Pytentiostat documentation

Oliver Rodriguez

Module: potentiostat

Module to select the potentiostat to use and perform electrochemical experiments (CV, LSV, CA, OCP).

pytentiostats.Setup(model=0, path_exe=‘,’, folder=‘,’, verbose=1)

Initializes the potentiostat to use and selects folder to save data files.

Parameters

- model: *str* or *False*. Potentiostat model to use. Accepted values: 0, False, ‘chi760e’
- path_exe: *str*. Directory where the potentiostat executable is located.
- folder: *str*. Name of and existing directory where data files will be saved.
- verbose: *boolean*. Print information on the console. 1 by default.

Methods

pytentiostats.Setup.info() Prints the potentiostat model, executable path and directory to save the data files.

pytentiostats.Technique(text=’‘, fileName=’CV’)

Class to initialize electrochemical techniques.

Parameters

- text: *str*. String containing the macro or commands to send to the potentiostat.
- fileName: *str*. Base filename to save data files.

Attributes

- bpot: *boolean*. False by default. If True, activates the second working electrode.

Methods

- *Technique.writeToFile()*. Saves the macro or commands send to the potentiostat as a file.
- *Technique.run()*. Performs the measurement and calls writeToFile to save the macro.
- *Technique.message()*. Used to output to console when the experiment starts and finishes.
- *Technique.bipot(E=-0.5, sens=1e-6)*. If pytentiostats.Technique.bpot is set to True, it will activate the second working electrode with the specified parameters.

pytentiostats.CV(Eini=-0.2, Ev1=0.2, Ev2=-0.2, Efin=-0.2, sr=0.1, dE=0.001, nSweeps=2, sens=1e-6, fileName=‘CV’, header=‘CV’)

Parameters

- Eini: *double*. [V] initial potential.
- Ev1: *double*. [V] first vertex potential.
- Ev2: *double*. [V] second vertex potential.
- Efin: *double*. [V] final potential.
- sr: *double*. [V/s] scan rate.
- dE: *double*. [V] potential increment.
- nSweeps: *int* number of sweeps.

- sens: *double.* [A/V] current sensitivity.
- fileName: *str.* Name to save data file. File will be overwritten if it exists.
- header: *str.* Header of the txt data file. Use it to include comments.

pytentiosats.LSV(Eini=-0.2, Efin=0.2, sr=0.1, dE=0.001, sens=1e-6, fileName='LSV', header='LSV')

Parameters

- Eini: *double.* [V] initial potential.
- Efin: *double.* [V] final potential.
- sr: *double.* [V/s] scan rate.
- dE: *double.* [V] potential increment.
- sens: *double.* [V/A] current sensitivity.
- fileName: *str.* Name to save data file. File will be overwritten if it exists.
- header: *str.* Header of the txt data file. Use it to include comments.

pytentsiostats.CA(Estep=0.2, dt=0.001, ttot=2, sens=1e-6, fileName='CA', header='CA')

Parameters

- Estep: *double.* [V] potential step to apply.
- dt: *double.* [s] time increment.
- ttot: *touble.* [s] total time of the step.
- sens: *double.* [V/A] current sensitivity.
- fileName: *str.* Name to save data file. File will be overwritten if it exists.
- header: *str.* Header of the txt data file. Use it to include comments.

pytentiostats.OCP(ttot=2, dt=0.01, fileName='OCP', header='OCP')

Parameters

- ttot: *double.* [s] total time of the step.
- dt: *double.* [s] time increment.
- fileName: *str.* Name to save data file. File will be overwritten if it exists.
- header: *str.* Header of the txt data file. Use it to include comments.

Module: file

Performs common operations on files created by the pytensiostat library.

pytentiostats.Read(text=False, model=False)

Reads a file created with the pytensiostat library.

Methods

*pytentiostats.Read.read(text=0, model=0)

- text: *str* or *False.* Text to locate in file to use as starting point to read.
- model: *str* or *False.* Potentiostat model used when creating the files.

*pytentiostats.Read.search(text)

- text: *str.* Text to locate in file to use as starting point to read.

pytentiostats.LoadXY(fileName=‘file’, folder=‘.’, skiprows=0, delimiter=‘,’, model=0)

Loads a general data file with X and Y columns.

Parameters

- fileName: *str*. Filename of the file to read with extension included.
- folder: *str*. Directory where the file is located.
- skiprows: *int*. Number of rows from the top to skip.
- delimiter: *char*. Character used to delimiter columns: ‘,’, ’ ‘,’’̂, etc.
- model: *str* or False. Potentiostat model used when creating the files.

Returns data: LoadXY instance - x: *numpy array* - y: *numpy array*

pytentiotsats.LoadCV(fileName=‘file’, folder=‘,’, model=0)

Loads a cyclic voltammogram data file obtained with the pytentiostat library.

Parameters

- fileName: *str*. Filename of the file to read with extension included.
- folder: *str*. Directory where the file is located.
- model: *str* or False. Potentiostat model used when creating the files.

Returns data: LoadCV instance

- E: *numpy array*. [V] potential array.
- i: *numpy array*. [A] current array.

pytentiotsats.LoadLSV(fileName=‘file’, folder=‘,’, model=0)

Loads a linear sweep voltammogram data file obtained with the pytentiostat library.

Parameters

- fileName: *str*. Filename of the file to read with extension included.
- folder: *str*. Directory where the file is located.
- model: *str* or False. Potentiostat model used when creating the files.

Returns data: LoadCV instance

- E: *numpy array*. [V] potential array.
- i: *numpy array*. [i] current array.

pytentiotsats.LoadCA(fileName=‘file’, folder=‘,’, model=0)

Loads a chronoamperogram data file obtained with the pytentiostat library.

Parameters

- fileName: *str*. Filename of the file to read with extension included.
- folder: *str*. Directory where the file is located.
- model: *str* or False. Potentiostat model used when creating the files.

Returns data: LoadCV instance

- t: *numpy array*. [s] time array.
- i: *numpy array*. [i] current array.

pytentiotsats.LoadOCP(fileName=‘file’, folder=‘,’, model=0)

Loads a chronoamperogram data file obtained with the pytentiostat library.

Parameters

- fileName: *str*. Filename of the file to read with extension included.

- folder: *str*. Directory where the file is located.
- model: *str* or False. Potentiostat model used when creating the files.

Returns data: LoadCV instance

- t: *numpy array*. [s] time array.
- E: *numpy array*. [V] potential array.