Maximum Power Point Tracking MATLAB-Script

2022

Henri Salonen

May 1, 2022

Introduction

The Maximum Power Point Tracking Script is a MATLAB script made for characterizing Perovskite solar cells. The main use of the script is to make MPP tracking measurements for cells being studied in conjunction with a solar simulator and an SMU that communicates via SCPI commands. The script also measures I-V curves and provides figures of merit from them. It can be used to perform a forward and reverse polarity voltage sweep if needed.

Contents

1 Requirements and Quick-start List **3**

2 Script **4**

2.1 Requirements 4

2.1.1 MATLAB 4

2.1.2 NI-MAX 4

2.1.3 Power Management 4

2.2 Script Functions 5

2.2.1 IV Curve Measurement 5

2.2.2 Maximum Power Point Tracking 5

2.3 Script Options 7

3 Troubleshooting **8**

1 Requirements and Quick-start List

The MPP-tracking script requires MATLAB version 2010a or newer. Latest tested version is 2021a. NI-MAX is also needed for communication with the SMU. The software is designed to be used with a Keithley 2450 SMU using standard SCPI communications.

1. Initialize the solar simulator according to the instructions.
2. Verify sample position with the reference cell.
3. Connect the SMU via USB to the measuring computer. Power on the SMU and connect the sample.
4. Copy the script to the folder in which you wish the data to end up in.
5. Ensure the options in the script file are as required from the measurement. See Table 1 for more detail.
6. Run the script from the folder. The syntax is `mppt(name, u_start, u_stop, a_sample)`, where name is the base for the filenames, `u_start` and `u_stop` are the beginning and end voltages respectfully and `a_sample` is the area of the sample in mm^2.
7. The SMU runs one or two IV-sweeps with the parameters given according to options.

2 Script

2.1 Requirements

2.1.1 MATLAB

The script is a MATLAB script and therefore requires MATLAB to run it. It has been tested with version 2021a, but should function with MATLAB 2010a onward. There is a concern that MATLAB will change its VISA coding, which may at some point make this script incompatible with newer MATLAB versions.

2.1.2 NI-MAX

The script uses the National Instruments Measurement and Automation Explorer (NI-MAX) for communication between the computer and the SMU. Ensure that NI-MAX is installed on your own computer if using it for the measurements. Searching for NI-MAX will result in finding the downloads. At the time of writing the instructions for downloading the software are found at Ni.com.

2.1.3 Power Management

The USB communication with the SMU has a problematic tendency to get disconnected by power saving options. For Windows users, go to Settings, 'Power & sleep' and select 'Additional power settings' from the right of the window. Then select 'Change plan settings' from the window that popped up. Select 'Never' from the drop down menu for sleep mode when the computer is plugged in. Also select 'Change advanced power settings' and from 'USB settings' select 'USB selective suspend setting' to be 'Disabled' when plugged in. Click 'OK' to confirm these selections and 'Save settings'.

2.2 Script Functions

2.2.1 IV Curve Measurement

The script measures IV curves using the voltage sweep function from the SMU. Depending on the selection of measurement type, the script will measure at least one IV sweep as detailed. Voltage step, rate and preliminary hold time are configurable. See Table 1 for more details on settings.

1. The SMU sets the voltage to the starting voltage for the set amount of time.
2. The SMU starts the voltage sweep, stepping from starting voltage to ending voltage at constant steps at a constant rate.
3. After the sweep is complete, MATLAB queries the data and plots it.
4. (Optional) Another sweep is performed as in 1.-3., but with the start and ending voltages reversed.
5. Figures of merit are calculated from the first sweep and presented.
6. (Optional) If a maximum power point tracking measurement is about to be performed, the user is queried on whether they wish to proceed.

2.2.2 Maximum Power Point Tracking

Maximum power point tracking is implemented as a perturb and observe algorithm utilizing averaging and voltage dwell time to determine maximum power point direction. Options for starting from a user set voltage or to estimate maximum power point from the previously measured I-V curve. Additional options for voltage steps and voltage change intervals are available. See Table 1 for more details.

Figure 1 depicts how the algorithm finds and tracks the maximum power point. If a user defined starting voltage (by default $0\,\mathrm{V}$) is used the algorithm starts with a larger voltage step to reach maximum power point faster. After finding the peak power it will change to using a smaller step for finer control. Between steps the algorithm averages the last half of the measurements of the interval taken to minimize the effect of any hysteresis present as well as mitigating the effects of noise in the signal.

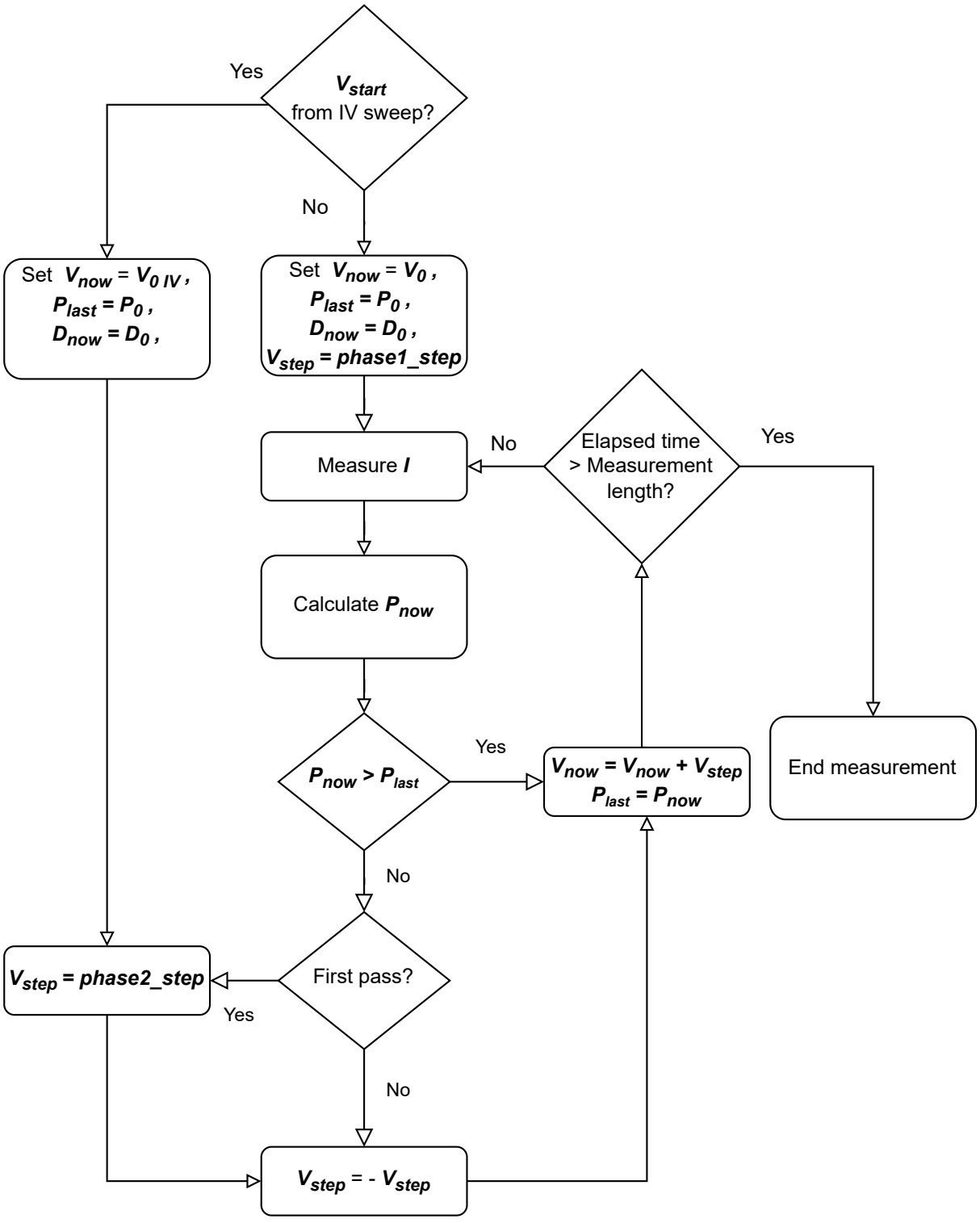

Figure 1: A flowchart of the operation of the algorithm. Note that the equals signs are not mathematical operators, but instead assignments of variables in the program.

2.3 Script Options

The script has a number of different options to use for measurements. Table 1 details the parameters, options and their selection syntax, as well as default values. Parameters are the values given when calling the function, while the options are found and changed in the `User Settings` portion of the script file.

Parameter	Def.Value	Description
`Name`	`'s1e1'`	Base of filename $string$
`u_start`	1.2	Start voltage for sweep (V)
`u_stop`	-0.2	End voltage for IV sweep (V)
`a_sample`	20	Area of the sample (mm^2)
Setting		
`meas_type`	1, 2 or 3	1: Measure only both IV-sweeps 2: Measure one IV-sweep and MPPT 3: Measure both IV-sweeps and MPPT
`meas_time`	`[h, m, s]`	Measurement length Hours (`h`), minutes (`m`) and seconds (`s`)
SMU Parameter	Value/Syntax	

`current_range`	`' 100E-6'`
	`' 10E-3'`
	`' 100E-3'`
		Max value of current expected (A)
		`''` and whitespace mandatory

`voltage_range`	`' 1'`
	`' 2'`
	`' 20'`
		Max value of voltage expected (V)
		`''` and whitespace mandatory

`visa_addr`	Check `NI-MAX`	USB address of SMU
IV Sweep Parameters	Default Value	Meaning
`IV_step`	0.01	Volts per step (V)
`IV_rate`	0.2	Delay per step (s)
`pre_sweep_pause`	3	Starting voltage hold before sweep. (s)
MPP Tracking Parameters	Default Value	Meaning
`phase1_step`	0.1	Initial tracking voltage step (V)
`phase2_step`	0.01	Voltage step after MPP is found (V)
`v_interval`	2	Interval of voltage steps (s)
`v_start`	0	Initial voltage of MPP tracking (V)
`p_lamp`	0.1	Lamp power (W/cm^2)
`m_start_dir_up`	$true$	Start direction, True: up, False: down
`v_from_sweep`	$true$	Start voltage from IV curve MPP

Table 1: List of parameters

3 Troubleshooting

There are a number of possible issues that have been discovered during testing.

1. Issue: The measurement has stopped with an error message about VISA connection.

This most likely is due to power management settings. Before you measure, make sure the power management settings are in order (See: 2.1.3). The error will also have set the SMU into an unresponsive state from USB commands, so there is a need to reboot the SMU. The script should run through saving procedures in the case of a communication error, but as an additional MATLAB will have saved the measurements into the workspace, should the data need to be manually saved.

2. Issue: The script measures an I-V curve, but fails when starting MPPT.

This is most likely a MATLAB fault. Restarting MATLAB will probably fix the problem. If not, a full restart of the computer should. Restarting the SMU may also be required.

3. Issue: The graph has stopped updating, but the measurement is not over.