MC's PlotXY—A general-purpose plotting and post-processing open-source tool

Massimo Ceraolo
University of Pisa, Italy

A B S T R A C T

MC's PlotXY, in the following just called PlotXY, is a free piece of software able to make line plots and bar charts from data deriving from measuring devices or simulation programs using several formats. It allows to visualize and post-process these data without the need to have on the target computer the original software that generated it. It was written from the ground-up having researchers and scientists in mind, and therefore is optimized for speed, clarity, soundness.

Its special characteristic distinguishing it from the large base of existing plotting software is its flexibility: using this small piece of software, the user can do plotting and post-processing activities on the results obtained from other sources (simulation software or measuring devices) without having these sources installed on the computer where post-processing is made.

Moreover, it allows reading file formats rarely available on non-specific software, such as ATP, Comtrade, LabView formats. Although the support of some of these formats is not complete, being the software open-source, anyone can contribute to expanding them, operating on just a single cpp file.

It is developed using the Qt platform, which allows good GUI design, and already available for Windows, Macintosh, and Linux systems.

This paper describes the software characteristics and some programming techniques used to develop it.

© 2019 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Motivation and significance

MC’s PlotXY program (in the following simply referred to as PlotXY\textsuperscript{1}) was first built in 1998. Its initial motivation was to supply a powerful and easy-to-use plotting program for outputs from the freeware ATP program, which at those times had not a satisfactory plotting tool, in the author’s opinion. Its first description is now very old [1].

In recent years, it has completely been re-written, using the great power of Qt simulation environment, which, among other things, allows to compile smoothly a program for different platform targets. Taking advantage of this rewrite, modern features have been added to keep it up-to-date, such as the capability

\footnotetext{1}{E-mail address: massimo.ceraolo@unipi.it.}

[1] Other programs that today can be found under “PlotXY” name. This paper just describes the one whose full name is “MC’s PlotXY”.

https://doi.org/10.1016/j.softx.2019.01.017

2352-7710/© 2019 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
to output plots in SVG PNG and PDF and to manage modern 4k resolutions and multiple screens.

There are many packages able to do plotting and post-processing today, either commercial or freely distributed. Many of them are programs that in addition to doing numerical computations, have their own plotting routines to show their result, for instance:

- Mathworks’ Matlab [2]. This is a powerful and huge commercial program widely used worldwide for scientific computation, and includes sophisticated plotting and post-processing features;
- Scilab [3]. This is an open source program able to perform the majority of tasks of Matlab;
- Dassault Système’s Dymola [4] and OpenModelica [5]. They are tools to simulate models written in Modelica language. Dymola is commercial, OpenModelica is open-source.
- Siemens’ Amesim [6]. It is a simulation package, designed to build and simulate models written in a proprietary format, as well as in Modelica language.
- National’s LabView [7]. It is a systems engineering software for applications that require test, measurement, and control with rapid access to hardware and data insights. Measured data are written in ascii files having a peculiar format, that allows other pieces of software to read the measured data and accompanying information.

A specific plotting tool is Gnuplot [8], which is very powerful and distributed for free. But it is oriented to plot mathematical functions and does not make plots from stored data.

Many other plotting/post-processing programs exist, but according to the author’s knowledge, they all share characteristics of the previous examples: they are either plotting routines built-in in complex software, in many cases is commercial (such as those in the bullet list above), or routines devoted to mathematical functions plotting.

Typically they support their own proprietary and other formats, but they often lack the capability to understand some exoteric formats such as pl4 outputs from ATP [9] and COMTRADE [10].

None of them is in direct competition with PlotXY, because of its combined features:

- It is able to read some exoteric formats such as ATP pl4 and COMTRADE, but also has some support for Mathworks’ Matlab [2], National Instruments’ LabView [11] and other formats;
- it is open-source, and therefore anyone can propose improvements or additions. Indeed the program has been continuously updated even in very recent years following user suggestions;
- it is very small and portable, does not require the execution of an installation routine, and has a very steep learning curve.

This program however would not be appealing if it lacked the flexibility to use that today is required to visualization programs. It is rich in features that have been developed over the years, which are discussed in the following sections.

Another PlotXY-related paper has already appeared [12], which however was just a short communication, and did not consider the most recent development, mainly its conversion from proprietary to open-source code. This paper contains an updated and enhanced program description.

2. Software philosophy and characteristics

Before showing some examples of plot, here the basic choices behind the proposed software are presented.

2.1. GUI

The program has been conceived to be used in conjunction with other pies of software, with a very steep learning curve, on the same computer, at the same time. Therefore the following basic GUI choices were made:

- No menus. Carefully chosen commands are fast accessible through push-buttons
- Multiple windows, to ease as much as possible share of desktop space with other programs. They can be independently positioned everywhere on the computer desktop; in case of need they can be cleanly arranged clicking on the “arrange” button.
- Appearance repeatability across sessions. The user of this program typically selects positions and sizes of all the program widows carefully so that they share desktop space with other programs. These positions and sizes, can be retained from one session to the following one (this is obtained using OS functions such as the MS Windows’ registry).

2.2. Development platform

GUI programs must access operating systems’ Application Program Interface commands, to interact with it with mouse, display, keyboard, timers, etc. This makes normally migrating GUI software from one platform to another very complicated.

Instead, Qt [13] multi-platform programming environment comes with extensive libraries that operate as an intermediate layer between the application software and OS’s primitives, and therefore allows the paradigm “code once, deploy everywhere”. Qt is distributed GPL or LGPL, is very powerful and fast.

PlotXY is currently being compiled under Qt 5.7, but is compatible for compilation with any Qt versions, starting from 5.0. It consists of a set of .cpp and .h files plus some additional miscellaneous files.

To ease its distribution, it was wished to have in any case a single executable. As a consequence of this, the Windows version is self contained in a “.exe” file, the Mac version in an “app”, the Linux one in an Appimage [14].

2.3. Software main capabilities

It has been stated in Section 1 that the basic motivation of this program is to be a free and lean plotting and post-processing program, able to read data from multiple sources, and that this makes it different from the majority of existing other solutions.

All the implemented features were conceived having this idea in mind. These are listed in the following subsections, with some comments about how they are implemented and why.

2.3.1. Plotting capability

PlotXY had the basic features today expected from any plotting program such as possibility to plot signals both versus time or versus other signals, to plot several variables simultaneously taken from the same file or different files, fast multi-layer zoom and unzoom, peeking at numerical values. The user can see either the numbers as read from the input file, or linear interpolated values, if he prefers this.

A special care taken in PlotXY is on the choice of the numbers of axes. PlotXY tries to avoid, whenever possible, numbers containing too many digits on the axes. Much of the code was devoted to this, since some typical way numbers rare shown from other software are not straightforward. This is detailed in Section 4.3.

The program is designed around line plots. However it can also show bar charts, which are very useful to display the output of Fourier analysis.
2.3.2. Input/output capabilities

The idea behind this program requires the capability to read several input formats can be read. Many have been implemented, and the list grows with the help of users that show their needs.

The voluntary way this program is developed, in the past did not allow full support of complex formats; since however the program has recently become open-source, future development of additional input formats can be done with the help of anyone volunteering to do so.

PlotXY is already capable to read subsets of ATP [9], COMTRADE [10], LABVIEW [7] files.

It can also read and write version-4 Matlab files, and read a subset of version-5 format. It would be very easy to make it fully compatible with more recent Matlab versions, but this would imply distributing Mathworks’ libraries along with PlotXY executable, which is against its philosophy. The program Matlab reading software, was indeed directly coded in C language.

It can also correctly read any output from two prominent tools to simulate Modelica models, Dymola [4] and OpenModelica [5]. Although official documentation of these file formats does not exist, tests made show full compatibility with any simulation outputs from these tools. Note that Modelica, developed by the non-profit Modelica Association [15] is a fast-growing simulation modeling language, which allows the corresponding models to be simulated virtually with any Modelica-capable tools.

Plots or charts can be copied to clipboard, or saved in PNG, SVG, PDF files. New files containing a selected number of variables among those present in the input files can be created.

2.3.3. Post-processing capability

For a program having the ambition to operating as a useful tool to analyze data from several sources, some post-processing capability is mandatory.

This has been added in a very powerful way, since users can mix curves in a rather arbitrary way: formulas containing sums and/or products of any number of signals, mixing them with constants and using parenthesis are analyzed and plotted. Moreover the integral of input variables or combination of input variables can be created and plotted against time, thus allowing computing energies from powers, stored charges from currents, etc. Some technical details on this are shown in [12].

2.4. Speed

Despite the modern development platform, there is still room for good programming techniques.

To plot a line plot on screen it may well happen that some plotting actions are visually redundant. If, for instance several points lie on a straight line, it is sufficient to draw a line from the first up to last point of the straight line, dropping the intermediate ones.

Today, this is partly done directly by development platform libraries. Nevertheless, it is shown in [12], with that the implementation of an original redundant point — elimination routine brought very large time savings, even in comparison with the preprogrammed graphical primitives of Qt platform.

3. The open source package

The program has recently become an open-source project, made available on GitHub (repository “max-privato/PlotXY”). Programs created from this repository’ source code have the only limitation that they cannot read binary files created from the ATP program [9]. This is because this format is considered an ATP secret.

Binaries including ATP-reading capability are distributed along with this paper, and kept updated in a cloud area whose address is kept updated in the readme.md of the GitHub repository.

4. Examples of plot windows and usage

4.1. Access to files and variables

Since the program is conceived to plot data from different sources, e.g. for comparing simulated and measured data, an easy access to different files, file types and to the list of enclosed variables was pursued.

The left part of Fig. 1, shows several files loaded, with a list of the corresponding basic info on its top, and a list of the variables of the selected one on its bottom-left. By changing the current file the variable list is updated: this allows to create plots comparing signal for different sources. E.g. different files produced by the same program, or files produced by different program, or even comparing measured and simulated data. Two plot examples are shown in the top-right part of the figure.

Basically, PlotXY reads from input files the numerical data to be plotted. However, to be a better Modelica plotting tool, it reads, and tells the users, also the value of parameters i.e., quantities that do not vary during simulation. Parameters are for instance resistances or inductances of a circuit, masses or moments of inertias of mechanical systems, etc. A list of parameters of a Modelica output is shown in the bottom part, corresponding to the simulation reported in file #4 (Rectifier3ph.mat) in the top-left part of Fig. 1.

4.2. Plot types

While obviously it is not useful to propose here a comprehensive list of plot features, it is of interest illustrating some special characteristics, and the reasons that have brought to choosing them.

4.2.1. Line plots

Line plots have the usual features of zooming and unzooming, customizing color and line style, adding grids, using log or DB scales, etc., which are usually requires from plotting programs. Here we discuss only some special features that not always are present even in good plotting software.

Elimination of visual-redundant-points. When plotting plots with huge number of points it may well happen that some plot segments do not have to be traced on screen, since the correspondent pixels are already set. Avoiding this superfluous drawing can significantly enhance plotting speed. Details of this in [12]. Furthermore, to avoid sluggishness when the plot window is resized, during resizing only a bitmapped image of the displayed plot is resized, which is a very fat operation; the plot is redrawn only when resizing action is finished (i.e. when no resize operation is requested for 100 ms).

Peeking at numerical values. Since PlotXY had scientists and engineers in mind, it shows numerical values exactly as they are taken from the file: linear interpolation is possible, but not mandatory. Users can peek at numerical values either placing the mouse pointer near to curves (individual curve peeking) or moving a ruler with simultaneously shows all the data values corresponding to an individual point on the horizontal axis.

Twin vertical axes. Not all plotting section of simulation programs allow this feature, which is indeed very important when we want to compare curves having different orders of quantities, or even unit of measure

X-Y Plots: this allows plotting one variable against another. Some examples on how this feature appears when using PlotXY are shown in Fig. 2, showing time plots (left) the corresponding X–Y plot (middle) as well as a zoomed part of the latter (right).

Bar charts. Fourier analysis of periodic quantities gives rise to discrete values for amplitudes and phases. Nevertheless, it is not
uncommon to see these values displayed with line charts, which are confusing. PlotXY, instead, correctly uses bar chart, which include a specific feature to peek at numerical values, zooming capability, etc.

4.3. Automatic units of measure and prefixes, smart resizing

Many plotting programs display very large or small numbers in exponential form.

In the upper-left part of Fig. 3, for instance we see how the OpenModelica [5] graphical interface program does this. This is not good technical practice, since all the labels share the same exponent, which is better used just once. Other recent tools share this weakness, such as [4]. PlotXY, instead, adds a unique compact power-of-ten coefficient. When unit of measure is known, either because inferred from the variable name or because specified by the user, the standard prefixes ("µ", "m", "k", etc.) are used to make reading even easier. This is for instance shown in the top-right part of Fig. 3, where axes quantities are voltage and time.
When available, unit of measure information is taken from the input file; when this is impossible a convention based on the first letter of variable names can be used.

To have good scalability, the number of tick-marks on the axes must be reduced for small plots and augmented for larger ones. This is done by PlotXY, which uses a variable number of them that can be a minimum of three for tiny plots. Consider for instance the lower-right plots in Fig. 3, which compares the tiniest PlotXY and OpenModelica versions of the plot.

5. Impact

As discussed in Section 1, PlotXY has a unique set of features that makes it unique in comparison with existing plotting and post-processing software.

It is open-source, executable (Microsoft’s Windows Apple’s Mac OS X, and Linux) versions are distributed under LGPL license.

It allows making plots and post-processing data from many different sources, and exporting partial data, and bitmapped or vector-graphics plots. In allows creating multiple plots, with data taken from different sources, for instance simulation or measuring software.

It takes advantage of the state-of-the-art Qt development environment, can be useful to scientific and technical users, operating in many technical and scientific fields.

Acknowledgments

I want to express my gratitude to the University of Pisa, which has left me free to use my time, when not engaged in institutional tasks, to make the special activity of creating and maintaining PlotXY. This was gratifying per se and allowed me to make something useful to people from all over the world.

I acknowledge Perry Clements, from Canada, for the idea of creating an appimage to distribute PlotXY with a single executable under Linux, and to have personally created that appimage.

References

[1] Ceraolo M. Get a clear view simulated and measured data. IEEE Comput Appl Power 2000;13(4):36–42.
[2] Mathworks/Matlab Internet site http://www.mathworks.com.
[3] Scilab Internet site www.scilab.org.
[4] Dymola Systems Engineering modelling and simulation platform, https://www.3ds.com/products-services/catia/products/dymola/.
[5] OpenModelica official Internet site: www.openmodelica.org.
[6] Siemens’ LEM Imagine. Lab Amesim, https://www.plm.automation.siemens.com/en/products/lms/imagine-lab/amesim/index.shtml.
[7] National Instrument’s LabView official site http://www.ni.com/en-us/shop/labview.html.
[8] http://www.gnuplot.info/.
[9] European EMTF-ATP Users Group Internet site: ATP http://eegu-test.hostingkunde.de/.
[10] C37111-2013 - IEEE/IEC Measuring relays and protection equipment – Part 24: Common format for transient data exchange (COMTRADE) for power systems, https://standards.ieee.org/findstds/standard/C37.111-2013.html.

[11] labView format specifier: http://zone.ni.com/reference/en-XX/help/371363N-01/lvconcepts/format_specifier_syntax/.

[12] Ceraolo M. Experiences in creating a software tool to analyse and post-process simulated and measured data. In: Software practice and experience. ISSN 00380644, Wiley, http://dx.doi.org10.1002/spe.2632.

[13] Qt software development toolkit Internet site: https://www.qt.io/.

[14] AppImage Internet site www.AppImage.org.

[15] Modelica Association official site, www.modelica.org.