First Edition, Version 1.0

Please cite this book as:

Claudius Ptolemaeus, Editor,
System Design, Modeling, and Simulation using Ptolemy II, Ptolemy.org, 2014.
http://ptolemy.org/books/Systems.

17 Signal Display

Christopher Brooks and Edward A. Lee

Contents

Ptolemy II includes a number of **signal plotters**, shown in Figure 17.1. These can be found in the `Sinks` library, as shown in Figure 17.2. This appendix gives an overview of these capabilities, with emphasis on how to customize the plots. Once plots have been customized, saving the model containing the plotter will make the customization persistent.

17.1 Overview of Available Plotters

The plotters shown in Figure 17.1 provide a number of capabilities, and are all built on a common infrastructure. The most basic is **SequencePlotter**, which simply plots data values received on the input port. An example of a plot is shown in Figure 17.3. The mechanisms for customizing the title, axes, legend and signal plots will be covered in the following section.

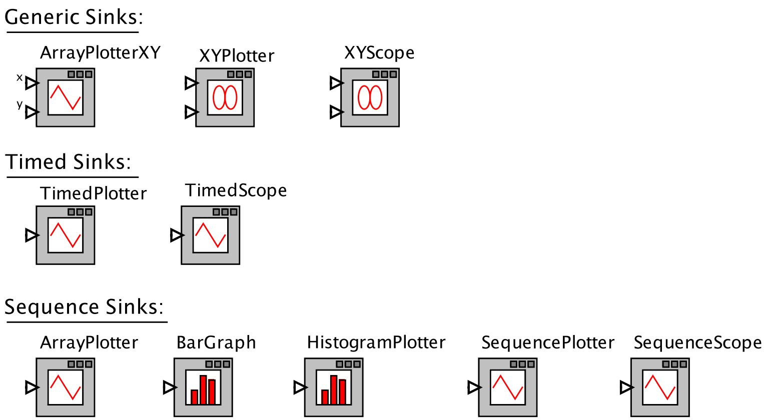

Figure 17.1: Available signal plotters.

The actors shown in the libraries in Figure 17.1 provide a variety of ways of displaying data. **ArrayPlotter** is similar except that it operates on input arrays rather than sequences. Whereas the SequencePlotter actor plots all of the input data over the entire run of the the model, the SequenceScope actor plots windows of input data, optionally overlaying them, as shown in Figure 17.4. The **SequenceScope** actor is more useful with long or infinite runs. It functions more like an oscilloscope in that it forgets old data, plotting only recent data, and overlaying windows of the data on each other.

The **TimedPlotter** actor plots input data as a function of the time stamps of the inputs, as shown in Figure 17.5. This plotter is useful in domains that advance model time, such as DE and Continuous. **TimedScope** is similar, though like SequenceScope, it functions like an oscilloscope and forgets old data.

The **XYPlotter** actor plots input data from one input port vs. input data from its other input port, as shown in Figure 17.6. **XYScope** is similar, though like SequenceScope, it functions like an oscilloscope and forgets old data. **ArrayPlotterXY** is similar, except that it operates on arrays of data rather than sequences.

BarGraph plots input arrays in the form of a bar graph. **HistogramPlotter** calculates a histogram of input data and then plots it as a bar graph, as shown in Figure 17.7. As shown in Figure 17.2, there is also a **ComputeHistogram** actor which calculates a histogram without plotting it.

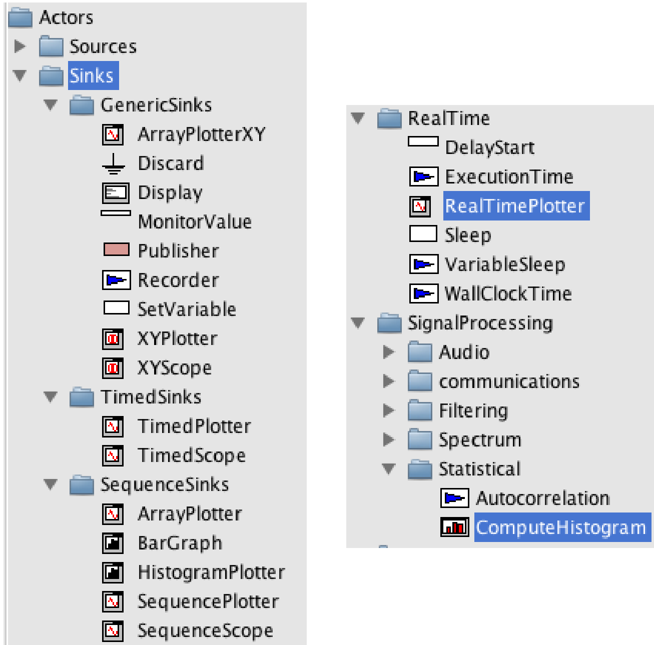

Figure 17.2: A variety of signal plotters can be found in the `Sinks` library.

Also shown in Figure 17.2 is **RealTimePlotter**, which plots input values as a function of real time elapsed on the computer executing the model.

17.2 Customizing a Plot

When used with its default configuration, the SequencePlotter will produce a plot like that shown in Figure 17.8, which is also shown in Figure 3.1. The default title is uninformative, the axes are not labeled, and the horizontal axis ranges from 0 to 255, which is not meaningful.* In the model that created this plot, which is shown in Figure 3.1, in one iteration, the Spectrum actor produces 256 output tokens. By default, the SequencePlotter just numbers these samples 0 to 255, and uses those numbers as the horizontal axis for the plot. But the horizontal axis may have more meaning in the model. In this particu-

*Hint: Notice the “$x10^2$” at the bottom right, which indicates that the label “2.5” stands for “250”.

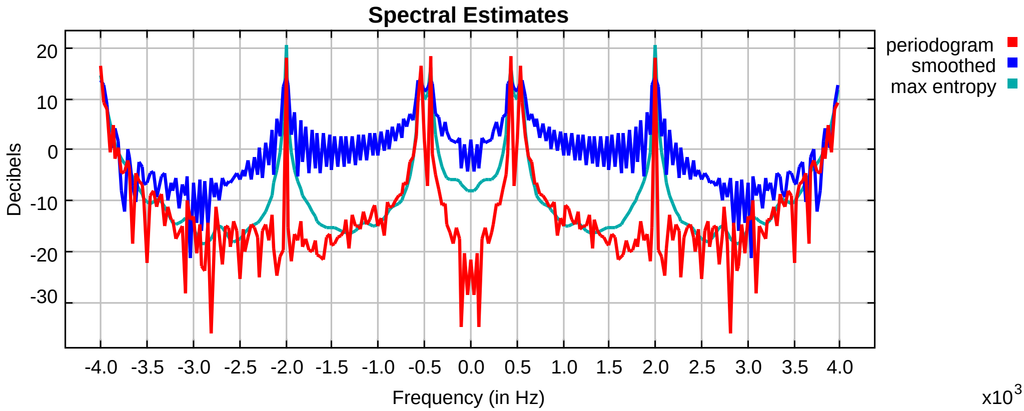

Figure 17.3: Example of a plot produced by the SequencePlotter actor.

lar example plot, the plotted data represent frequency bins that range between $-\pi$ and π radians per second.

The SequencePlotter actor has some pertinent parameters, shown in Figure 17.9, that can be used to improve the labeling of the plot. The *xInit* parameter specifies the value to use on the horizontal axis for the first token. The *xUnit* parameter specifies the value to increment this by for each subsequent token. Setting these to "-PI" and "PI/128" respectively results in the plot shown in Figure 17.10.

This plot is better, but still missing useful information. To control more precisely the visual appearance of the plot, click on the second button from the right in the row of buttons at the top right of the plot. This button brings up a format control window. It is shown in Figure 17.11, filled in with values that result in the plot shown in Figure 17.12. Most of these are self-explanatory, but the following pointers may be useful:

- The grid is turned off to reduce clutter.
- Titles and axis labels have been added.
- The X range and Y range are determined by the fill button at the upper right of the plot.
- Stem plots can be had by clicking on "Stems"
- Individual tokens can be shown by clicking on "dots"
- Connecting lines can be eliminated by deselecting "connect"

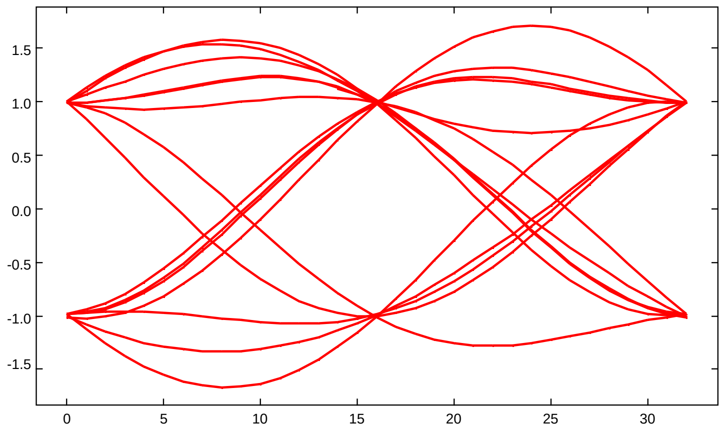

Figure 17.4: Example of a plot produced by the SequenceScope actor, which overlays successive windows of data, like what an oscilloscope does.

- The X axis label has been changed to symbolically indicate multiples of PI/2. This is done by entering the following in the X Ticks field:

  ```
  -PI -3.14159, -PI/2 -1.570795, 0 0.0, PI/2 1.570795, PI 3.14159
  ```

  The syntax in general is: *label value, label value, ...*, where the label is any string (enclosed in quotation marks if it includes spaces), and the value is a number.

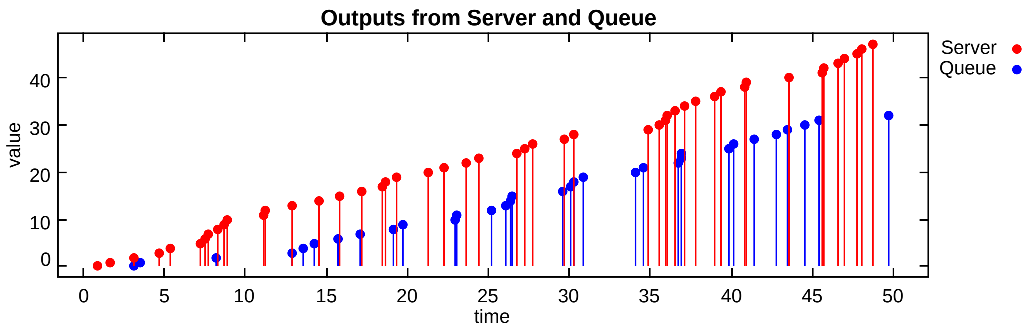

Figure 17.5: Example of a plot produced by the TimedPlotter actor, which plots input data as a function of the time stamps of the inputs.

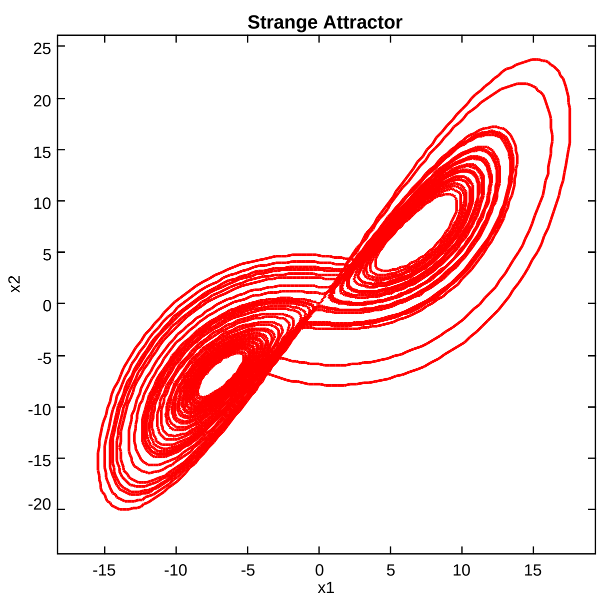

Figure 17.6: Example of a plot produced by the XYPlotter actor, which plots input data on one input port vs. input data on the other port.

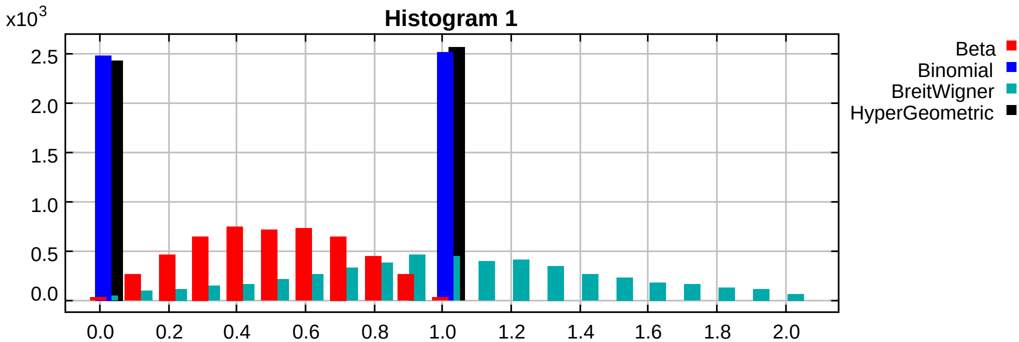

Figure 17.7: Example of a plot produced by the HistogramPlotter actor, which calculates and plots a histogram based on input data.

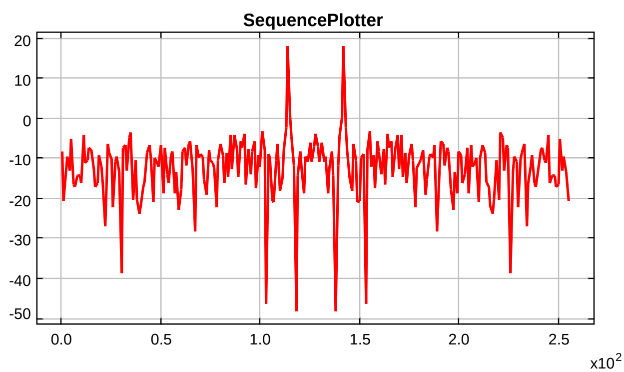

Figure 17.8: In its default configuration, a plot produced by SequencePlotter does not have informative labels.

Figure 17.9: Parameters for the SequencePlotter actor.

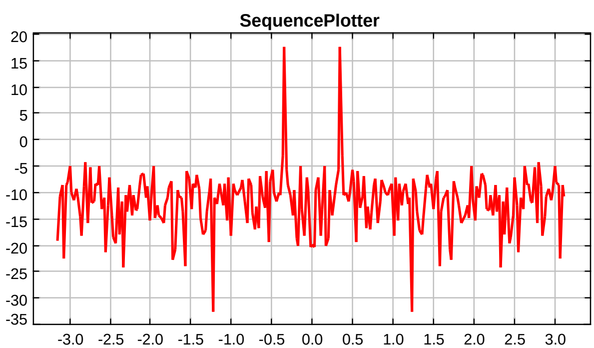

Figure 17.10: Better labeled plot, where the horizontal axis now properly represents the frequency values.

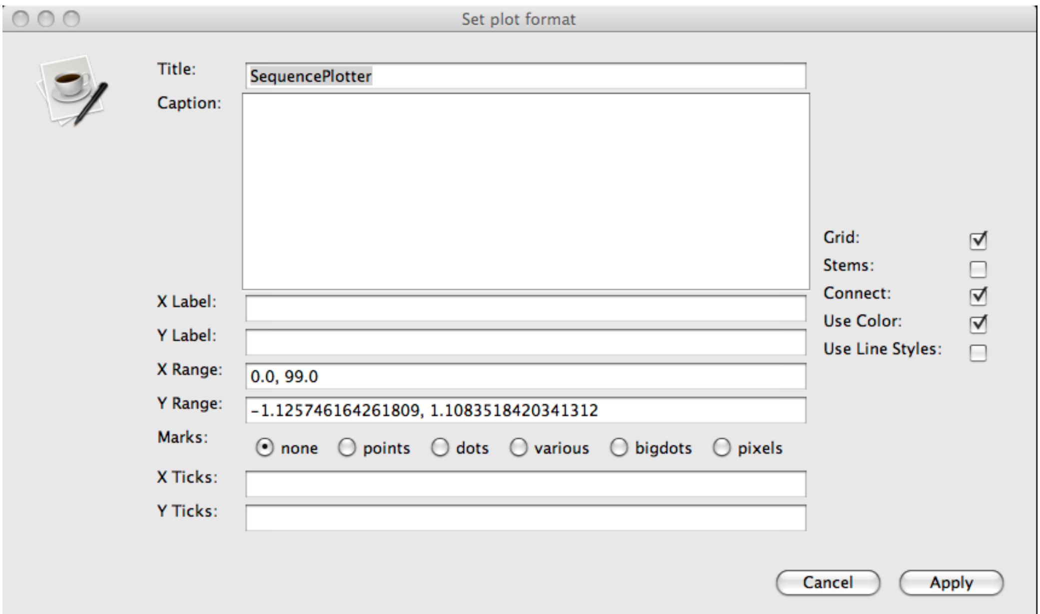

Figure 17.11: Format control window for a plot.

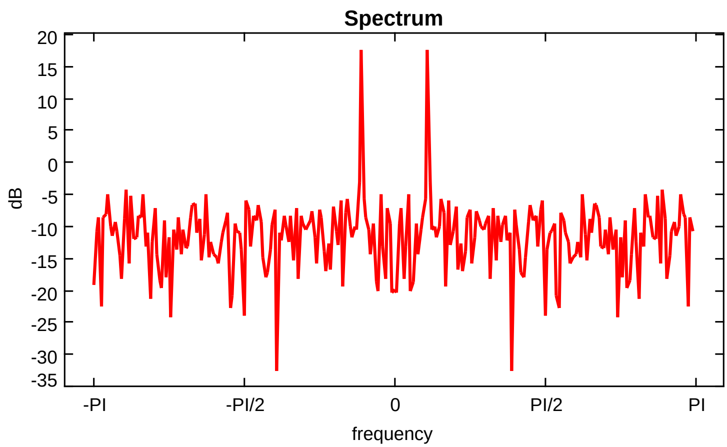

Figure 17.12: Still better labeled plot.