User Manual

Post-processing in (QT + Python) GUI: Post-Processing Plotter (Pcube)

Ajey Dikshit
IIT Bombay

June 2022

1 Steps to load the file in QT Creator

1. Ensure QT Creator version 5.12 is installed with the required Python packages.
2. Extract the download github zip. A tree of the extracted file is shown in Figure 1.

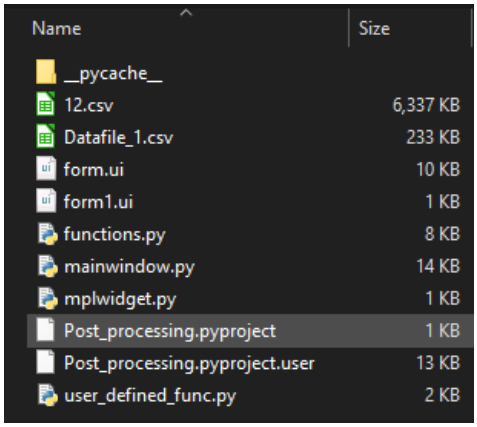


Figure 1: Extracted files from the github zip.

3. Open the `*.pyproject` file in the QT Creator.
4. Set the run configuration to `mainwindow.py` as shown in the Figure 2.

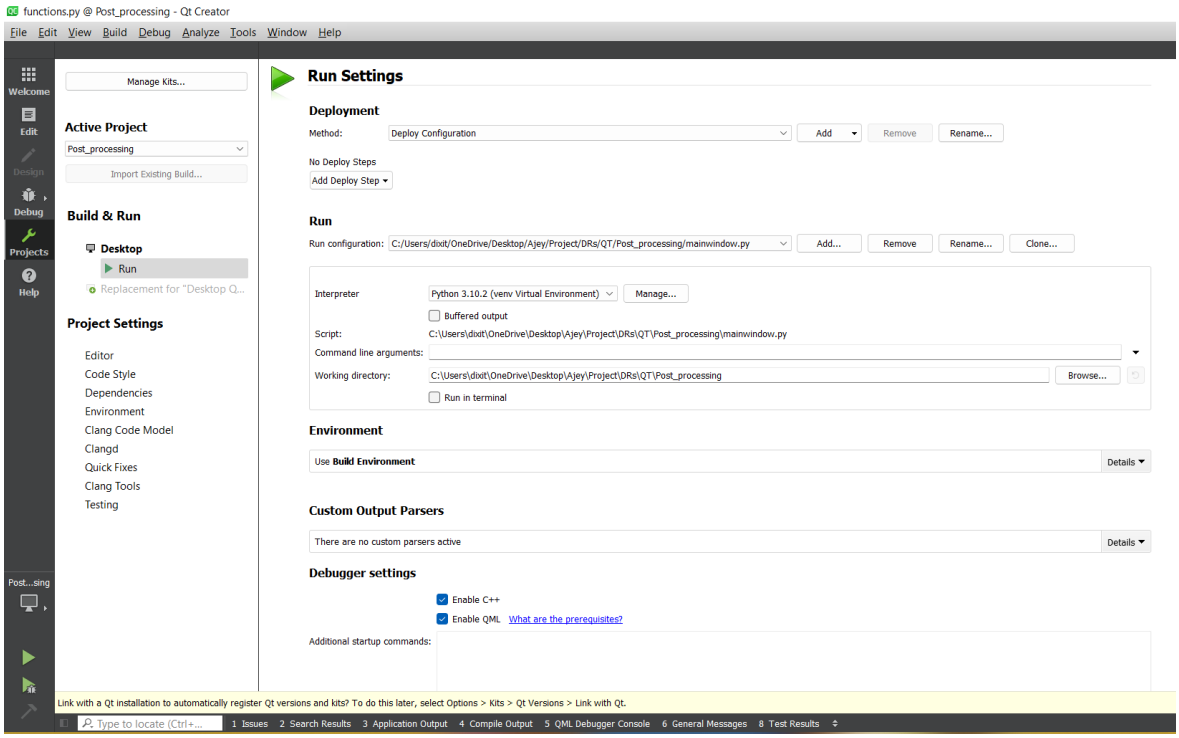


Figure 2: Changing the run configuration for a project.

5. Switch to edit mode and then click run (green triangle) button or press `Ctrl + R` for executing the project. With a successful run, the following Pcube GUI will appear (see Figure 3).

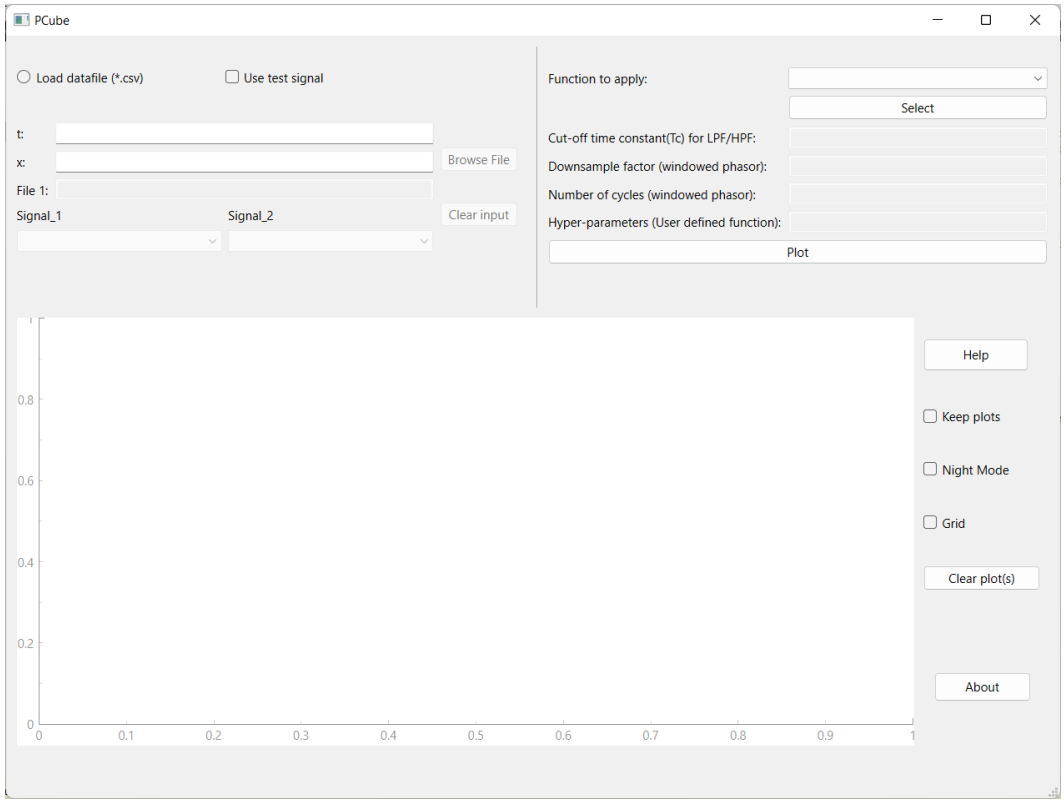


Figure 3: Pcube GUI

2 Using the Pcube GUI

There are three main area of Pcube GUI, these are input area, function area, and the plotting area as shown in Figure 4.

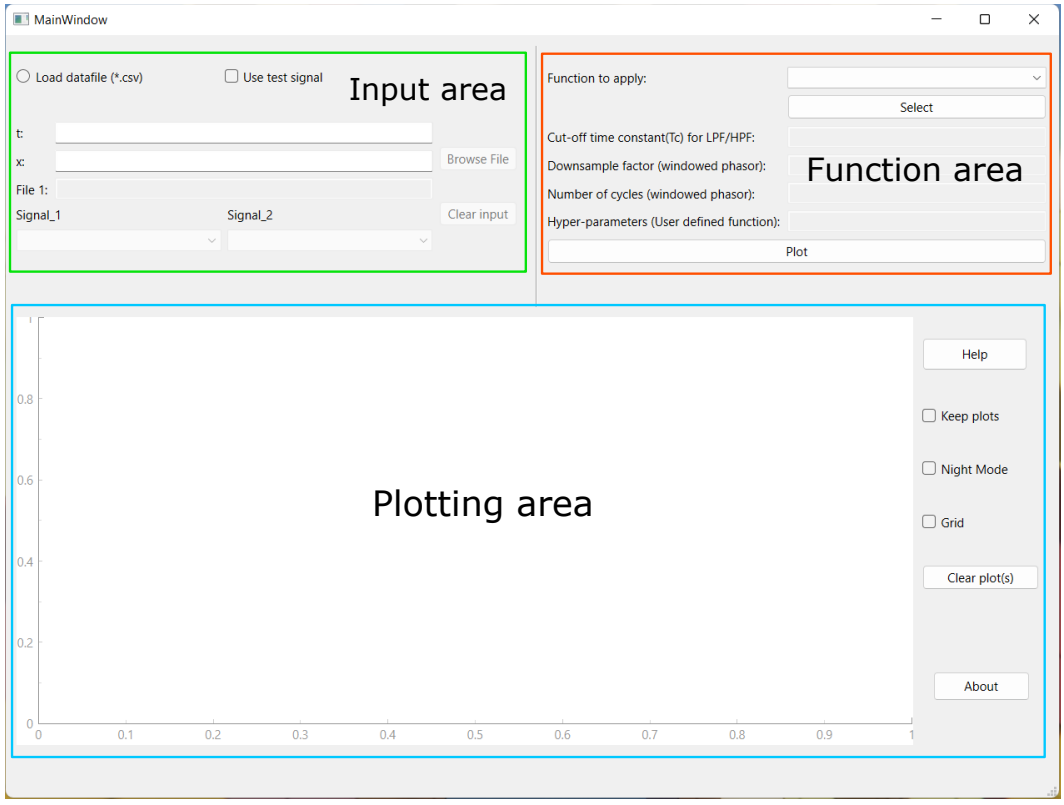
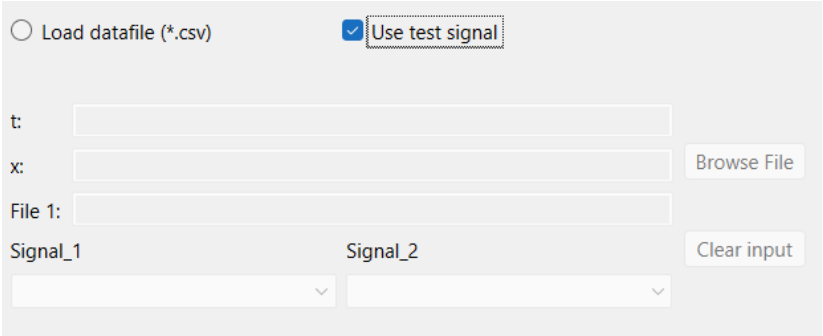


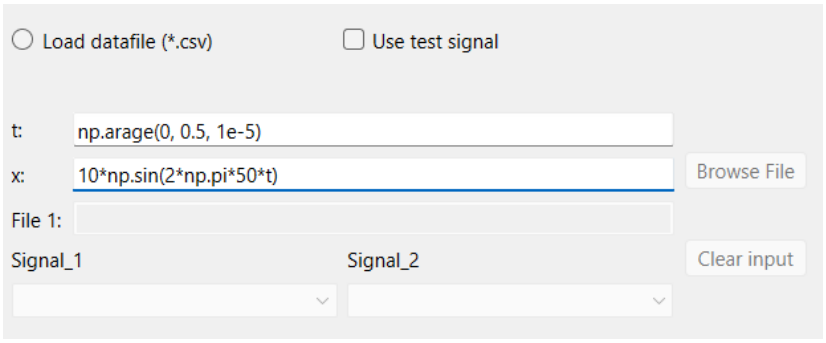
Figure 4: GUI

Input area of Pcube, facilitate three modes of data entry which are listed below:

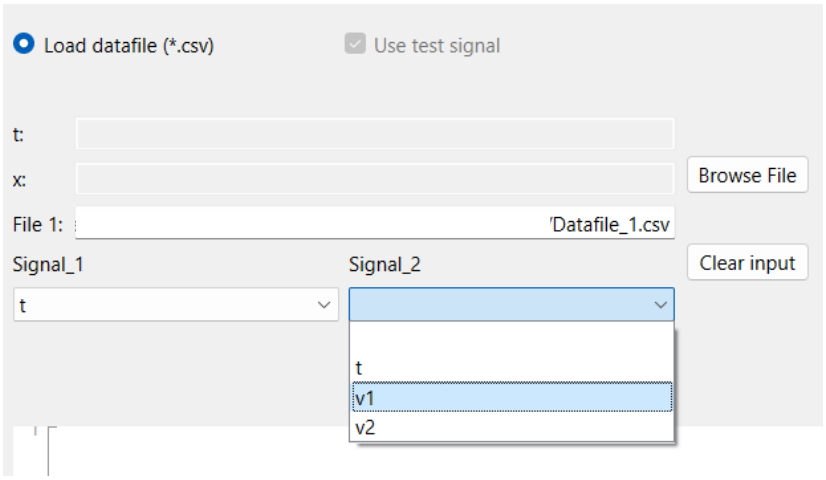
1. The quickest data entry can be made using the test signal check button. This test signal is a pre-defined signal containing a signal with multiple harmonics mainly DC, 50 Hz, and 500 Hz component. This feature can be used to test various post-processing functions quickly in Pcube GUI.



2. Using expressions: here a signal can be constructed using standard python mathematical syntax.



3. Using *.csv upload file feature: Here the first row of the *.csv file should be present, since these entries will be used in the signal slots as shown below. Signal_1 slot is for x-axis whereas Signal_2 slot is for the y-axis in the plot.



Function area of Pcube, facilitates various post-processing functions which can be applied on the signals uploaded from the input area. These functions are listed in the following figure.

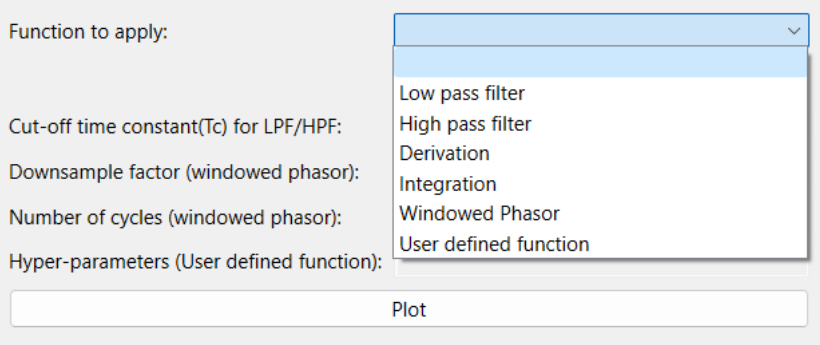


Figure 5: List of the post-processing functions.

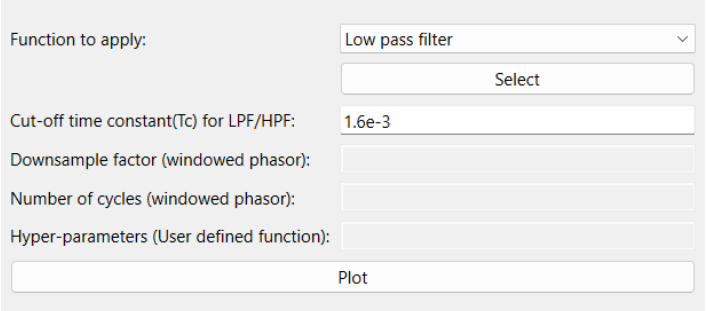


Figure 6: First order low pass filter

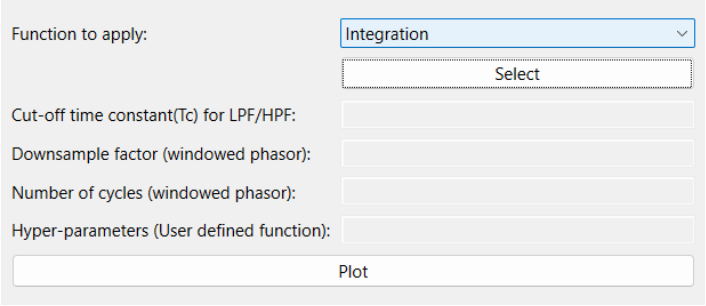


Figure 7: Integration

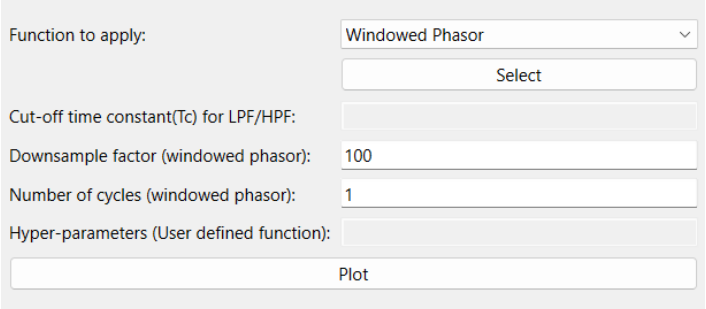


Figure 8: Windowed Phasor

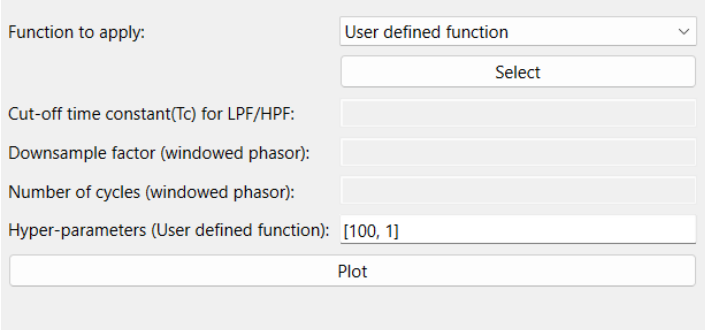


Figure 9: User defined function

Plotting area of Pcube, provides a canvas area where the input signals and the post-processed version of the signals are plotted.

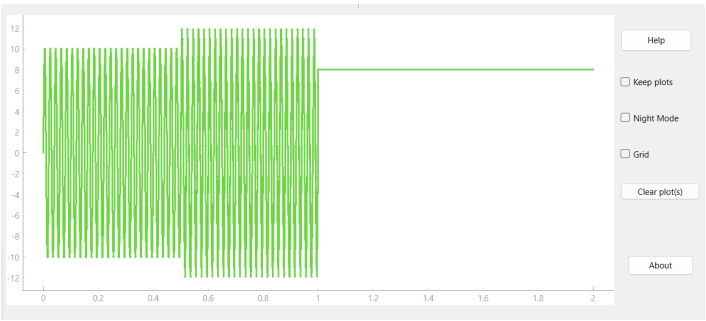


Figure 10: No grid, white background

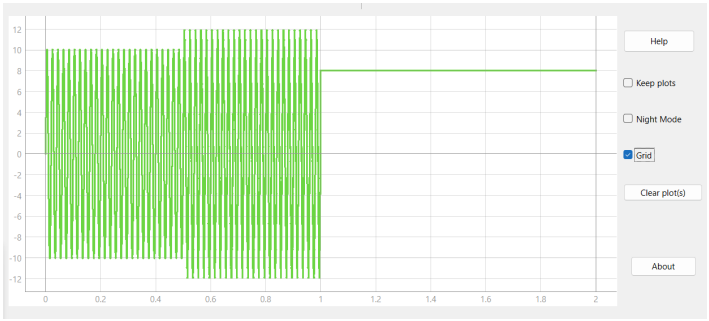


Figure 11: Grid, white background

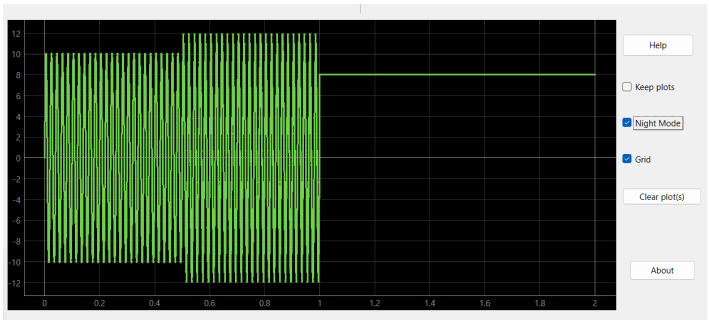


Figure 12: Grid, black background

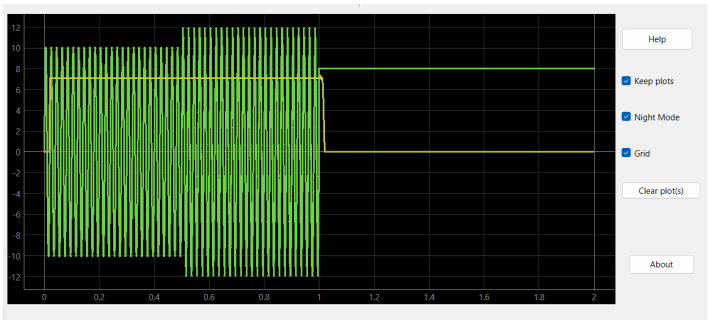


Figure 13: Windowed phasor applied, downsample factor:100, cycles:1

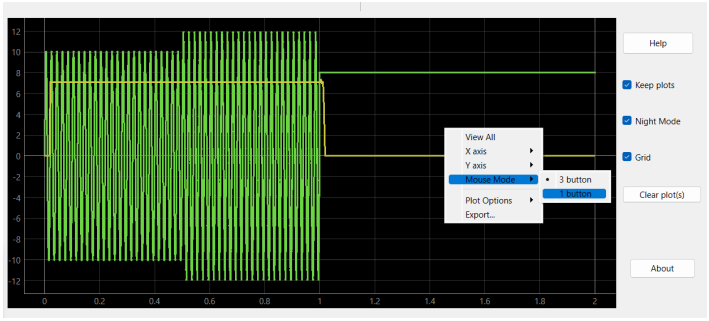


Figure 14: Mouse modes

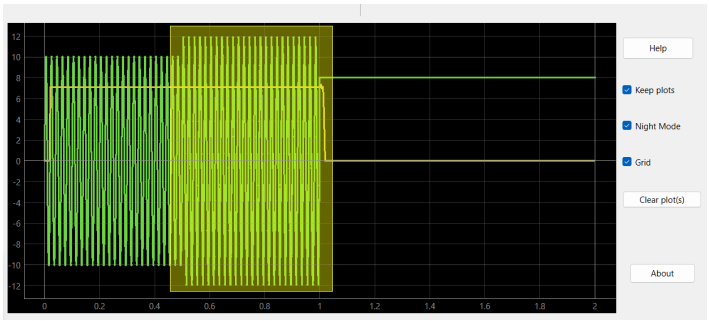


Figure 15: Selecting an area of plot

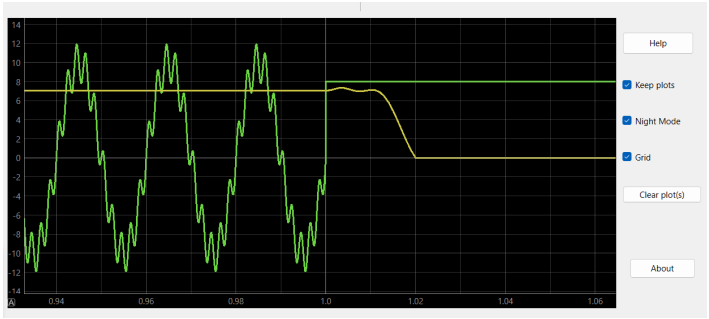


Figure 16: Zoomed in plot

3 User defined function

When writing an user defined function, the user has to keep few things in mind.
The input for the function has to be 3 arguments, which should include 2 inputs followed by a list of all the other inputs, the code has to be written using the list indexing. Even if no parameters exist, a 3rd argument must be passes (just don't use it in the function).An example has been given below:

```
def user_func(t, x, tc):
h = t[1] - t[0]
y = np.zeros(len(t))
i = 0
while i <= len(x) - 1:
    y[i] = (x[i] - x[i - 1]) / h
    i = i + 1
return [y, t]
```

If the user defined function has no hyperparameters they have to pass an empty list `[]` in the hyperparameter cloumn, the name of the user defined function can't change, and the function must return 2 values.
If giving file input, it is better to have a row dedicated to headers as it will make the user's life quite easy.