PySigView Documentation

Release 0.0b1

Jan Cimbalnik

Jan 05, 2022

Contents:

PySigView (Python Signal Viewer) is a signal viewer written in pure python using PyQt. It utilizes the power of GPU through Vispy library. This allows for displaying signals with high sampling frequencies over a long period of time. The basic functionality of signal viewing is extended though plugins.

CHAPTER 1

PySigView core

PySigView core has a couple of basic concepts. Based on VisPy library the viewer utilizes GPU acceleration to view signals with high sampling frequencies over a long period of time with the possibility to quickly zoom in and out. The viewer core operates with 1D NumPy arrays so the signals can have different sampling frequencies.

1.1 Adding data formats

The viewer can support any data format as long as there exists a library to read the data into python. To add the data format to the viewer a short wrapper code must be created to provide the data to the viewer through the specified API (LINK).

Note: PySigView has a separate library for the data server (pysigview_cs). To introduce a new format to the server the wrapper file has to be copied into this library as well.

1.2 Buffering

The viewer contains buffering capabilities. The buffer is circular and its extent is defined in number of windows. The buffer can be turned on/off in the viewer Preferences. For computers with low RAM there is an option to turn on hard drive buffering which is slower but provides enough space for buffering. Buffering utilizes Bcolz library for fast compression in memory and/or on disk.

CHAPTER 2

Interface

The basic interface consists of the main window with top bar and widgets. The main widget is the area where signals are displayed while the rest are dockable widgets (plugins) that can be moved and anchored around the main widget.

2.1 Top bar

Top bar includes basic functions of the viewer such as opening files, modifying preferences and bug reporting.

2.1.1 File menu

The file menu contains actions to either open file or MEF session and close PySigView.

Another action is to connect to a PySigView server which can subsequently serve data to the viewer from a remote (or even local) server as if it were a local file.

Note: This functionality requires the package pysigview_cs to be installed. The package is installed by default with PySigView.

Apart from this basic functionality it allows for saving and loading PySigView session. The PySigView session stores the current state of the viewer including any plugin data. This can be useful when saving unfinished work such as signal annotation or when sharing your work with a colleague.

2.1.2 Tools menu

Tools menu contains preferences action to modify the preferences for the application and plugins.

2.1.3 Help menu

Help penu contains action to report a bug on GitHub.

2.2 Main widget and channels

The main widget and channels widget are tightly bound together and are the only widgets that are required for the PySigView to work. Upon opening a file the available channels are displayed in the lower hidden channels field (NUMBER). To display signals drag&drop the marked channels to either higher visible channels field (NUMBER) or to the main widget (NUMBER).

Tip: To select multiple channels either hold down the SHIFT key or CTRL key

2.2.1 Signal display widget

The signal display widget is in the middle of the window and acts as a center point for all dockable widgets. The signals are displayed here and it provides basic operations.

Basic keyboard shortcuts:

- Up arrow - increase the amplitude of displayed signals
- Down arrow - decrease the amplitude of displayed signals
- Right arrow - move one window forward
- Left arrow - move one window backward
- Q - increase time span
- A - decrease time span
- Shift + Right arrow - move half window forward
- Shift + Left arrow - move half window backward

Basic mouse operations:

- Right mouse button - activate zoom. Moving the mouse right/left zooms in and zooms out along the x-axis respectively. Moving the mouse up/down zooms in and zooms out along the y-axis respectively.
- Left mouse button - activate pan. Moving the mouse in the desired direction while zoomed in moves the field of view.
- Wheel - activates vertical zoom - only a portion of the displayed signals is zoomed in. This is useful for finding small events.
- Shift + mouse - signal highlighting mode.
- Ctrl + mouse - signal measurement mode. A click in the mode creates a fixed point from which the amplitude and time is measured.

Top bar tools:

- Camera (NUMBER) - Take a snapshot of the signal display widget.
- Grid (NUMBER) - Enable / disable grid in the signal display widget.
- Up/down arrows (NUMBER) - Enable / disable signal autoscale.
- Diskette (NUMBER) - Save the currently displayed data to disk.
- Circle arrow (NUMBER) - Reload recording metadata (useful when the recording is being written).
- Play button (NUMBER) - Automated window shift. Useful when looking through the signal.

- Research / browse dropdown (NUMBER) - Research mode displays all data but is slower. Browse mode down-samples the data proportionally to the number of pixels but is faster.
- Color mode dropdown (NUMBER) - Apply color map to either individual channels or channel groups.

2.2.2 Channels

This plugin is the only one that must be installed with PySigView. It serves for basic operations with signals. The bottom part (NUMBER) shows hidden channels while the top part (NUMBER) shows visible channels.

The visible channels pane consists of so called “collections” which are space holders for individual signal “containers”. Collections can be drag&dropped within the visible channels pane to change their locations. Individual containers can be drag&dropped to different collections to draw over each other. This is useful for comparison of two signals.

Signal containers are used for providing information about signals and for simple operations with them. Removing the tick serves for hiding the signal. The color dot shows the current signal color and for changing it.

When the signal container is unfolded it provides basic information about the viewed signal with some modifiable fields. Time span and autoscale can be modified for individual signals.

2.2.3 Annotations plugin widget

This plugin adds the ability to create / modify / load / save annotations. There are four types of annotations depending on whether they are a one time event or a start/stop event and whether they belong to one channel or the whole recording. The annotation plugin adds functionality to the main signal display widget:

- Shift + 1x Left mouse button (NUMBER) - Add one time annotation to the highlighted channel.
- Shift + 2x Left mouse button (NUMBER) - Add start/stop annotation to the highlighted channel.
- Shift + 1x Right mouse button (NUMBER) - Add one time annotation to the whole recording.
- Shift + 2x Right mouse button (NUMBER) - Add start/stop annotation to the whole recording.

The annotations can be modified in the annotation list pane. Annotations are divided into annotation sets which can be enabled/disabled and their color can be changed. The icon to the right of the set name displays a dialog with the list of individual annotations. Here the annotation info can be modified. The dialog also allows browsing through annotations by either activating the browse mode or by clicking on the index of the annotation (leftmost number in the row). When Delete is pressed the currently selected annotation is removed from the set.

Top bar tools:

- Folded page icon (NUMBER) - Load annotations form a python pickle.
- Diskette icon (NUMBER) - Save annotations into a python pickle.
- Database arrow down icon (NUMBER) - Download annotations from a database. A database connection has to be active (see DATABASE PLUGIN).
- Database arrow up icon (NUMBER) - Upload annotations from a database. A database connection has to be active (see DATABASE PLUGIN).
- Plus icon (NUMBER) - Add a new annotation set.

Note: The underling data structure of annotations is a pandas.Dataframe. Hence the python pickles used for saving and loading annotations contain pandas dataframes that can be modified outside the viewer.

2.2.4 Navigation bar plugin widget

This plugin serves for basic navigation through the recording. It displays the current displayed time point (uUTC / date) and the current time span. Both of these fields can be modified to either jump to a specific time or to manually change the time span.

The bottom bar shows the current position in the recording, the extent of data loaded in the buffer(see BUFFER) (if activated) and recording discontinuities if the data format supports them. Left clinking on a point in the lower pane moves the displayed channels to the selected time point.

2.2.5 Database plugin

This plugin allows to create database connections especially for downloading and uploading annotations or for getting information about channels.

2.2.6 Measurements plugin

The plugin

2.2.7 IPython console plugin

The plugin allows for direct interaction with the application.

TODO

Tip: The console cane be utilized for applying custom scripts. For processing a displayed signal and visualizing the results in a separate window for example using matplotlib library. The signal emitted when the signal is changed can be connected to dynamically change the result window.

2.2.8 Transforms plugin

The plugin introduces signal processing into PySigView. Individual signal transformations can be chained to create final result.

To create a transformed signal drag&drop signal collections or containers to the transforms widget area (NUMBER) and select one of the containers. The signal will be previewed in the preview area (NUMBER). Select one of the transform from the list (NUMBER) and enter the parameters. Upon clicking on Set button (NUMBER) the transform signal will be visualized in the preview window. To attach the transform to the transform chain click the Apply button (NUMBER). You can either apply the transform to a single channel or all of the channels by ticking the option (NUMBER). Another transform can now be introduced into the transform chain. Once the transform chain is completed the transformed signals will be created by clicking Apply button (NUMBER). The signals can be either transformed or transformed copies can be created.

2.2.9 Measurement plugin

The measurement plugin provides detailed information about signal frequencies.

To introduce a signal into the plugin simply measure it in the signal display widget (using Ctrl + mouse). The signal will be previewed in the preview pane (NUMBER) and its corresponding spectrum in the lower pane.

The plugin supports two options - either a spectrum is produced or a spectrogram image. The parameters can be adjusted using parameter widgets (NUMBER)

CHAPTER 3

Indices and tables

- genindex
- modindex
- search