MATLAB data acquisition

Documentation for the OPM data acquisition MATLAB scripts, originally written by Megan in 2021 for acquiring data using the NI-9205 Data Acquisition Unit.

Contents

[Data Acquisition UI 1](#_Toc86836242)

[UI elements 2](#_Toc86836243)

[Study protocols 4](#_Toc86836244)

[Open existing protocol 4](#_Toc86836245)

[Create a new protocol or edit an existing one 4](#_Toc86836246)

[Parameter choices 5](#_Toc86836247)

[Developers’ notes 5](#_Toc86836248)

## Data Acquisition UI

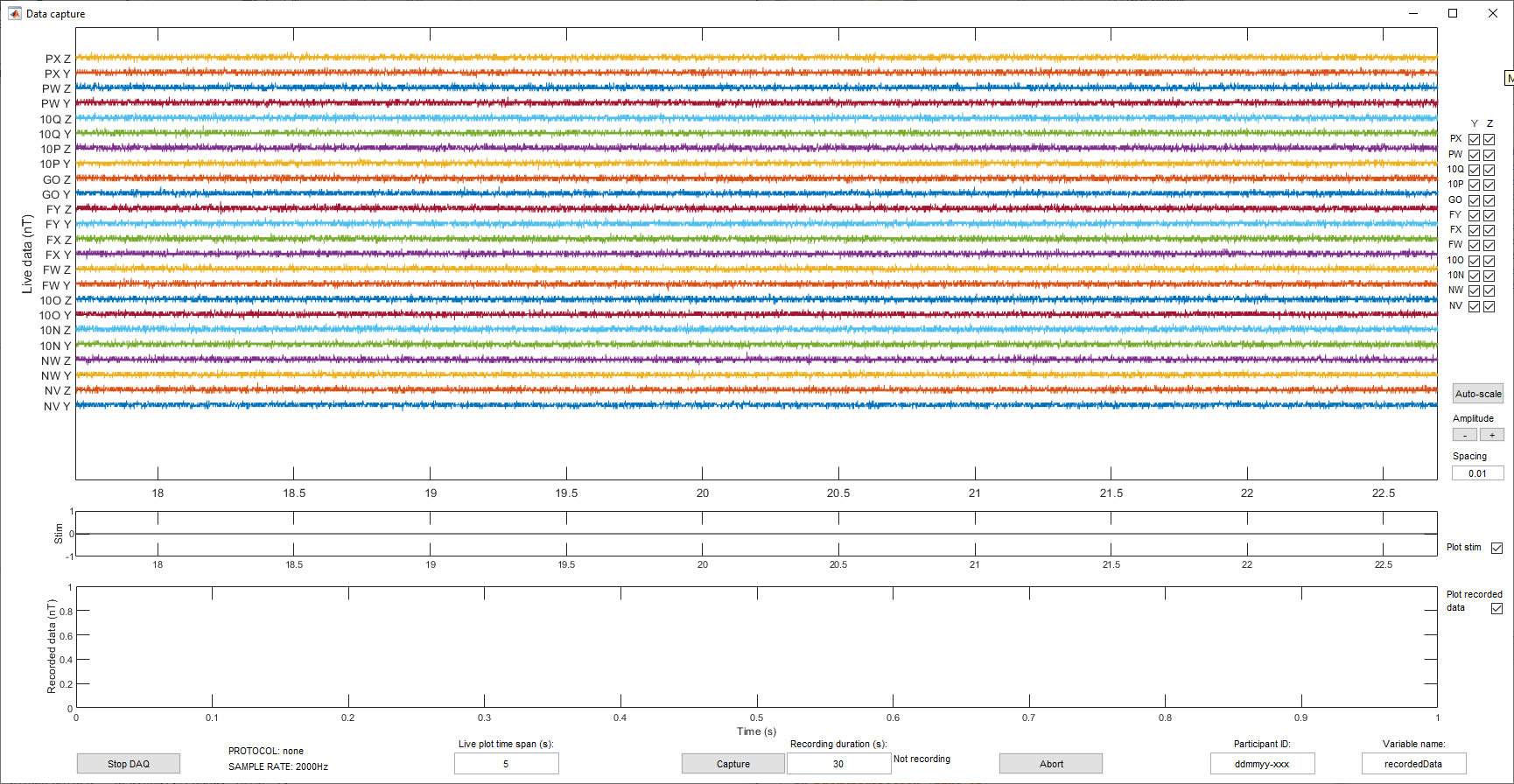
To acquire data, first ensure that the NI-9205 is plugged in to the computer via USB. Then, open MATLAB and run: *C:\Users\CUCHDS\Documents\MATLAB\MatlabDataAcq\DataAcquisition\DataAcquisition.m*

This will open a dialogue box asking whether you would like to use a study protocol. These are files that store parameters for data acquisition. See the ‘Study protocols’ section for more information about protocols, including how to create them.

Once a protocol has been selected (or if you have selected not to use one), the data acquisition UI (user interface) will open, as shown in Figure 1.

The text fields and the channel checkboxes on the UI will be populated with the parameters of the chosen study protocol. If no protocol was selected, by default a sample rate of 2kHz will be used.

### UI elements



*Figure 1: UI for data acquisition.*

**1**

**2**

**3**

**5**

**8**

**9**

**4**

**6**

**10**

**7**

**11**

**12**

**13**

This section gives some more information about the elements of the UI labelled in Figure 1.

#### Live plot

The live plot shows the real time OPM data. A couple of things to note:

* The data will not be recorded unless the ‘Capture’ button is selected.
* While the UI is open, data is constantly being acquired - even when dialogue boxes pop up after data recording and the plot appears stationary. After the box closes, the plot displays all the accumulated data at a faster rate until it has caught up with the current time. This is normal, although be aware that you are seeing data acquired in the past during this stage.

#### Channel labels

Each sensor is labelled using the last 2 or 3 characters of its serial number (visible on the front and top of the electronics boxes). Each sensor has two channels which represent the magnetic fields recorded along the sensor’s Y and Z axes. Record both components to maximise the number of channels.

#### Channel checkboxes

The channels shown can be selected using the checkboxes in the top right of the figure. Note that if a channel is selected/unselected, it will be included/excluded from **both** the live plot and any subsequent data recordings.

#### Amplitude scaling

These controls can be used to alter the amplitude of data shown on the live plot. Note: these controls have no effect on recorded data.

* The auto-scale button automatically determines the spacing between the channels based on the standard deviation of the data. This doesn’t change the displayed amplitude, just minimises space between channels.
* If auto-scaling is turned off, use the +/- buttons to change the displayed signal amplitude for either selected channels or all channels. Click on the lines on the live plot to select individual channels, and click again to deselect. If no channels are selected, amplitude will be altered for all channels.
* If auto-scaling is turned off, the value in the ‘Spacing’ text field determines the vertical spacing between the channels.

#### ‘Stim’ channel plot

This plot shows the triggers received from the stimulus presentation PC.

#### Plot display checkboxes

These can be used to show and hide the plots showing the stim channel and previously recorded data. Note that the stim channel will always be recorded during data capture regardless of whether the stim channel plot is visible. The recorded data plot will automatically appear on completion of data recording.

1. *Recorded data plot*

The bottom axes will show the recorded data at the end of a data recording. Until this time, the plot should be empty.

#### Stop DAQ button

This button stops the data acquisition. To restart, close the UI and run DataAcquisition.m again.

#### Protocol and sample rate

These are non-editable text fields showing which protocol is being used and the sample rate of data acquisition. To change these, close the UI and run DataAcquisition.m again choosing a different protocol.

#### Live plot time span

Edit this to change the time span of the data that is shown in the live plot.

#### Data recording

The OPM data can be recorded using the ‘Capture’ button, for the number of seconds shown in the editable ‘Recording duration’ text field.

The resulting data recording will consist of a 2D matrix of dimensions *No. Samples* x *No. Channels+2*, where the number of samples is *Recording duration*\**sample rate*. Each column corresponds to one channel. The first channel of the recording (the first column) gives the time stamps at which each sample was recorded. The last channel is the ‘stim’ channel which contains the triggers from the stimulus PC. The other channels contain the OPM data.

After recording, the data will automatically save to the MATLAB workspace as a variable. A dialogue box will also appear giving the option to save the dataset to file as a ‘.mat’.

If you would prefer the dataset saved in csv format, please see *C:\Users\CUCHDS\Documents\MATLAB\ConvertDataFormat.m*

The ‘Abort’ button can be used to interrupt data capture. The incomplete dataset will still be saved to the workspace, with the option to save the dataset to file.

#### Participant ID

Enter the participant’s ID number here. After data recording, if the data is saved to file, a filename will be automatically generated using the date, protocol name and subject ID.

#### Variable name

Edit this text field to define the name of the variable saved to the workspace after recording (containing the recorded data). This can be changed at any time before or during recording. If multiple recordings are captured during the same acquisition session, consecutive integers are automatically added to the variable name so that previous data won’t be overwritten (e.g., *recordedData*, *recordedData1*, *recordedData2* etc.). Note that this does not apply to data saved to file, which will always be saved using variable name *captureData*.

## Study protocols

By creating a protocol, you can ensure that you are visualising and recording data in a consistent way across different scans. Protocols used by this software are stored as .mat files and can store:

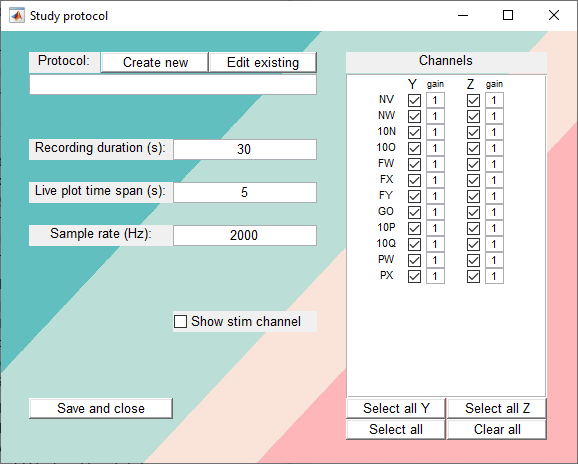
* Length of recording (s)
* Time span shown in the live data plot (s)
* Sample rate (Hz) *(see next page for more info)*
* Desired channels
* Amplitude scaling factors (gain) for data visualisation *(see next page for more info)*
* Desired stim channel

### Open existing protocol

If you wish to open an existing protocol that has already been saved, run DataAcquisition.m and then select ‘Load’ on the dialogue box. This will open a file explorer window. Navigate to the desired protocol and click ‘Open’. The UI for data acquisition will open and start visualising data using the parameters and selections stored in the protocol.

### Create a new protocol or edit an existing one

If you wish to create a new study protocol, or to edit an existing protocol, run DataAcquisition.m and select the ‘Create/edit’ button on the dialogue box. This will open the window in Figure 2.



*Figure 2: UI for creating/editing a study protocol.*

The ‘Create new’ button will open a file explorer window. Navigate to where you’d like to save the protocol, give it a file name that can be easily associated with your study, and click ‘Save’.

The ‘Edit existing’ button will also open a file explorer window. Find the protocol you’d like to edit and click ‘Open’.

The filename of the protocol you are about to create/edit will appear in the text field underneath.

Edit the other parameters as appropriate for your study and click ‘Save and close’. See the sections below for more information about certain parameters.

The text fields and the channel checkboxes on the UI will then be populated with the parameters of the protocol you have just edited.

### Parameter choices

This section gives a bit more information about the sample rate and amplitude scaling factors, which should be considered when defining them in your study protocol.

#### Sample rate

This section outlines a couple of things to consider when choosing the sample rate, i.e., the number of data points which will be recorded per second.

Firstly, a sample rate of **at least** 2000Hz should be used when recording data with the OPMs. This is to avoid aliasing artefacts.

Secondly, the sample rates that can be set precisely are limited by the clock rate of the data acquisition hardware, which is 80MHz. A sample rate value can only be used if 80MHz divided by the value gives an integer. (E.g. for a sample rate of 2kHz, 80,000,000/2,000 = 40,000. This is an integer so a 2kHz sample rate can be used). The sample rates that can be set precisely (that are factors of 500 in the range 2000-5000) are: 2000, 2500, 4000, 5000Hz. It is possible to set sample rates of other frequencies, but keep in mind that the hardware will automatically round this to the nearest factor of 80MHz. For example, a sample rate of 3000Hz will be automatically changed to 3000.075Hz.

#### Amplitude scaling factors (gain)

The values next to each checkbox in the channels panel on the protocol UI are the amplitude scaling factors. The data acquired from each channel will be multiplied by the scale factor corresponding to that channel when displayed on the live data plot.

If a particular channel is known to be very noisy, and so has high amplitude compared to the other channels, then you may wish to reduce the scaling factor next to the corresponding channel checkbox so that this channel doesn’t dominate the data visualisation in the live plot. Conversely, you may want to increase the scaling factor if a channel is known to have low signal amplitude.

*Note: this has no effect on data recordings. The amplitude scaling factors only affect the amplitude of the data in the live plot, with the aim of ensuring that all channels can be clearly visualised.*

## Developers’ notes

This section is intended for anyone who wants to edit the MATLAB code for the data acquisition program. I’ll give an outline of what each of the scripts does so hopefully it’s easier to find the bits that you want to change/ add to.

Firstly, the scripts are all in a git repository, so while making adjustments I recommend taking advantage of that for version control. You can open a Linux style ‘terminal’ by clicking the Git Bash icon in the start menu, or use Git GUI if you’re not used to command line.

Also, in case it’s useful, the program was originally based on this example I found online. While it’s now pretty different, the fundamental format is mostly still there.

<https://uk.mathworks.com/help/daq/software-analog-triggered-data-capture.html>

The scripts in the DataAcquisition directory are as follows:

#### DataAcquisition.m

This script initiates and configures the data acquisition and then acts as a wrapper, calling the other scripts. It stops the data acquisition at the end.

#### CreateEditProtocol.m

This script creates the UI that appears before data acquisition if the user chooses to create or edit a protocol. It then saves the protocol parameters to file as a .mat file after the user is finished editing.

#### createDataCaptureUI.m

This script creates the UI for visualising and recording the data. It creates the figure, the axes, the buttons, the text fields, the checkboxes and all the text labels.

#### dataCapture.m

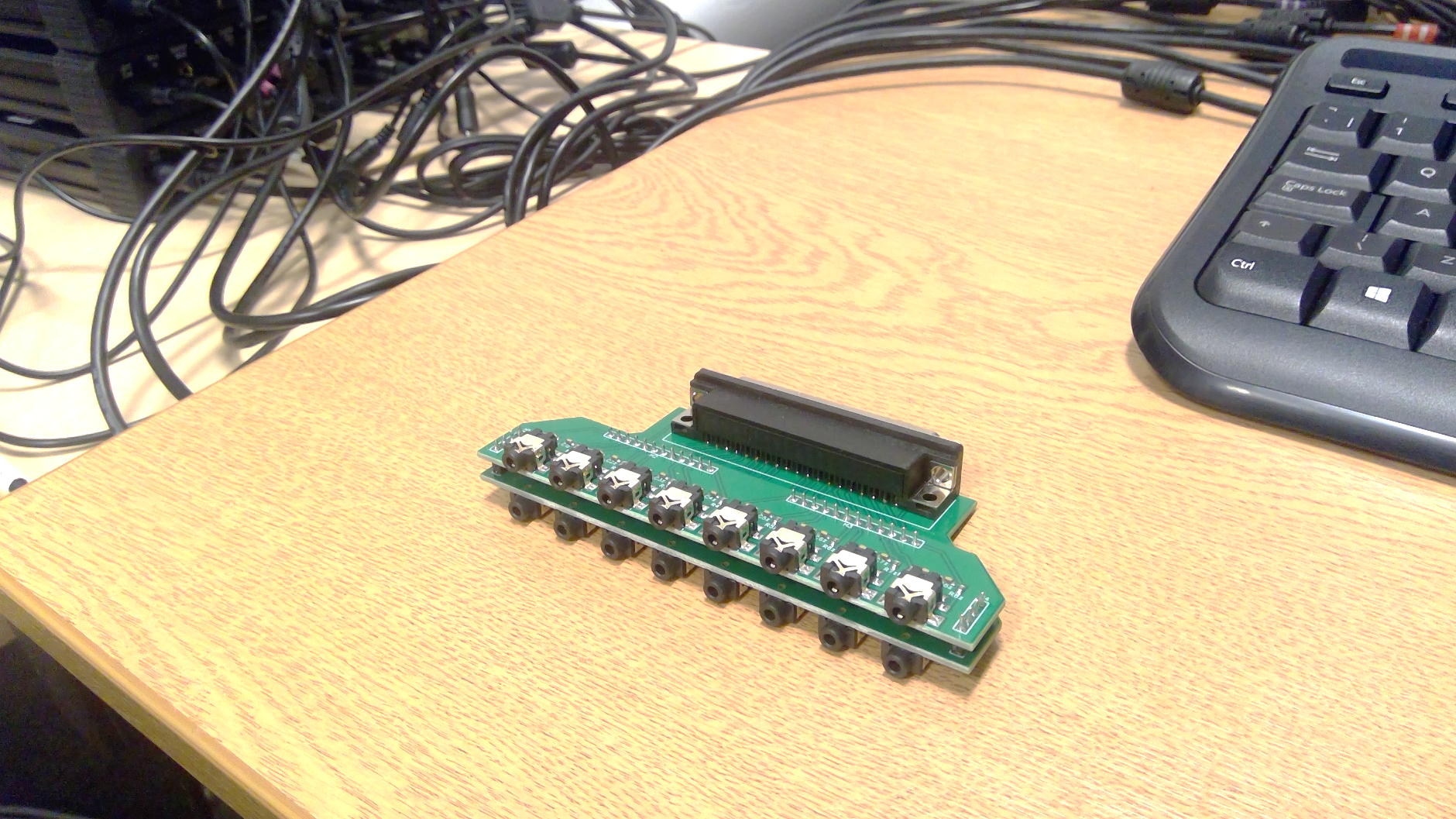
This script does most of the work. This is the one that reads, plots, records, and saves the data. It continuously runs during acquisition.

There are some comments at the top of the script with some information about how it runs. Each run through, it reads in a chunk of data (latestData). To get the data needed for the live plot, the latestData chunk is appended to the dataBuffer matrix, then any earlier data that is no longer needed is removed.

The stim channel data is received through 4 ports on the Quspin adaptor board (10,11,14,15) as a binary value, which is converted to a decimal value before plotting and recording. (4 ports = 8 channels, i.e. 8 digit binary number, i.e. 0-255 in decimal)

During recording, the captureData variable stores the recorded data, so the latestData chunk is also appended to this.

#### ChannelLabels.m



*Figure 3: Quspin adaptor board*

16

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

This script contains the cell array containing the channel labels. The order of the channels is determined by the order they are plugged into the Quspin adaptor board. Figure 3 shows the order that the channels should be entered in the cell array based on where they are plugged into the adaptor board. Edit this cell array if the OPM analogue output cables are ever rearranged.

*Note: only enter the OPM channel labels here. The stim channels are dealt with in dataCapture.m.*

#### ArrangePlotPositions.m

This script shifts the plots around depending on whether the stimulus channel and recorded data plots are being displayed. For example, if the recorded data plot is hidden, the stim channel plot will be moved down and the live data plot will be expanded.