with the RUN button.

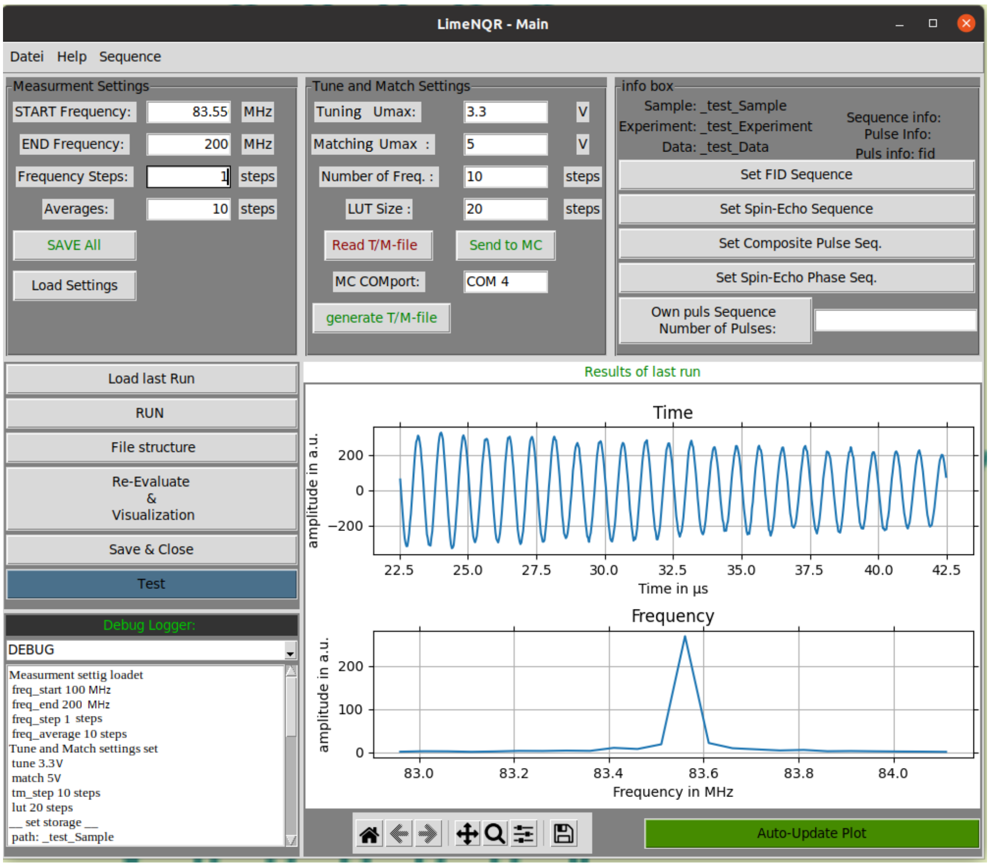

Figure 4.1: Main window, where all controls flow together

4.2 File Handle

All relevant file handling parameters can be initially changed from the default settings and updated in any later stage. At the beginning and end

of every experiment, comments can be added which will be logged to `comment_data.txt` and `comment_experiment.txt` file.
It is hard-coded to save all measurement data in the `Storage_vault` folder, but this can be changed very easily so that it can also be automatically stored to a decentralised file handling server.

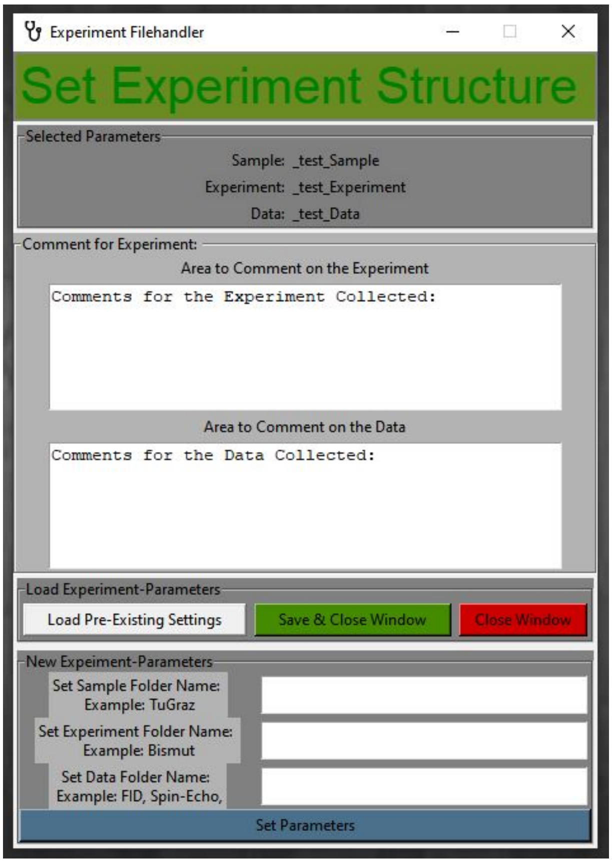

Figure 4.2: File handler window, used for editing the structure of storing the data

4.3 Sequence generator

All Standard sequences can be directly called from the main window. It includes the free induction decay (FID) sequence, the Spin-Echo sequence, the Composite Pulse sequence, the Spin-Echo phase sequence and a arbitrary own sequence generator to define your custom sequence. With the own sequence an arbitrary sequence can be generated with up to ten pulses and their respective offsets.

All standard sequences can be called directly from the main window. These include the Free Induction Decay (FID) sequence, the Spin-Echo sequence, the Composite Pulse sequence, the Spin-Echo Phase sequence and a custom sequence to be defined. The custom sequence can be used to create any sequence with up to ten custom multiple pulses and their respective offsets.

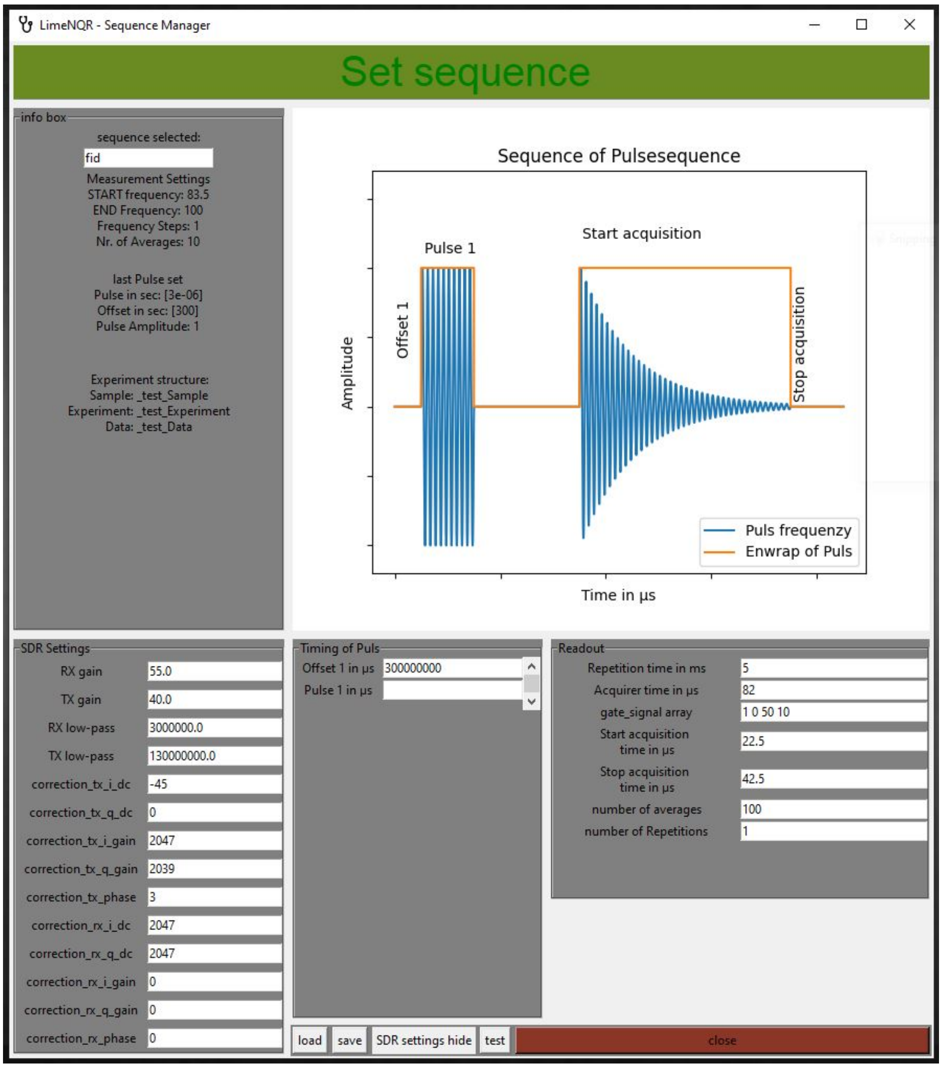

Figure 4.3: Sequence manager window, example configured specific for the FID sequence settings

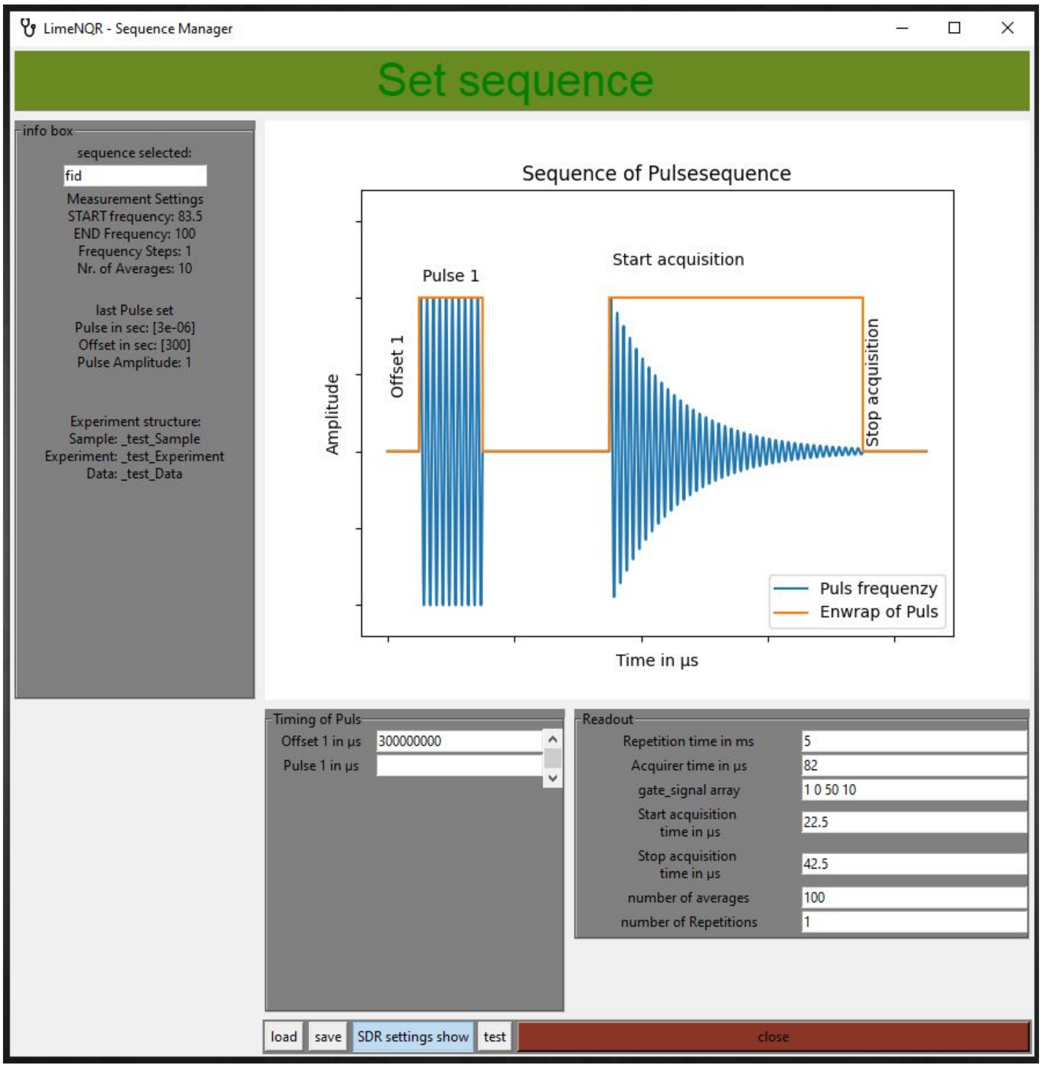

Figure 4.4: Sequence manager window, example configured specific for the FID sequence settings with the addition of masking the SDR Settings for a simplified overview

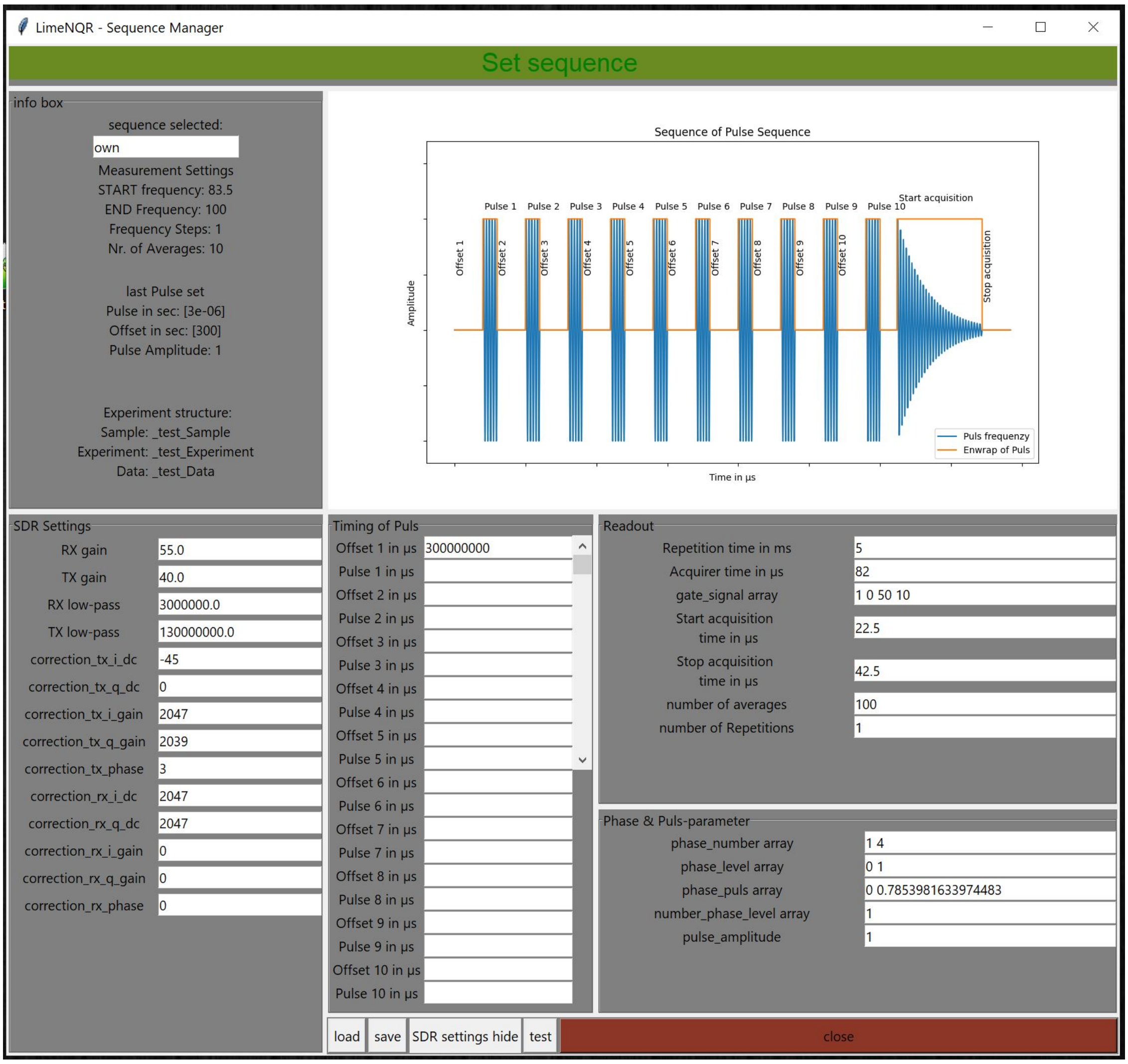

Figure 4.5: Sequence manager window, displaying the maximum number of 10 pulses and all possible set parameter

4.4 Future

There are numerous possibilities for expansion, as this interface is a first implementation for this Project to use an SDR for NQR analysis. Initially it was only a proof of concept and it will evolve into a stable and robust stand-alone system with numerous improvements from now on.

Manual tuning and matching is a very time consuming and repetitive task for a broadband measurements, therefor an automatic procedure is highly desirable. A basic interface for such a procedure has been provided and a respective protocol and control widget should be added in the near future.

A great potential lies in evaluating and comparing of the recorded measurements subsequently. A basic interface for such a procedure has been provided. A desirable feature would be the possibility of representing several data sets in one figure so as to compare them. The code has been written so that it offers the required flexibility. This could look like, multiple recorded files will be selected and loaded together into the same plot. The respective options are prepared to be easily implemented in the future.

The possibility for a better user experience lie in a continues amendment to the requirements. Possibilities for modification lie in the presentation of the work flow. For a modern design, this can be done through the coloring and shaping of buttons and input areas. Furthermore, there is great potential in the backend handling of variables and measurement data.

4.4.1 Puls shaping

The foundation for shaped pulses were made but more implementations need to be done. Dummy classes are written such that envelope vectors can be used to write arbitrary pulse shapes. All implementations have been made in such a way that there is a simple possibility of extension. Handling of such pulse shapes on the SDR needs to be verify.

4.4.2 Re-Evaluate Plotting

This feature can be called from the main window and allows it to read saved *.h5 files. When calling the window you are first prompted with a file selector. This reads in the file and plot its data. Further development needs to be made as this would have overloaded the amount of work for this thesis.

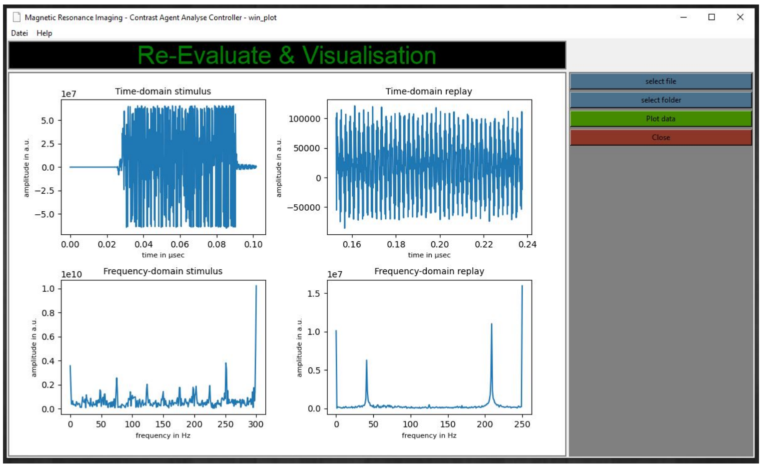

Figure 4.6: Re-Evaluate and Visualisation window, this has the potential to be developed into a visualisation and rework station for saved data.

Bibliography

[1] Andrin Doll. Pulsed and continuous-wave magnetic resonance spectroscopy using a low-cost software-defined radio. *AIP Advances*, 2019.

[2] Lukas Kaltenleitner. Software defined radio based nuclear quadrupole resonance spectrometer. *TU Graz*, 1(1):1–155, feb 2022.

[3] Hermann Scharfetter. An electronically tuned wideband probehead for NQR spectroscopy in the VHF range. *Journal of Magnetic Resonance*, 271:90–98, oct 2016.

[4] Wikipedia. Software defined radio.