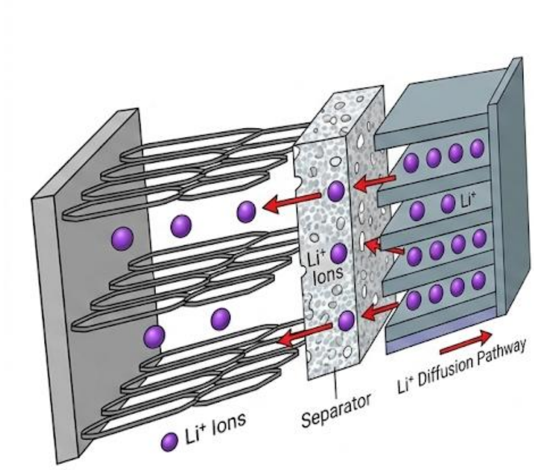
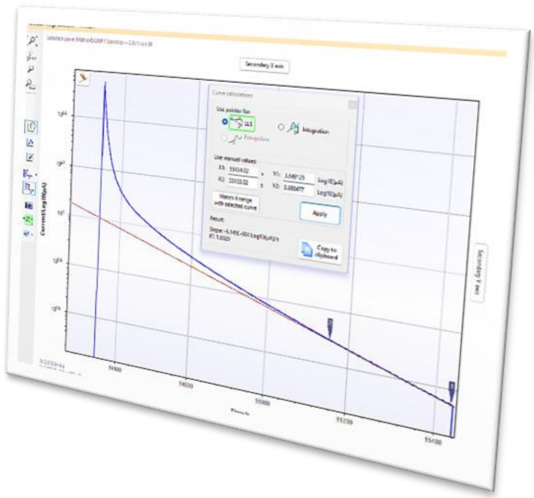
Potentiostatic Intermittent Titration Technique (PITT)





Application Note PSAN0423

© 2026 PalmSens BV

www.palmsens.com

Contents

1 Introduction

1.1 Concepts

PITT stands for Potentiostatic Intermittent Titration Technique. It consists of a single charge–discharge experiment performed using a sequence of small potential steps. Each potential pulse is followed by an open-circuit relaxation period, during which the current decays toward zero and the system approaches equilibrium. The potential steps are incremented (during charging) until the defined upper cutoff potential is reached and then decremented in the same manner during the discharge sequence until the lower cutoff potential is reached.

During each relaxation step, the current gradually decays. The transient current response following each potential step contains information related to the transport properties of electroactive species. In the original formulation of PITT, analytical expressions were derived linking the current transient after a small potential perturbation to the chemical diffusion coefficient in mixed ionic–electronic conductors. Although originally developed for lithium–metal alloy systems, PITT is nowadays most commonly applied to determine diffusion coefficients in battery electrode materials, especially lithium-ion systems.

1.2 Goal

The primary objective of this application note is to provide a practical, step-by-step guide for configuring and executing the Potentiostatic Intermittent Titration Technique (PITT) using PalmSens instrumentation and software.

While this document guides the procedure for the measurement, the theoretical derivation of PITT relies on specific boundary conditions (such as diffusion models, phase transitions, and electrode geometries). Because these variables are highly dependent on the specific chemistry and design of the system, the user must possess a thorough understanding of their own system's characteristics to accurately interpret the resulting data. For a deeper dive into the mathematical models and theoretical foundations of PITT please consult relevant electrochemistry and energy storage literature.

DISCLAIMER

PITT is not an instant analysis tool. It is an advanced, equilibrium-based characterization method that requires system-specific knowledge to yield meaningful kinetic insights.

This application note serves as a practical tutorial for running PITT using pre-developed MethodSCRIPT code. Although typical PITT experiment can be implemented using the native tools of PSTrace, this application note focuses on the use of MethodSCRIPT™ to enable automation, increased flexibility, and advanced data processing such as automatic linear fitting. MethodSCRIPT™ is the scripting language of our latest generation of instruments and modules. It was initially developed to facilitate custom software and OEM integration. In addition, by enabling full control over the instrument, MethodSCRIPT is a powerful tool for implementing specialized electrochemical techniques.

This application note is a step-by-step guide, and it is suitable for users without programming experience. If you are looking for detailed information about MethodSCRIPT check this page, our developer portal and the MethodSCRIPT documentation.

1.3 Compatible Devices

The following potentiostat models support the MethodSCRIPT code for PITT experiments:

- Nexus
- EmStat4 series* (EmStat4S, EmStat4X, MultiEmStat4)

*We recommend the EmStat4 HR version. The LR version may work in some cases, but it does not support the voltage used in this example (its range is ±3 V). The EmStat Pico and Sensit Series (Sensit Smart, Sensit BT, Sensit Wearable) support MethodSCRIPT, but their limited current and voltage aren't suitable for most battery experiments.

1.4 Compatible Software

This application note was written with MethodSCRIPT version 1.9 and PSTrace version 5.13. Using higher versions of MethodSCRIPT could require some changes due to commands that are deprecated code in that version. If your device doesn't support MethodSCRIPT, skip to section 3 to see alternatives. MultiTrace supports MethodSCRIPT too, although we recommend PSTrace to edit the code.

2 Setting up and Running the Experiment

2.1 Enabling MethodSCRIPT Editor

In PSTrace, the list of techniques includes one called '**MethodSCRIPT Sandbox**' which allows you to generate and run scripts. Make sure that this feature is enabled. Check the box 'Show MethodSCRIPT Editor' in the top menu *Tools > General Settings...* (see Figure 1). The box is not checked by default.

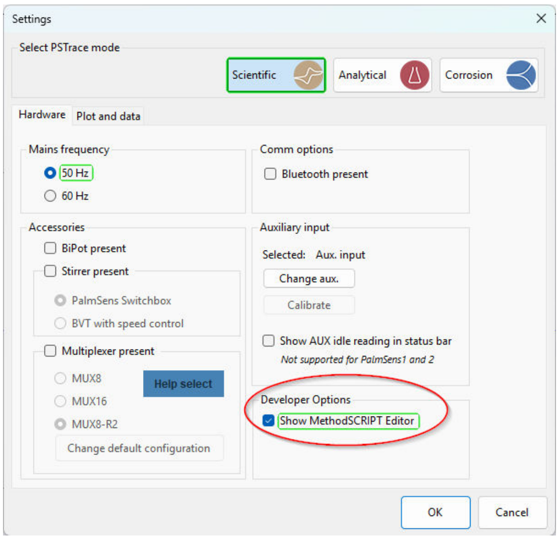


Figure 1 MethodSCRIPT option checked in General Settings

2.2 Load the Example

Go to the 'Method' menu, select 'Load' and choose the 'PITT with automatic slope' method. It can be found in the PSData > Battery testing examples folder. In a default installation, PSData is located in the Documents folder. When a MethodSCRIPT is loaded, the MethodSCRIPT Sandbox is chosen as technique. The advanced options under the ellipsis button (•••) display options for customized units (see.Figure 2) Here you can assign the variable type "AS" a custom unit and symbol for plots in PSTrace. This setting is recorded in the method file and it will be loaded automatically. The slope-time plot uses a custom symbol (α) and unit (Ln(A)/s) for the slope. This symbol and unit will be associated with the variable type AS. Customized units appear as custom axes labels in the resulting plot.

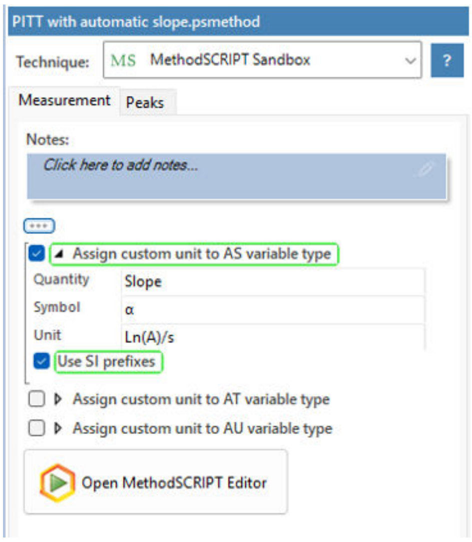


Figure 2 PITT method example loaded in PSTrace Method Editor

Alternatively, you can copy a text with the referred MethodSCRIPT code into the editor of the Sandbox. However, this way the unit customization will not be loaded. You can customize the axes manually by filling in the 'Assign custom unit…' fields. If you don't set this, the slope will be read as a potential (voltage) value, and it will be overlaid with the OCP (Open Circuit Potentiometry) plots.

2.3 Explore and Edit the Parameters of the PITT

Click on 'Open MethodSCRIPT Editor' and to see the code. Scroll down to lines 44-52. These lines contain all the parameters that need to be set for this cycling experiment. See **Error! Reference source not found.**. Change the purple numbers to your required parameters.

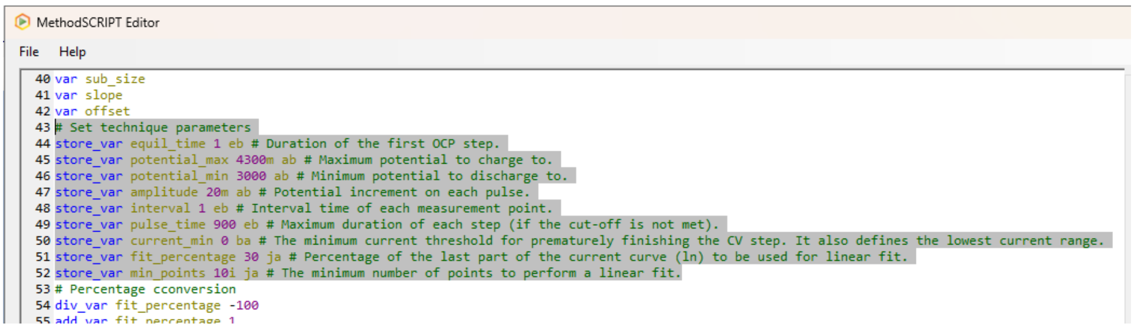


Figure 3 MethodSCRIPT lines referring to the main experiment parameters

The code contains many comments (green) explaining the purpose of each code. Note that the MethodSCRIPT editor shows different color for text according to different functions:

- Dark yellow for variables
- Blue for codes that perform an action, create or adjust a setting, or perform a logic or math operation. Example: var xx declares the variable xx
- Violet for numbers in general. It must be associated with an associated value, otherwise an error message will pop-up. When the value must be an integer, it is followed by an 'i'. Also, it can be followed by a metric SI suffix, i.e. 4.3 is represented as 4300 and a prefix m, therefore 4300m.
- Green for comments. Comments can be added to a line by inserting a # character followed by the comment. Comments do not affect the execution of the script.
- Red for string variables used for communication messages.

In its simplest form, an experiment in MethodSCRIPT consists of a *measurement loop* (meas_loop_xxx). For example, an OCP experiment with 1 s interval and 15 minutes (900 seconds) duration can be defined with the command: meas_loop_ocp e 1 901
In this command, the measured potential is stored in the variable e, which can later be plotted using pck commands.
In this example, however, variables were used instead of fixed numerical values to define the measurement loop parameters. These variables are declared at the beginning of the script using the store_var command. This approach makes the script easier to modify, as the same variable can be reused in multiple measurement loops. It also improves readability by grouping all key parameters in one place.

2.4 Run the Experiment

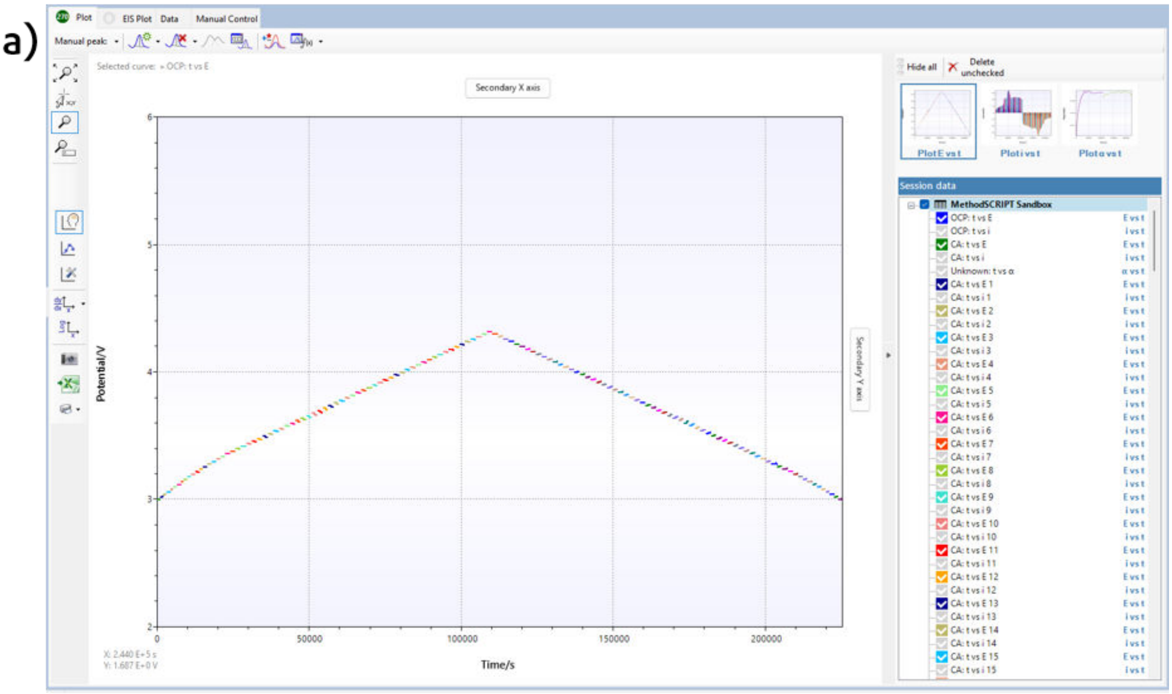
HINT

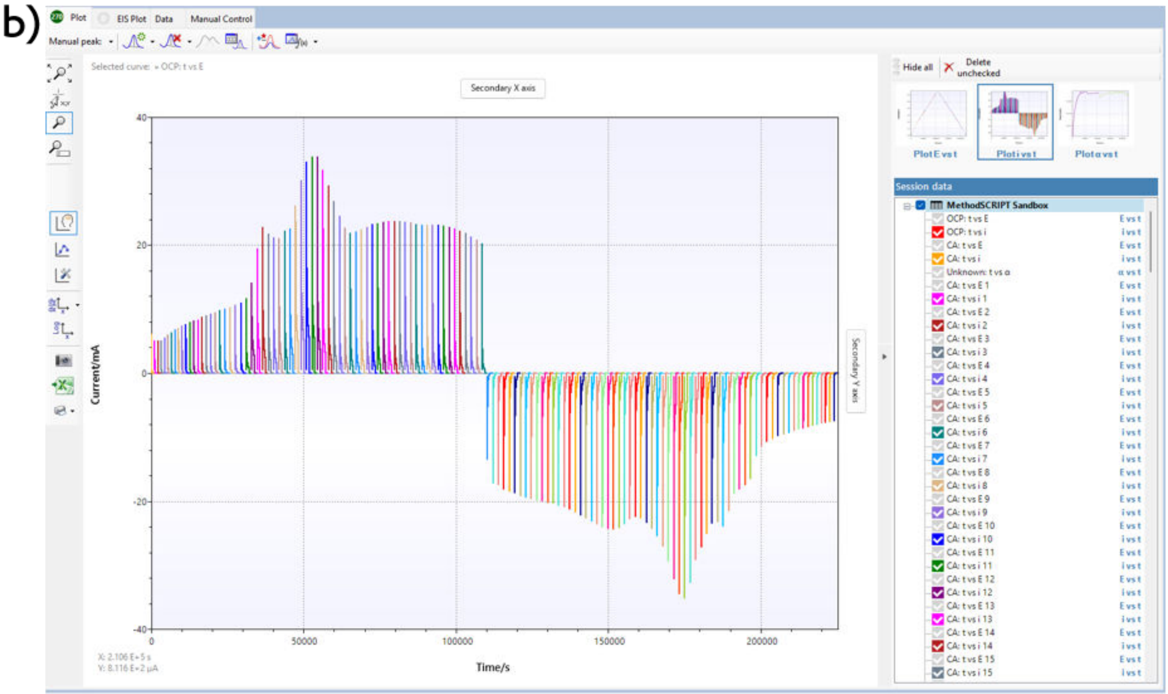
Before the full experiment, run a short preliminary experiment to verify the setup. For example, reduce the step times and connect a dummy cell. This allows you to quickly confirm that the response behaves as expected before running the full experiment.

After setting the parameters, simply close the MethodSCRIPT Editor window and click Run. This experiment will generate 3 plots:

1. Voltage versus time as 'Plot E vs t'
2. Current versus time as 'Plot i vs t'
3. Slope versus time as 'α vs t'

See result examples obtained with a real Li-ion battery on Figure 4.


a)
Secondary X axis
Potential/V
Time/s
Session data
MethodSCRIPT Sandbox


b)
Secondary X axis
Current/mA
Time/s
Session data
MethodSCRIPT Sandbox

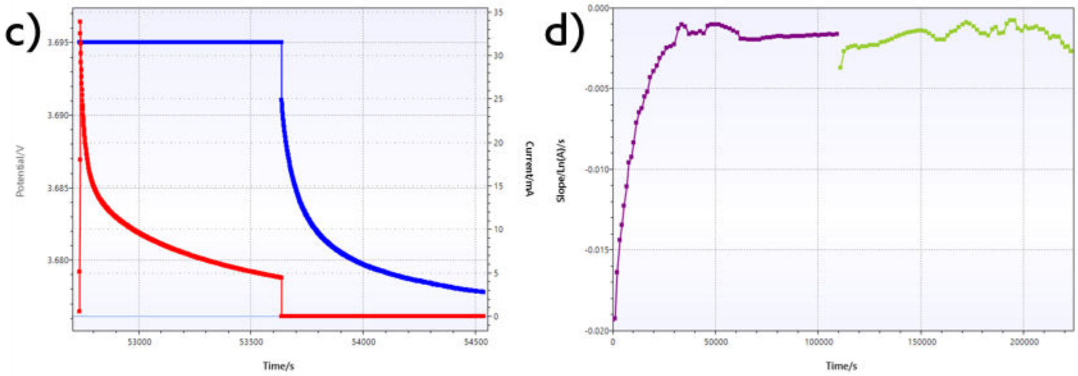


Figure 4 Plot example of a PITT experiment in a real Li-ion battery. a) full plot of Potential versus Time; b) full plot of Current versus Time; c) zoomed-in single potential pulse (potential in blue, current in red); d) automatic slope results (charge step in purple, discharge step in green)

2.5 Further Customization

If you want to run an experiment slightly differently, you can start with the original MethodCRIPT code and adjust it accordingly. For detailed information about available commands and syntax, refer to the MethodSCRIPT manual. Here are some tips for developing MethodSCRIPT smoothly:

- Use a dummy cell and adjust the parameters to have a full run in a few seconds.
- When an invalid script is executed, an error message is usually displayed including the line that produces the error.
- The MethodSCRIPT manual contains a list of error messages along with explanations of their causes. However, some configuration issues may not generate an error message but can still lead to unexpected behavior.

Pay special attention to the topics below.

2.5.1 Potentiostat Settings

The settings listed below are automatically reset to default values whenever the potentiostat mode is changed using the set_pgstat_mode command:

- set_max_bandwidth
- set_acquisition_frac_autoadjust
- set_range_minmax
- set_autoranging
- set_range

Therefore, you must declare them again every time you change the potentiostat mode. If these parameters are not explicitly set after a set_pgstat_mode command, they will revert to factory values instead of the ones declared before. For example, the selected current range may be set to 100 µA (factory default for Nexus), rather than the previous values declared with set_range.

2.5.2 Bandwidth

PSTrace automatically calculates an appropriate bandwidth based on the selected technique, its parameters and the connected device. In this example script, we calculate the bandwidth using an equivalent formula used by PSTrace. Some techniques may require a different calculation.
If you are unsure which bandwidth to use, you can generate a MethodSCRIPT code from a PSTrace method using the same technique parameters. This provides a reliable starting point. In some cases,

reducing the bandwidth can help stabilize measurements with noisy or unstable samples. However, excessively low bandwidth may result in inaccurate data.

2.5.3 Potential and Current Ranges

There are 3 different MethodSCRIPT commands associated with potential and current ranges.

- set_range_minmax
- set_autoranging
- set_range

Remember to set them accordingly, as a bad setting may lead to bad data collection due to overload or underload. If no range is selected, the potentiostat may select a default value or apply a previous setting. Refer to the MethodSCRIPT manual to see more details about the ranges syntax.

2.5.4 Concatenating Plots

In this example, the set_script_output command is used to concatenate the Constant Voltage and Open Circuit Potentiometry steps into a single continuous plot. Without this command, each step would be displayed as a separate dataset, typically shown in different colors.
This behavior is related to how MethodSCRIPT transmits data and how PSTrace displays it. To concatenate datasets, data transmission is temporarily disabled using set_script_output 0 before the endloop command. It is then re-enabled with set_script_output 1 before starting the next dataset. This ensures that the data appears as a single continuous plot.

3 Extra: PITT and GITT without MethodSCRIPT

If you have a device that doesn't support MethodSCRIPT, you can still run PITT experiments by using PSTrace native tools (see chapter 3.1).
GITT (Galvanostatic Intermittent Titration Technique) is simpler than PITT and can be configured using MultiStep Amperometry (see chapter 3.2).

3.1 PITT using Script

There are some disadvantages in using the Script menu instead of MethodSCRIPT:

- Time data will be reset on each step, so you will not see the results as a continuous timeline;
- If you want to use Script for automatically saving data, each step will be saved as a separate file. It is possible to save the full experiment as a single file, but you need to wait for the end of experiment and make sure not to clear the plot area;
- There is no possibility for automating the linear fit / slope determination.

3.1.1 Create the Chronoamperometry and OCP Methods

In the Method Editor, select Chronoamperometry from the Technique dropdown menu and configure the parameters required for your constant voltage step:

a) Ensure you select the appropriate current and potential ranges for your specific samples.

b) Click the ellipsis button (...) to open advanced options and check 'Stop when I <' box if you want to set a current cut-off limit. Set the value of current cut-off to prematurely finish the pulse.

c) Save this method by clicking the Save button or navigating to Method > Save as.

d) Create a second method for the Open Circuit Potential (OCP) step. Select 'Open Circuit Potentiometry' from the Technique field, set your desired interval and run times, and save this method as well (see Figure 5 for an example).

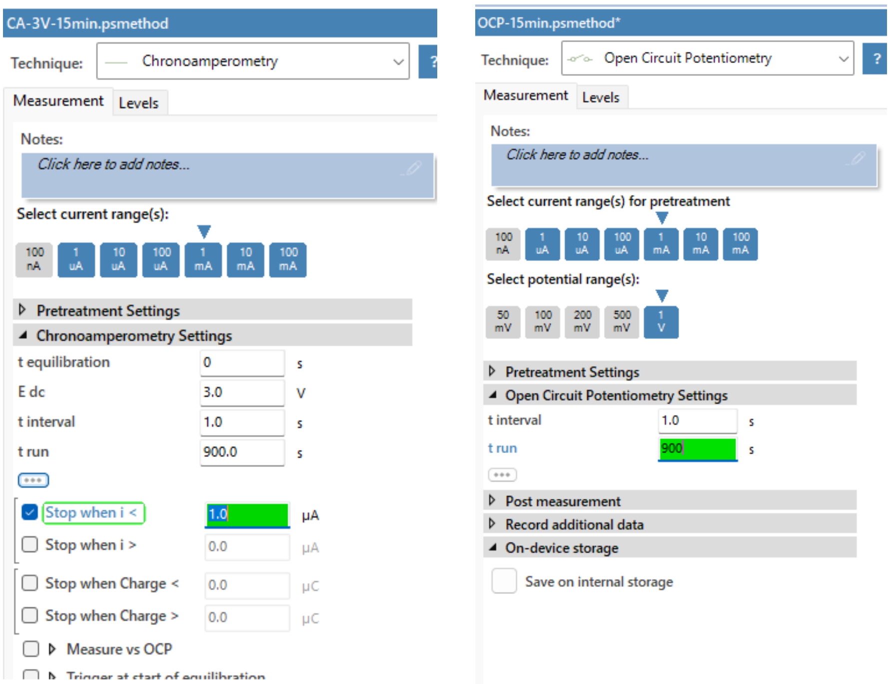


Figure 5 Creating methods to be used as PITT steps in a Script sequence

3.1.2 Set Up the Script Loop

a) Open the Script Editor by navigating to Script > Open script window...

b) Click on the Measurement block and uncheck 'Clear plot before start measurement'. Select your saved OCP method if you want to begin the experiment with an initial equilibration OCP. Configure the Save result field as required for your workflow.

c) Click Repeat to create a loop. Set the number of repetitions to match your total number of pulses for the charging step. Example: If you want to start at 3.0 V and charge up to 4.1 V in 20 mV steps, the calculation is:

$$\frac{(4.1\ \text{V} - 3.0\ \text{V})}{0.02\ \text{V}} = 55\ \text{steps}$$

Since the first iteration is the zeroth, you should add one more, so **56** in the repetition field. While the Repeat block is selected, click Measurement twice to add both your Chronoamperometry and OCP methods inside the loop. If necessary, you can click and drag the blocks to position them correctly inside the loop.

d) Ensure that 'Clear plot before start measurement' is unchecked for all blocks inside the loop. This guarantees that data from all iterations will overlay on the same graph.

3.1.3 Configure the Potential Increments

a) Select the Measurement block containing the Chronoamperometry step.

b) Go to the Advanced tab and select the Override block. This action places an override command directly beneath the selected measurement block.

c) In the override settings, configure the following:
   - Parameter: **E dc**
   - New value: Enter your desired amplitude (e.g., **0.02 V**)
   - Select '**New value is added**'
   - Check the '**Skip first iteration**' box.

To create the discharge sequence, follow the exact same steps outlined for the charge step with two specific adjustments:

i. Set the amplitude as a negative value (e.g., -0.02 V).

ii. Leave the 'Skip first iteration box' unchecked. If checked, the software would run the upper potential step twice

See Figure 6 for an example of a complete PITT Script.

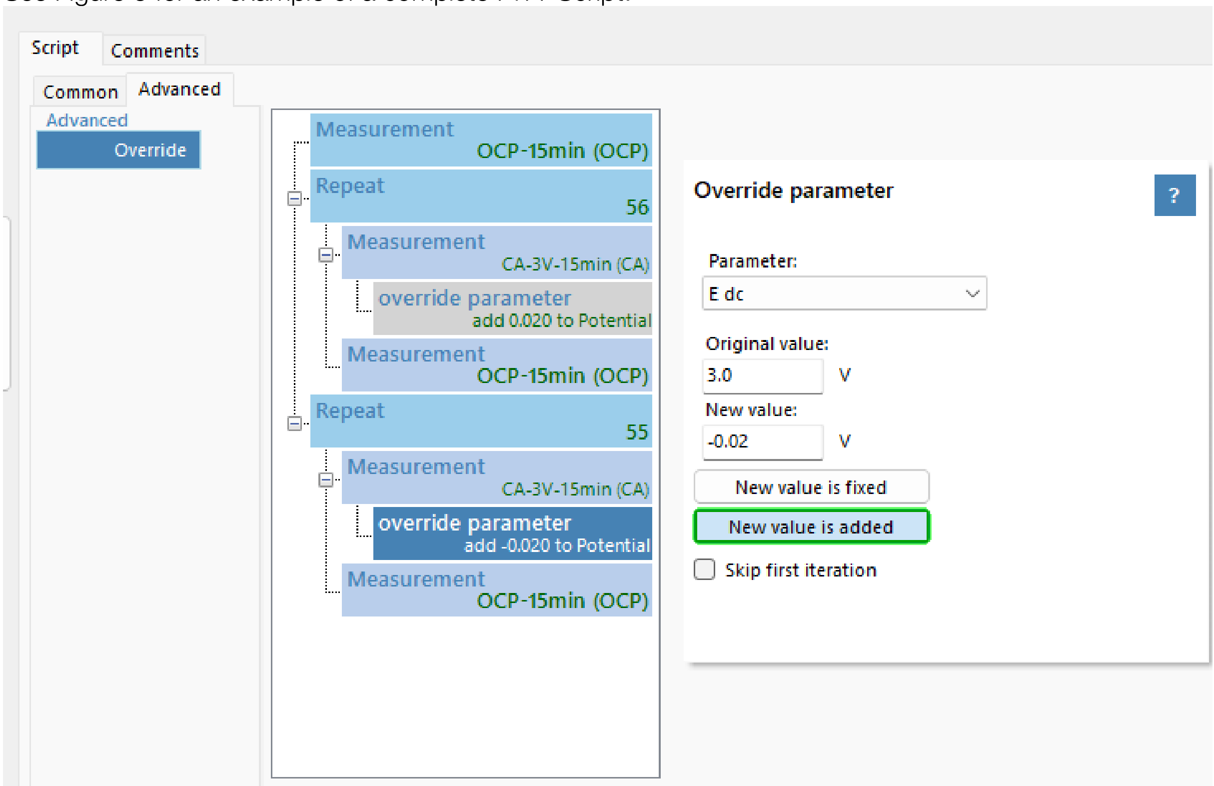


Figure 6 PITT script highlighting the override parameter option

HINT

Before the full experiment, run a short preliminary experiment to verify the setup. For example, reduce the step times and connect a dummy cell. This allows you to quickly confirm that the response behaves as expected before running the full experiment.

3.2 GITT using MultiStep Potentiometry

GITT (Galvanostatic Intermittent Titration Technique) consists of a series of constant current pulses. Each pulse is followed by an OCP step. As the applied current is always the same, GITT can be configured easily with the MultiStep Potentiometry technique available in PSTrace. Simply select it in the Technique filed and configure as follows:

- Applied current range: select one up to 3 times lower than the current you need to apply

- Cycles: the number of pulses
- Levels: set as 2.
- i level 1: set the current multiplier. I.e. if you want 5 mA, using the 10 mA range, set 0.5.
- i level 2: set as zero (it will be the OCP step).
- t1 and t2: pulse duration.

This will configure a GITT for charging a battery (positive current). See Figure 7 for an example of GITT experiment.

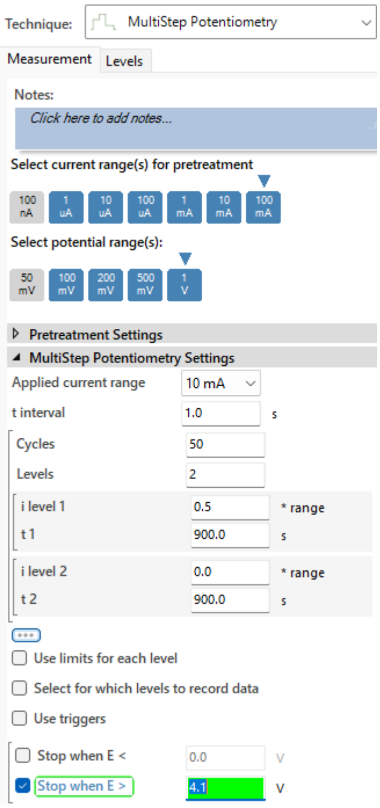


Figure 7 MultiStep Potentiometry method example for a typical GITT experiment

Likewise, set another MultiStep Amperometry experiment but with negative current in the ‘i level 1’ to be a discharge experiment. You can run these steps separately, or you can use a simple Script with 2 Measurement blocks to make them run sequentially.
The time will reset when starting the discharge step, but you will see a continuous timeline within each charge or discharge step.