

The pst-sigsys Package

(version 1.31)

Farshid Delgosha
fdelgosha@gmail.com

June 18, 2010

Abstract

This package is a collection of useful macros for disciplines related to signal processing. It defines macros for plotting a sequence of numbers, drawing the pole-zero diagram of a system, shading the region of convergence, creating an adder or a multiplier node, placing a framed node at a given coordinate, creating an up-sampler or a down-sampler node, drawing the block diagram of a system, drawing adaptive systems, sequentially connecting a list of nodes, and connecting a list of nodes to one node using any node-connecting macro. The author welcomes all comments for further improvements of this package and suggestions for adding new macros or features.

Contents

1 Introduction

To use the pst-sigsys package, add the command

```
\usepackage{pst-sigsys}
```

to the preamble of the document. This package loads pstricks [?], pst-node [?], and pst-xkey [?] packages. Moreover, it activates polar coordinates through the \SpecialCoor macro defined by the pstricks package. Hence, all macros support polar coordinates. Simultaneously loading the pst-sigsys package along with some other packages in regular T_EX might be impossible due to memory restrictions. If T_EX runs out of memory, load the etex package.

Section 2 keeps a change log from previous versions of the package. All macros defined by the pst-sigsys package are introduced in Section 3. The extra functionalities of the package are introduced in Section ???. Many practical examples are provided in Section ??? that illustrate the applications of the introduced macros.

2 Change Log

- **Version 1.3 (06/18/2010):** In the `\pstick` and `\psTick` macros, the tick angle is either directly specified by the user or set by the `angle` key when unspecified. The `ticklength` key refers to the entire length of a tick, not half of it. The new key `killzero` is added to the `\psstem` macro. In the `\psldots` and `\ldotsnode` macros, the angle of dots is either directly specified by the user or set by the `angle` key when unspecified. Two new macros `\psadaptive` and `\psknob` are added. The new keys `framewidth`, `frameheight`, and `FillColor` are introduced.
- **Version 1.2 (01/15/2010):** Five new macros `\pstick`, `\psTick`, `\pssignal`, `\ldotsnode`, and `\ncstar` are added. The macros `\pshtick`, `\psvtick`, `\pshTick`, and `\psvTick` are not available any longer since their functionalities are carried out by the newly defined macros `\pstick` and `\psTick`. Codes for the macros `\pscicleop`, `\psframeop`, `\psldots`, and `\nclist` are updated. Four new keys `gratioWh`, `gratioWv`, `gratioHh`, and `gratioHv` are added that allow frames with edges proportional by the golden ratio. The global round-cornering settings are removed because of their undesired effects in other packages. Hence, the option `notelegant` is not available any longer. Instead, the new style `RoundCorners` is introduced. The styles `BraceUp`, `BraceDown`, `BraceLeft`, and `BraceRight` are not available any longer. Instead, the macros `\psBraceUp`, `\psBraceDown`, `\psBraceLeft`, and `\psBraceRight` are defined. The option `pstadd` is not available any longer. If the package `pstricks-add` is loaded, the relevant styles are automatically defined. The macros `\RE`, `\IM`, `\sRE`, and `\sIM` are not available any longer because of their irrelevance to the objectives of the package.
- **Version 1.1 (04/01/2009):** Four new macros `\pshtick`, `\psvtick`, `\pshTick`, and `\psvTick` are added. The codes of macros `\psusampler` and `\psdsampler` are updated. However, there is no change in their user interface.
- **Version 1.0 (01/15/2009):** The first version of the package.

3 Macros

In this section, we introduce all the macros defined by the `pst-sigsys` package. Every macro has some optional keys that can be assigned either directly inside brackets right after the macro name or through the `\psset` macro provided by the `pstricks` package. In the syntax of every macro, the optional portions are identified by the shaded background. Unless directly stated, all coordinates specified by (*coor*) could be either in the cartesian format (x, y) or the polar format $(\rho; \theta)$ ¹. After the introduction of every macro, some examples are provided to illustrate the usage of that macro.

¹Recall that `pst-sigsys` activates the polar coordinates on loading. Hence, there is no need to use the `\SpecialCoor` macro.

3.1 \psaxeslabels

```
\psaxeslabels [keys] {arrows} (x_0,y_0)(x_1,y_1)(x_2,y_2){x-label}{y-label}
```

This macro is a simplified version of the `\psaxes` macro defined by the `pst-plot` package [?]. As depicted in Figure 1, the `\psaxeslabels` draws two straight lines, one vertical and one horizontal, that intersect at the point (x_0, y_0) . These lines are enclosed by a virtual rectangular box with the lower left corner at (x_1, y_1) and the upper right corners at (x_2, y_2) . The two lines are labeled by x -label and y -label, respectively. Similar to the `\psaxes` macro, the use of `arrows` is optional. The keys specific to the `\psaxeslabels` are summarized in Table 1.

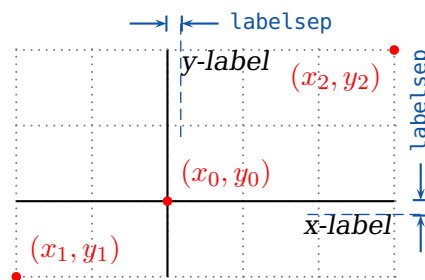
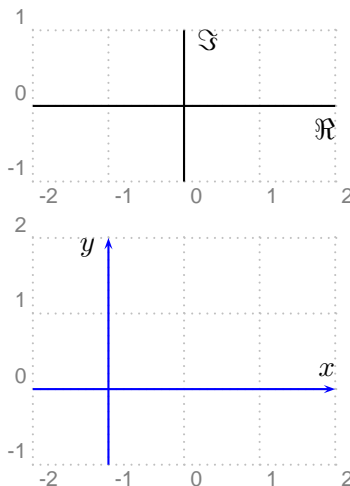


Figure 1. `\psaxeslabels` macro

Table 1. `\psaxeslabels` keys

Key	Value	Default	Description
<code>xlpos</code>	<code>t</code> <code>b</code>	<code>b</code>	Position of the x -label along the horizontal axis
<code>ylpos</code>	<code>l</code> <code>r</code>	<code>r</code>	Position of the y -label along the vertical axis



```
1 \begin{pspicture}[showgrid=true](-2,-1)(2,1)
2   \psaxeslabels(0,0)(-2,-1)(2,1){$\Re$}{$\Im$}
3 \end{pspicture}
```

```
1 \begin{pspicture}[showgrid=true](-2,-1)(2,2)
2   \psset{linecolor=blue,xlpos=t,ylpos=l}
3   \psaxeslabels{->}(-1,0)(-2,-1)(2,2){$x$}{$y$}
4 \end{pspicture}
```

3.2 \pstick

```
\pstick [keys] {angle} (coor){ticklength}
```

As depicted in Figure 2, the `\pstick` macro draws a straight line with length *ticklength* centered at *(coor)* and angled *angle* with respect to the horizontal axis. If the optional parameter *angle* is absent, then the angle is determined using the *angle* key. This macro could be used for adding tick lines to coordinate axes in addition to many other usages. The keys specific to the `\pstick` are summarized in Table 2.

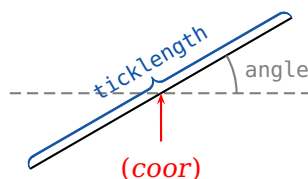
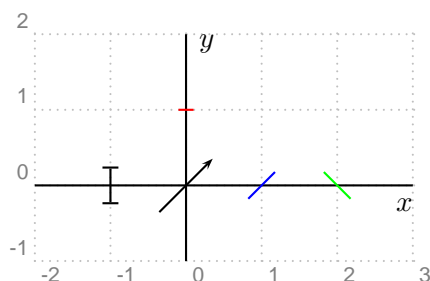


Figure 2. `\pstick` macro

Table 2. `\pstick` keys

Key	Value	Default	Description
<code>angle</code>	<i>num</i>	0	Tick angle



```

1 \begin{pspicture}[showgrid=true](-2,-1)(3,2)
2   \psaxeslabels(0,0)(-2,-1)(3,2){$x$}{$y$}
3   \pstick[linecolor=red](0,1){.2}
4   \pstick[arrows=| - |]{90}(-1,0){.5}
5   \psset{angle=45}
6   \pstick[arrows=->](0,0){1}
7   \pstick[linecolor=blue](1,0){.5}
8   \pstick[linecolor=green]{135}(2,0){.5}
9 \end{pspicture}

```

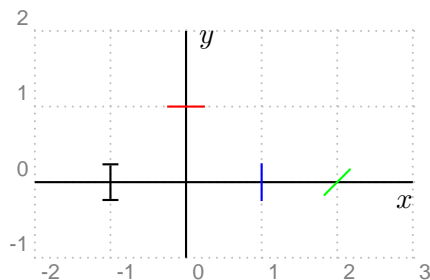
3.3 `\psTick`

`\psTick` [*keys*] {*angle*} (*coor*)

Similar to `\pstick`, the `\psTick` macro draws a straight line centered at *(coor)* and angled *angle* with respect to the horizontal axis. The only difference is that the tick length is specified by the *ticklength* key (Table 3). This macro is useful when multiple ticks are to be drawn all with the same length.

Table 3. `\psTick` keys

Key	Value	Default	Description
<code>ticklength</code>	<i>num</i> [dimen]	0.15	Tick length
<code>angle</code>	<i>num</i>	0	Tick angle



```

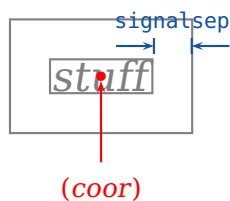
1 \begin{pspicture}[showgrid=true](-2,-1)(3,2)
2   \psaxeslabels(0,0)(-2,-1)(3,2){$x$}{$y$}
3   \psset{ticklength=.5}
4   \psTick[linecolor=red](0,1)
5   \psTick[arrows=| - |]{90}(-1,0)
6   \psTick[linecolor=blue]{90}(1,0)
7   \psset{angle=45}
8   \psTick[linecolor=green](2,0)
9 \end{pspicture}

```

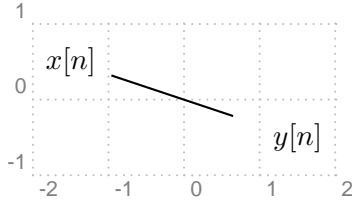
3.4 `\pssignal`

`\pssignal` [*keys*] (*coord*){*node*}{*stuff*}

This macro places *stuff* inside an invisible frame centered at (*coord*) and makes that a node labeled *node* (Figure 3). The separation of the frame and the *stuff* is determined by the key `signalsep` (Table 4).

**Figure 3.** `\pssignal` macro**Table 4.** `\pssignal` keys

Key	Value	Default	Description
<code>signalsep</code>	<i>num</i> [dimen]	5pt	Frame separation



```

1 \begin{pspicture}[showgrid=true](-2,-1)(2,1)
2   \psignal(-1.5,.5){x}{x[n]}
3   \psignal[signalsep=.5](1.5,-.5){y}{y[n]}
4   \ncline{x}{y}
5 \end{pspicture}

```

3.5 \psstem

```

\psstem[keys](x_0,\Delta){list}
\psstem[keys]{list}

```

The `\psstem` macro plots the sequence defined by *list* that is a comma-separated list of numbers. As shown in Figure 4a, if $list = n_1, n_2, n_3, \dots$, then `\psstem` draws vertical lines (stems) at $x_0, x_0 + \Delta, x_0 + 2\Delta, \dots$ on the horizontal axis with heights n_1, n_2, n_3, \dots , respectively. *It is important to remember that both x_0 and Δ must be integers.*² In case their values are not explicitly given, they are assumed $x_0 = 0$ and $\Delta = 1$. The stem ends are determined by the `stemhead` key. The `\psstem` macro is also capable of numerically tagging the stems. As depicted in Figure 4b, the tag of every stem is placed either below or above it depending on whether the corresponding number in the sequence is nonnegative (positive or zero) or negative, respectively. The distance of tags to stems is determined by the `labelsep` key. In some cases (e.g., when the `stemhead` is `>`), it is desirable to remove stems with zero heights. The key `killzero`, when used, removes such stems. The keys specific to the `\psstem` macro are summarized in Table 5.

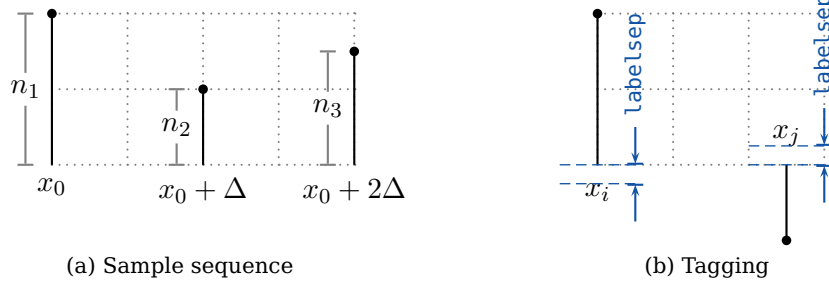
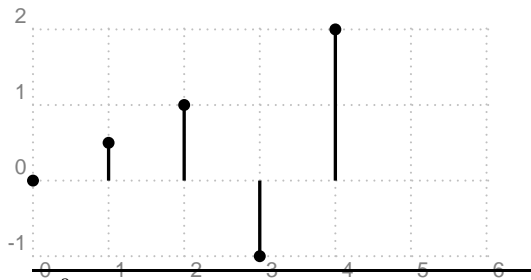


Figure 4. `\psstem` macro



```

1 \begin{pspicture}[showgrid=true](0,-1)(6,2)
2   \psstem[style=Stem]{0,.5,1,-1,2}
3 \end{pspicture}

```

²If you need to use non-integer values, then use the `xunit` key to arbitrarily choose any real value.

Table 5. `\psstem` keys

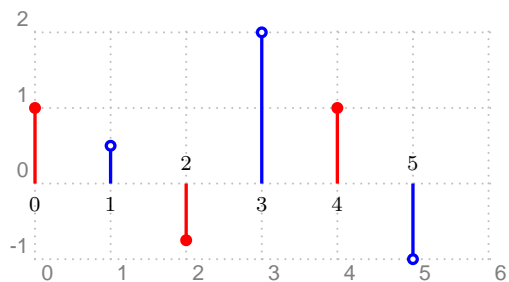
Key	Value	Default	Description
<code>stemhead</code>	<i>style</i>	<code>*</code>	Stem head. Possible choices are <code>*</code> , <code>o</code> , <code>></code> , <code><</code> , <code>>></code> , <code><<</code> , <code> </code> , <code>)</code> , <code>(</code> , <code>> </code> , and <code>< </code> .
<code>stemtag</code>	<i>Boolean</i>	<code>false</code>	Tagging the stems
<code>stemtagformat</code>	<i>format</i>	<code>\scriptstyle</code>	Tag format
<code>killzero</code>	<i>Boolean</i>	<code>false</code>	Removing zero-height stems



```

1 \begin{pspicture}[showgrid=true](0,-1)(6,2)
2   \psset{style=Stem,linecolor=blue,%
3     stemtagformat=\color{red}\scriptstyle}
4   \psstem[stemhead=>,stemtag](1,2){-1,1,2}
5 \end{pspicture}

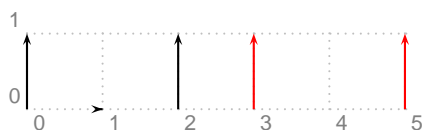
```



```

1 \begin{pspicture}[showgrid=true](0,-1)(6,2)
2   \psset{style=Stem,stemtag}
3   \psstem[linecolor=red](0,2){1,-.75,1}
4   \psset{stemhead=o}
5   \psstem[linecolor=blue](1,2){.5,2,-1}
6 \end{pspicture}

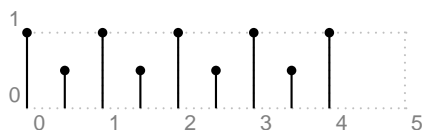
```



```

1 \begin{pspicture}[showgrid=true](5,1)
2   \psset{stemhead=>}
3   \psstem{1,0,1}
4   \psset{linecolor=red,killzero}
5   \psstem(3,1){1,0,1}
6 \end{pspicture}

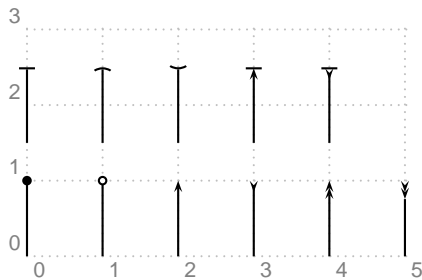
```



```

1 \begin{pspicture}[showgrid=true](5,1)
2   \psstem[xunit=.5]{1,.5,1,.5,1,.5,1,.5,1}
3 \end{pspicture}

```



```

1 \begin{pspicture}[showgrid=true](5,3)
2   \psstem[stemhead=*](0,1){1}
3   \psstem[stemhead=o](1,1){1}
4   \psstem[stemhead=>](2,1){1}
5   \psstem[stemhead=<](3,1){1}
6   \psstem[stemhead=>>](4,1){1}
7   \psstem[stemhead=<<](5,1){1}
8
9   \rput(0,1.5){%
10    \psstem[stemhead=|](0,1){1}
11    \psstem[stemhead=)](1,1){1}
12    \psstem[stemhead=(](2,1){1}
13    \psstem[stemhead=>|](3,1){1}
14    \psstem[stemhead=<|](4,1){1}
15  }
16 \end{pspicture}

```

3.6 \pszero

`\pszero` [*keys*] (*coord*){*node*}

This macro is used to generate a circle node centered at (*coord*) and labeled *node* that represents a zero of a system. It could also be used to generate several circles, all centered at (*coord*), representing high order zeros as shown in Figure 5. The radius of innermost circle is *zeroradius*, and it is incremented by *zeroradiusinc* for high order zeros. The line-width of all circles is determined by the *zerowidth* key. The key *order* determines the order of the zero. The key *scale* can be used to scale up or down the radius of the innermost circle, the radius increment, and the line-width of all circles. Table ?? summarizes keys corresponding to `\pszero` and their default values.

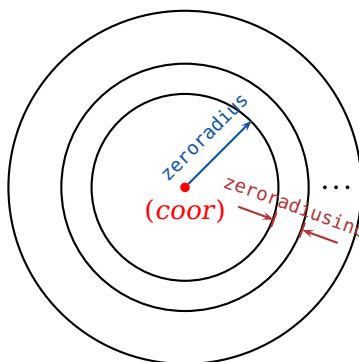


Figure 5. `\pszero` macro