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. 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 two tabs, one showing the control settings of the algorithm, and another showing the dictionaries that will be used to communicate with the instruments.
3. The graph window gives a visual of the current data set being measured, and comes with the Matplotlib control bar. The graph windows in on the last 50 data points by default, but manual window parameters can be chosen.

# 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 GPIB address is the address assigned to the instrument over the GPIB interface(?). 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. To query an instrument for its status, first send the query command, then press read. This works because a query is essentially a send followed by an immediate read.

# Writing dictionary files

An example dictionary file is presented below, 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 needs to be specified. The sheet containing the dictionary must be named ‘Dict’. The load function will ignore all other sheets except ‘Sheet 1’ which it will take as the control settings, this can be used to upload a control file if a custom one is needed.

|  |  |  |
| --- | --- | --- |
| **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 name of the range. |
| 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’,10,3),  (0,10,’10’,10,5) | List of ranges, each range goes in a set of round brackets. The elements are (lowest voltage, highest voltage, range label, number of readings to be taken, wait time in seconds, measurement pause time in seconds [Meter only].)  Range label is what needs to be sent to the instrument in DCVRange to set this range.  Wait time is the time the instrument requires to settle in this range.  Measurement pause time is the time to wait between measurements. This is specific to a range, and is used in case in a particular range the integration will be long, in which case a pause is needed between readings or else the PyVisa returns a time-out error. This might also be necessary for instruments that might return the same value if read several times before it takes another reading. |

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.

In the instruments panel, each command can be tested. One of the three instruments needs to be selected, and commands can be sent to the specified instrument from the ‘command’ text box using the ‘Send’ button. The ‘Read’ button will read whatever is on the instruments buffer, and print results in the text box below.

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.

# Control table

The control table can be created based on the instrument ranges provided. To do this, press the ‘Generate a table’ button in the ‘Control’ tab. This can only be done once the dictionary was uploaded, since it contains the dictionary files. If the ‘M Meter’ radio button is selected, the table will be created according to the ranges of the meter. If the ‘X Source’ button is selected the table is created according to the ranges of source X. The control table created will generate several sequences of voltage settings and range settings needed for all three instruments to step through and calibrate the specified instrument. Each ‘sequence’ is a stepwise ascent to the maximum of the range followed by an opposite descent.

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.

## What is in a control table?

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 if only a segment of the table needs to be analysed. Both start and finish points must be specified. Once the program is started, editing these will not make a difference.

The number of readings and specified wait time are both the max of the three instruments’ requirements for the ranges on which they will operate. This ensures that if one instrument takes longer than the others to settle, all instruments wait sufficiently long for this to happen.

Results are saved in a column to the right, they consist of the mean of however many measurements were specified, and a standard deviation.

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 ‘input file’ and ‘load table’ buttons. 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. The table must be an excel .xls workbook, with one sheet named ‘Sheet 1’ for the control settings. A dictionary sheet named ‘Dict’ must be included for the program to communicate with the instruments. The control table does not need to be initially specified as it can be generated.

# 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.
* 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 raw data is used by the analysis tool, not the table. So an old set of data can be analysed again.

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

Once done with a table, it can be saved using the ‘Output File’ button, if it were edited at all. It will be saved under the name specified or given a time stamp. All rows and columns above the instruments row (3) can be used to record info about the measurements. If info is written in there it will be shown again in the results and raw data files, which will help with tracking calibrations.

# Running the program

Once a table is loaded and instruments are correctly connected, the ‘start’ button 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 the dictionary given, and can be faulty. The program considers the situation safe if it successfully sent the user-specified make safe command string.

When the program is running, the buttons are still responsive, but at times it may take a period of time for the commands to reach the instruments if the program is still waiting on a command wait period specified in the dictionary file.

The waits in the table that are used to allow the instruments to settle do not induce any freezing time on the algorithm; While waiting for instruments to settle, the program is looping through a check to see if it should abort until the specified time is over, so it can be aborted at any time.

The program highlights the row from which it is operating, and disables buttons that should not be pressed during operation. The table should also not be edited during operation, although if things in the control sequence are edited prior to the table reading them the updates will be considered. The program will not change its stop and start locations once it starts running.

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 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.