Ref Step calibration software.

# Overview

The software allows for an automatic use of the ref-step algorithm for the calibration of a voltage source or meter (results are given for both). During the calibration, instructions are read from a table and sent to the instruments over a GPIB cable. live data is presented in a graph, and results are continuously saved into the same table as the instructions came from. Results can be analysed using the same program, and final linearity ratios are printed onto the data table.

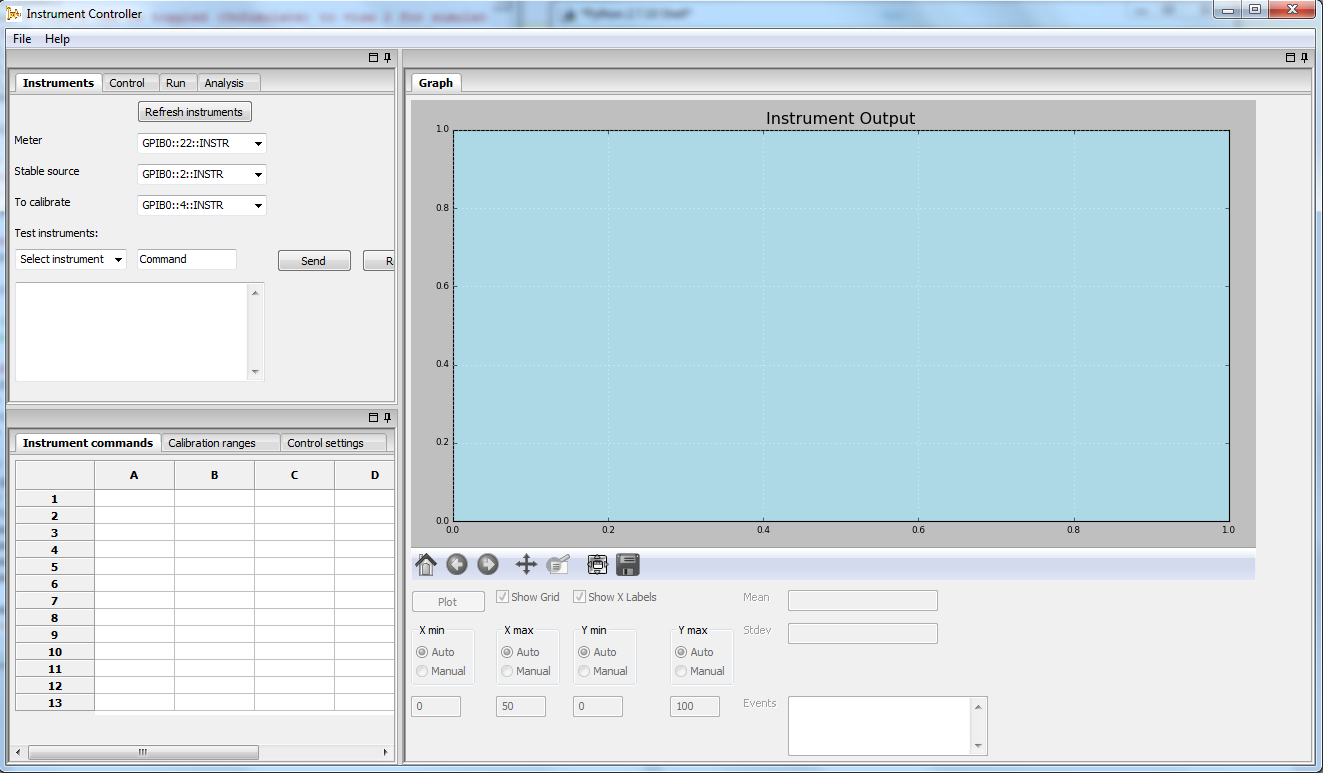
To work, the program needs to know how to communicate with the various instruments connected to it. For this it requires a ‘dictionary’ file to be provided, specifying necessary commands, such as the command for resetting the instrument. In principal, the commands can be anything, as long as they pass the safety checks of the program. The program may be given a customised set of voltages to calibrate with, or it will generate the necessary calibrations based on the instruments’ ranges that need to be specified in the dictionary file.

# Wiring up the instruments

Connect the low of both sources together. The high of source S connects to the low on the meter, and the high on source X connects to the high on the meter.

# The UI

The virtual instrument or user interface (UI) consists of three panels; a control panel, a panel for tables, and the graph panel.



1. The control area has two tabs, ‘Control’ used to control the algorithm, and ‘instruments’ which is used for testing the instruments and specifying which instruments to connect to over the GPIB. Here the GPIB address needs to be assigned to each instrument.
2. The tables window has three tabs for the dictionary file, ranges to be calibrated (optional) and the control settings detailing what is send to the instruments during the calibration.
3. The graph window gives a visual of the current data set being measured, and comes with the Matplotlib control bar. The the last 50 data points by default.

# Connecting to the instruments over the GPIB

In the instruments tab of the program, you will need to specify the GPIB address for each of the three instruments. The ‘Refresh instruments’ button on the top will refresh the addresses and present all currently recognised instrument addresses. You may test your instruments using the send and read buttons below. First select an instrument, write the command in the command box and click the send button. The read button will read whatever is on the buffer of the instrument. A query command is essentially a write followed by a read.

# Writing dictionary files

The next stage is to provide a way for the program to communicate with the instruments. For this a dictionary file is needed. The send and read buttons above can be used to test commands prior to putting them in the dictionary.

An example dictionary file is presented below, note that some commands are needed for both meters and sources, while others are specific only to one instrument type. The dictionary file needs to be loaded from the control tab, where an excel workbook is selected. The sheet containing the dictionary must be named ‘Dict’. If other sheets are present, the program looks for one labelled ‘Control’ and ‘Ranges’, if they are found they are also loaded, into their respective tables on the UI.

|  |  |  |
| --- | --- | --- |
| **Key word** | **Example** | **Description** |
| label | HP3458A-230 | The label to be printed in log files and on results table, this should be something that you can use to easily work out which instrument was used in the calibration. |
| NoError | 0\r\n | When the instrument’s error status is queried, what is the response for no error? This is used to determine if it is safe to run the program and check if it should stop in case of malfunctions. |
| reset | RESET | The reset command of your instrument |
| Init | END 2; DCV AUTO;NPLC 1 | The initialisation of the instrument, in this case (HP3458A) the instrument is required to be sent an end type for its command termination, and the voltage is set to DC and automatic, in case of an unexpected large voltage input, being on auto protects the instrument. NPLC sets the integration, 1 is short so that it quickly switches ranges if need be. |
| Make\_Safe | DCV AUTO | Safe state for the instruments, such as setting zero volts and outputs off. |
| Error | ERR? | Command to be sent to query the instrument for any errors. |
| SettleTime | 0.5 | After being sent a command, some instruments require time to implement it before they can be sent the next command. This is the specified time for the program to pause after a command is sent. The program will not wait this settle time when reading any instrument, it waits only after sending commands. |
| DCVRange | DCV $ | The program will read through the table of settings and will send this command to the instrument in order to change its DC volt range, replacing the $ sign with the label of the range (see Ranges). |
| MeasureSetup  (Meter only) | NPLC 20 | Prior to a burst of measurements, this is the set of instructions to be sent to the instrument. Here the integration is increased to ensure accuracy. |
| MsmntSetup (Meter only) | SKIP | Any command that needs to be sent to the instrument prior to saving a reading, like resetting the powerline measurement or turning display off if that interferes with speed. “SKIP” is a program command, when it reads “SKIP” it will bypass the sending command section altogether. Only applicable to this command. |
| Status | RANGE? | Command to ask for the instrument’s status, such as output values and ranges. This will later be recorded on the table of results. |
| operate (sources only) | OPER | Turning the outputs on, this will output whatever voltage has been sent in the settings. |
| Standby (sources only) | STBY | Standby, turning the outputs off. |
| SetVoltage | OUT $V | Voltage setting command, while keeping outputs off. |
| Ranges | (0,1,’1’),  (0,10,’10’) | List of ranges, each range goes in a set of round brackets. The elements are (lowest voltage, highest voltage, range label)  Range label is what needs to be sent to the instrument in DCVRange to set this range. |

NOTE: If quotation marks need to be sent to the instruments, they must be replaced with: \”. A slash and a DOUBLE quotation mark. If it is incorrectly entered, the program will run into an error. (? Check this again)

Termination characters must be added to the end of each command, both in the testing environment and in the dictionary unless they are the standard EOL commands. The dictionary file must have a column for the sequence of used key words (ones not in use can simply be excluded) followed by the column of commands for the instrument. This gives a total of six columns, two for each instrument. And the order for the instruments must be: meter, followed by source S and then source X. An example dictionary file is provided, named “sample.xls”.

# Range tables

Range tables are used to generate the control table. The range table specifies the lowest and highest points to be calibrated. It is possible that the instruments will not all be able to cover the specified range (eg some instruments can’t output 0V on their 1000V range) but this is handled in the program.

Also specified in the range table:

* The number of readings per point on the build-up, those are averaged and their standard deviation used in the analysis.
* Pre-reading delay is the delay to take before beginning this particular range, for example in order to let instruments warm up on ranges with high voltage outputs.
* Inter reading delay: If a delay is needed between each point on this range in the build-up, for example some instruments take longer to reach stability after being set to their high ranges.
* # repetitions: number of times to repeat this buildup (both positive and negative).
* # steps: How many steps can this range be split into, recommended is 10.

# Control table

The control table can be created based on the instrument ranges and the dictionary provided. To do this, press the ‘Generate table’ button in the ‘Control’ tab. The control table will contain instructions that are sent to the instruments line by line, and readings are printed back to the same table too.

Instruments not being calibrated will be set to their minimum range that encompasses all possible values that they will take, for example if for a certain sequence the meter will read a maximum of 10 volts, it is set to the 12 volts range throughout the sequence. Note that there is a tolerance of 0.5% taken, so if the maximum voltage is supposed to be set is 10v, the meter is set to a range that encompasses 10.05v.

The control table has fixed columns for the ranges of each instrument and the setting, and has columns for the status of each instrument. At the top of the table in row 4 are the instruments used. These titles are extracted from the dictionary files’ labels. In row 5 are the start and end points of the sequence, these can be edited before running the program for testing purposes. Both start and finish points must be specified. Once the program is started, editing these will not make a difference.

Warning flags are printed on the far-left column, warnings include a warning if a reading exceeds 0.1% of the range from the nominal readings and any errors that the instruments have unless they are the no-error response (specified in the dictionary).

## Inputting a file

Files can be loaded into the program using the ‘File>Open dictionary’ button. The tables loaded must be of the same structure as those created by the program, so it is recommended to upload tables that were created by the program in the first place. For this purpose, you can also save a table once it is created ‘File>Save tables’. The table must be an excel .xls workbook.

# Analysis

Once data is collected, the ‘Analyse data’ button will load the raw file specified in the analysis file name (program automatically fills it in once it finishes a run).

The file is sent to the analysis object, which does the following:

* Reads and saves the entire file into arrays of its own
* Identifies start and finish points of each sequence, by identifying the mid points. So, sequences are not restricted to 10 steps and not all sequences must have the same number of steps.
* Computes linearity and gain ratios for the ascending and descending sections for both the meter and the source X, using the same data, for each segment of data.

Table Sample results table, m stands for meter, x for source x. Each instrument has two ratios, one for ascending one for decending.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ratio | STDEV | Effct. DoF |
| x linearity | 1.000007 | 1.2E-05 | 40.20919 |
| x gain | 1.000005 | 1.19E-05 | 41.37655 |
| x linearity | 1.000001 | 1.21E-05 | 46.07434 |
| x gain | 1.00001 | 1.69E-05 | 31.12255 |
| m linearity | 1.000008 | 1.14E-05 | 36.16598 |
| m gain | 0.999999 | 1.09E-05 | 34.83735 |
| m linearity | 1.000001 | 1.15E-05 | 46.54867 |
| m gain | 1.000002 | 1.08E-05 | 49.94697 |

Final ratios are printed to the same excel file, on a new sheet. The linearity ratio is the ratio of the top of the range to the bottom of the range, for example 10v to 1v both in the same range. Gain ratio is the ratio between the top of one range to the top of another, with the previous example that would be 10v on one range to 1v on the range below.

# Log files, Raw data and saving.

A log file containing all communications between the program and the instruments is saved. This provides a way of checking what the program did and where it went wrong (if it did).

Raw data files are also produced, containing everything in the control table as well as a list of the actual individual readings taken in each row. The analysis tool makes use of the raw data file saved, meaning that the program can be used to analyse existing data files from other calibrations.

These files are automatically created and saved by the program, which names them with a time stamp of the time the algorithm started.

When saving a table, extra data can be put into rows above row 3, it will remain there in the raw files and when a control table is saved, this can help with identifying and keeping track of calibrations.

# Running the program

Once a table is loaded and instruments are correctly connected, the ‘start’ button in the ‘Run’ tab will set the program running. The stop button simply stops the program where it is, and if the start button is pressed again it will start over. It is not recommended to start a sequence part way through. ‘Make safe’ will send all instruments the command specified for making them safe to approach. This, and all other commands, should not be relied on as they are based off of the dictionary given, and cannot guarantee that the instruments are safe. The program considers the situation safe if it successfully sent the user-specified make safe command string, but it cannot guarantee that the situation is indeed safe.

If the program encounters a reading that is more than 5% of its range from the nominal reading it will attempt to make safe, and abort.

If it is unable to communicate with the instruments, it will also consider the situation dangerous and attempt to make safe and abort.

If the program did not successfully send the make safe commands, the make safe button turns red and it aborts with a message saying with which instrument it failed to communicate.

# Diagnosis of faulty results

The graph in the graph panel can be used to get a visual of the data in real time, if results are critically out of line and do not follow the ref step pattern this will be easily visible. If a data point has an unexpectedly large standard deviation, the graph can be paused in order to zoom in on the section in question (the matplotlib graph tool bar is below the graph). The raw logs can also be used to determine if this was an accidental surge in voltage, or if perhaps the settling time needs to be increased if data points are clearly drifting. The table (and therefore logs) will contain warnings when a measurement set has exceeded 0.1% of the range from the nominal reading. If a voltage reading exceeds 5% of the range from the nominal reading, the program deems this unsafe and attempts to run the make safe routine prior to aborting.