The pst-sigsys Package

(version 1.1)

Farshid Delgosha
fdelgosha@gmail.com

April 1, 2009

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, and connecting a list of nodes using any node-connecting macro. I welcome 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 following command to the preamble of your document.

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

It loads pstricks [3], pst-node [4], and pst-xkey [1] packages. Moreover, it activates polar coordinates through the \SpecialCoor macro defined by the pstricks package. Hence, all macros support polar coordinates.

The pst-sigsys provides the following options.

- **notelegant:** When drawing block diagrams, I have found it more elegant to have round corner frames and line breaks. Hence, the pst-sigsys package sets the following PSTricks keys when loaded.

```
framesep=0.125
framearc=0.25
linearc=0.1
```

  To disable them, load the package with the notelegant option.

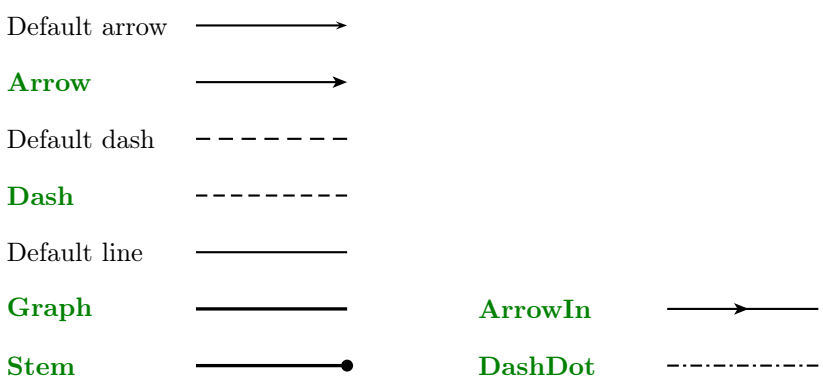
- **pstadd:** The pst-sigsys defines some PSTricks styles that can be used only with the pstricks-add package [2]. Use this option to define those styles that are introduced in Section 3.

2 What's New?

In version 1.1, 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.

3 Styles Defined by pst-sigsys

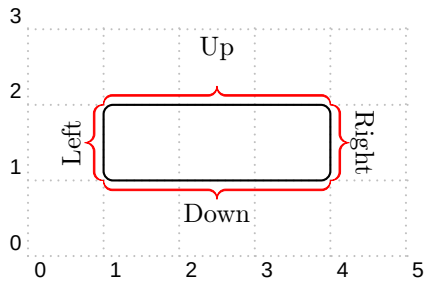
The pst-sigsys package defines a few useful PSTricks styles for drawling arrows and dashed lines as shown in Figure 1. (New styles are in green.) Some of these styles, which are shown in Figure 1b, can be used only with the pstricks-add package. The usage of these styles is shown in Section 6 with many examples.



(a) pstricks style **(b)** pstricks-add styles

Figure 1. New styles

In addition, the `pst-sigsys` package defines the four styles `BraceUp`, `BraceDown`, `BraceRight`, and `BraceLeft` in conjunction with the `psbrace` macro define by the `pstricks-add` package. In these styles, the distance of the text from the brace is controlled by the `labelsep` key. Note that the new assignment for the `labelsep` key must precede the usage of any one of the `Brace` styles for the distance to take effect. The usage of these styles is shown by the following example.




```
\begin{pspicture}[showgrid=true](5,3)
  \psframe(1,1)(4,2)
  \psset{linecolor=red}
  \psbrace*[labelsep=5mm,style=BraceUp]%
     (4,2)(1,2){Up}
  \psbrace*[style=BraceDown](1,1)(4,1){Down}
  \psbrace*[style=BraceRight](4,1)(4,2){Right}
  \psbrace*[style=BraceLeft](1,2)(1,1){Left}
\end{pspicture}
```

4 Simples Macros

The `pst-sigsys` package defines four macros `\RE`, `\sRE`, `\IM`, and `\sIM` that generate the symbols $\mathcal{R}$e, $\mathcal{R}_{\mathrm{e}}$, $\mathcal{I}$m, and $\mathcal{I}_{\mathrm{m}}$, respectively. (The small symbols are in script size.) These symbols can be used to refer to the real and imaginary parts of a complex number. All four macros can be used both inside and outside the math mode.

The real part of the complex number $c = a+jb$ is $a = \mathcal{R}\mathrm{e}(c)$ and its imaginary part is $b = \mathcal{I}\mathrm{m}(c)$.

```
The real part of the complex number $c=a+jb$
is $a = \RE(c)$ and its imaginary part
is $b = \IM(c)$.
```

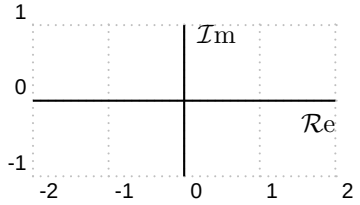
5 Graphical Macros

In this section, we introduce all the graphical macros defined by the `pst-sigsys` package. Every macro has some keys that can be assigned either directly inside optional brackets right after the macro name or through the `\psset` macro provided by the `pstricks` package. Unless directly stated, all coordinate inputs specified by *coor* could be either in cartesian form (x, y) or polar form $(\rho; \theta)$. (Recall that `pst-sigsys` activates the polar coordinates on loading. Hence, there is no need to use the `\SpecialCoor` macro.) After the introduction of every macro, some examples are provided to illustrate the usage of that macro.

5.1 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 [5]. As depicted in Figure 2, 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 and upper right corners at (x_1, y_1) and (x_2, y_2), respectively. The two lines are labeled by *x-label* and *y-label*, respectively. Similar to `\psaxes` macro, the use of *arrows* is optional. The keys employed by the `\psaxeslabels` are summarized in Table 1.




```
\begin{pspicture}[showgrid=true](-2,-1)(2,1)
  \psaxeslabels(0,0)(-2,-1)(2,1){\RE}{\IM}
\end{pspicture}
```

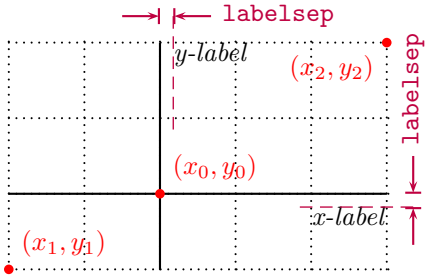
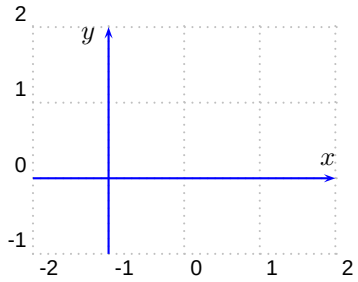


Figure 2. `\psaxeslabels` macro

Table 1. `\psaxeslabels` Keys

Key	*Value*	*Default*	*Description*
`xlpos`	`t | b`	`b`	Position of the x-label along the horizontal axis
`ylpos`	`l | r`	`r`	Position of the y-label along the vertical axis




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

5.2 pshtick

`\pshtick`[*keys*](*coor*){*ticklength*}

As depicted in Figure 3, the `\pshtick` macro draws a horizontal line centered at *coor* with length 2*ticklength*. This could be used for adding a tick line to coordinate axes.

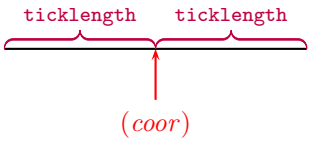
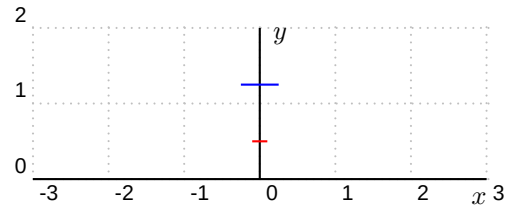


Figure 3. `\pshtick` macro




```
\begin{pspicture}[showgrid=true](-3,0)(3,2)
   \psaxeslabels(0,0)(-3,0)(3,2){$x$}{$y$}
   \pshtick[linecolor=red](0,.5){.1}
   \pshtick[linecolor=blue](0,1.25){.25}
\end{pspicture}
```

5.3 psvtick

`\psvtick`[*keys*](*coor*){*ticklength*}

Similar to the previous macro, the \psvtick macro draws a vertical line centered at *coor* with length 2*ticklength* (Figure 4). This could be used for adding a tick line to coordinate axes.

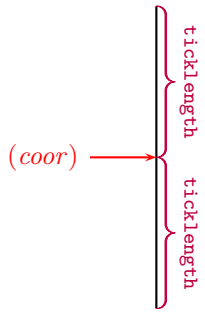
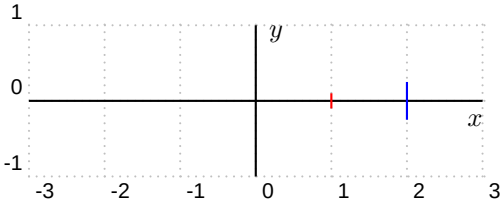


Figure 4. \psvtick macro

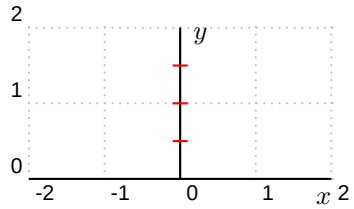



```
\begin{pspicture}[showgrid=true](-3,-1)(3,1)
   \psaxeslabels(0,0)(-3,-1)(3,1){$x$}{$y$}
   \psvtick[linecolor=red](1,0){.1}
   \psvtick[linecolor=blue](2,0){.25}
\end{pspicture}
```

5.4 pshTick

\pshTick[*keys*](*coor*)

Similar to \pshtick, the \pshTick macro draws a horizontal line centered at *coor*. The only difference is that the tick length is specified by the `ticklength` key. This is useful when multiple ticks are drawn all with the same length.

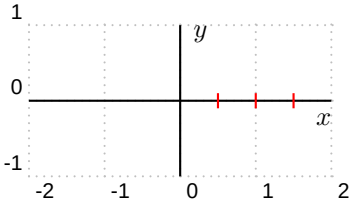



```
\begin{pspicture}[showgrid=true](-2,0)(2,2)
   \psaxeslabels(0,0)(-2,0)(2,2){$x$}{$y$}
   \psset{ticklength=.1,linecolor=red}
   \pshTick(0,.5)
   \pshTick(0,1)
   \pshTick(0,1.5)
\end{pspicture}
```

5.5 psvTick

\psvTick[*keys*](*coor*)

Similar to \psvtick, the \psvTick macro draws a vertical line centered at *coor*. The only difference is that the tick length is specified by the `ticklength` key. This is useful when multiple ticks are drawn all with the same length.

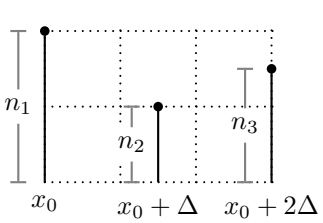



```
\begin{pspicture}[showgrid=true](-2,-1)(2,1)
   \psaxeslabels(0,0)(-2,-1)(2,1){$x$}{$y$}
   \psset{ticklength=.1,linecolor=red}
   \psvTick(.5,0)
   \psvTick(1,0)
   \psvTick(1.5,0)
\end{pspicture}
```

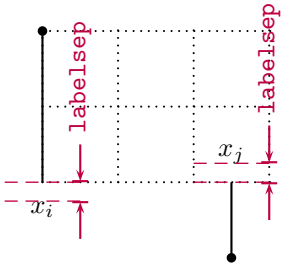
5.6 psstem

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

The \psstem macro plots the sequence defined by *list* that is a comma-separated list of real or integer numbers. As shown in Figure 5a, if *list* = $n_1, n_2, n_3, \ldots$, then \psstem draws vertical lines (stems) at $x_0, x_0 + \Delta, x_0 + 2\Delta, \ldots$ on the horizontal axis ($y = 0$) with heights $n_1, n_2, n_3, \ldots$, respectively. *Note that both* x_0 *and* Δ *must be integers.* In case their values are not explicitly given, they are assumed $x_0 = 0$ and $\Delta = 1$. The \psstem macro is also capable of numerically tagging the stems. As depicted in Figure 5b, the tag of every stem is place 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. The keys employed by the \psstem macro are summarized in Table 2.



(a) Sample sequence

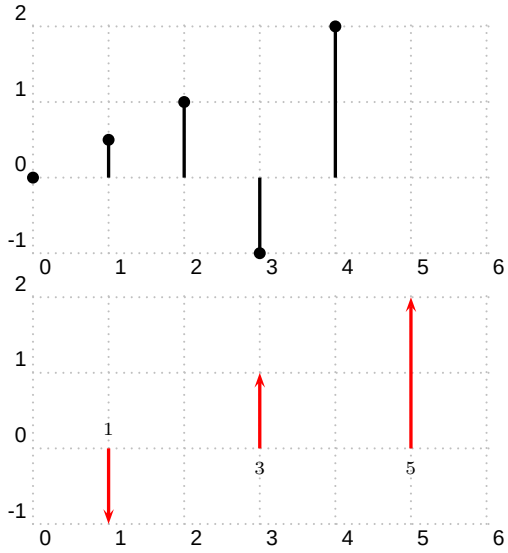


(b) Tagging

Figure 5. \psstem macro

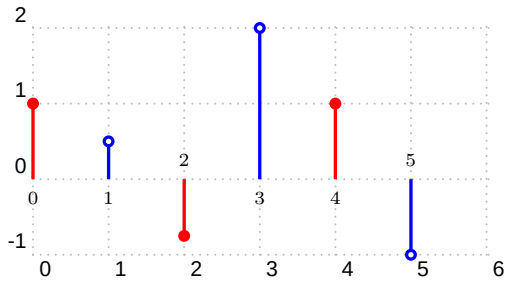
Table 2. \psstem Keys

Key	*Value*	*Default*	*Description*
stemhead	*style*	*	Stem head. Possible choices are *, o, >, <, >>, <<, |,), (, >|, and <|.
stemtag	*Boolean*	false	Tagging the stems
stemtagformat	*format*	\scriptstyle	Tag format

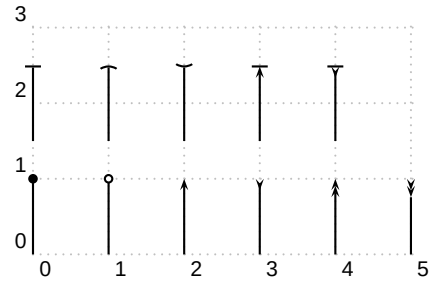



```
\begin{pspicture}[showgrid=true](0,-1)(6,2)
  \psstem[style=Stem]{0,.5,1,-1,2}
\end{pspicture}
```

```
\begin{pspicture}[showgrid=true](0,-1)(6,2)
  \psset{style=Stem,linecolor=red}
  \psstem[stemhead=>,stemtag](1,2){-1,1,2}
\end{pspicture}
```




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




```
\begin{pspicture}[showgrid=true](5,3)
  \psstem[stemhead=*](0,1){1}
  \psstem[stemhead=o](1,1){1}
  \psstem[stemhead=>](2,1){1}
  \psstem[stemhead=<](3,1){1}
  \psstem[stemhead=>>](4,1){1}
  \psstem[stemhead=<<](5,1){1}

  \rput(0,1.5){%
    \psstem[stemhead=|](0,1){1}
    \psstem[stemhead=)](1,1){1}
    \psstem[stemhead=(](2,1){1}
    \psstem[stemhead=>|](3,1){1}
    \psstem[stemhead=<|](4,1){1}
  }
\end{pspicture}
```

5.7 pszero

\pszero[*keys*](*coor*){*node*}

This macro is used to generate a circle node centered at *coor* and labeled *node* that represents a zero of a system. It could also be used to generate several circles, all centered at *coor*, representing high order zeros. As shown in Figure 6, 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`

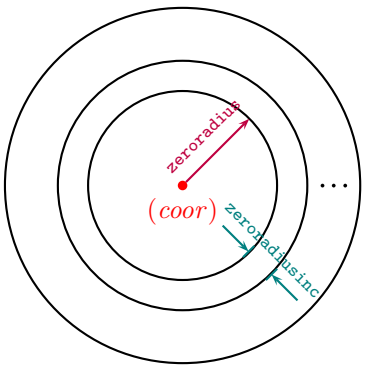
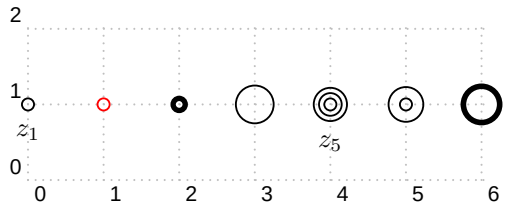


Figure 6. \pszero macro

determines the order of the zero. The key `scale` can be used to scale up or down the radius of innermost circle (`zeroradius`), the radius increment (`zeroradiusinc`), and the line-width of all circles (`zerowidth`). Table 3 summarizes keys corresponding to \pszero and their default values.

Table 3. `\pszero` Keys

Key	*Value*	*Default*	*Description*
`zerowidth`	*num/dimen/*	0.7pt	Line-width of all circles
`zeroradius`	*num/dimen/*	0.08	Radius of the innermost circle
`zeroradiusinc`	*num/dimen/*	0.07	Radius increment
`order`	*int*	1	Order of the zero
`scale`	*num*	1	Scale factor




```
\begin{pspicture}[showgrid=true](6,2)
  \pszero(0,1){z1}   \nput{-90}{z1}{$z_1$}
  \pszero[linecolor=red](1,1){z2}
  \pszero[zerowidth=2pt](2,1){z3}
  \pszero[zeroradius=.25](3,1){z4}
  \pszero[order=3](4,1){z5}
    \nput{-90}{z5}{$z_5$}
  \pszero[zeroradiusinc=.15,order=2](5,1){z6}
  \pszero[scale=3](6,1){z7}
\end{pspicture}
```

5.8 pspole

`\pspole`[*keys*](*coor*){*node*}

This macro is used to generate a cross node, as shown in Figure 7, centered at *coor* and labeled *node* that represents the pole of a system. The key `scale` can be used to scale up or down the pole line-width (`polewidth`) and pole length (`polelength`). The keys corresponding to the `\pspole` macro are summarized in Table 4.

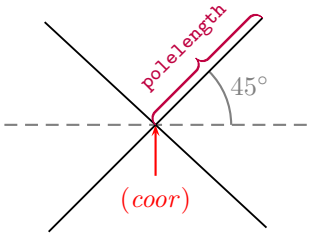
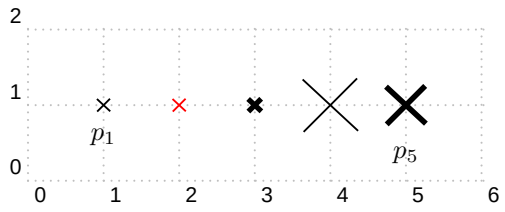


Figure 7. `\pspole` macro

Table 4. `\pspole` Keys

Key	*Value*	*Default*	*Description*
`polewidth`	*num/dimen/*	0.7pt	Cross line-width
`polelength`	*num/dimen/*	0.12	Cross length
`scale`	*num*	1	Scale factor




```
\begin{pspicture}[showgrid=true](6,2)
  \pspole(1,1){p1}   \nput{-90}{p1}{$p_1$}
  \pspole[linecolor=red](2,1){p2}
  \pspole[polewidth=2pt](3,1){p3}
  \pspole[polelength=.5](4,1){p4}
  \pspole[scale=3](5,1){p5}
    \nput{-90}{p5}{$p_5$}
\end{pspicture}
```

5.9 pscircleop

\pscircleop[*keys*](*coor*){*node*}

This macro draws a cross inside a circle that are both centered at *coor* and labeled *node* as shown in Figure 8. The length of the cross and its line-width are controlled by the `oplength` and `opwidth` keys, respectively. Note that the line-width of the enclosing circle is separately controlled by the `linewidth` key. The distance between the circle and the cross is determined by the key `opsep`. The type of the operation (whether plus or times) is controlled by the key `operation`. Another way of determining the operation inside the circle is through the key `angle` that determines the angle of the cross. The key `scale` can be used to scale up or down the cross line-width (`opwidth`), the cross length (`oplength`), the separation between the cross and the circle (`opsep`), and the circle line-width (`linewidth`). The keys corresponding to the `\pscircleop` macro are summarized in Table 5.

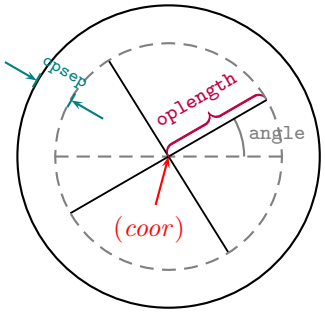
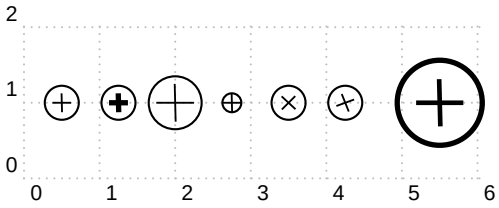


Figure 8. `\pscircleop` macro

Table 5. `\pscircleop` Keys

Key	*Value*	*Default*	*Description*
`opwidth`	*num[dimen]*	0.7pt	Cross line-width
`oplength`	*num[dimen]*	0.125	Cross length
`opsep`	*num[dimen]*	0.1	Separation between the cross and the frame
`operation`	`plus|times`	`plus`	Operation
`angle`	*angle*	0	Cross angle
`scale`	*num*	1	Scale factor

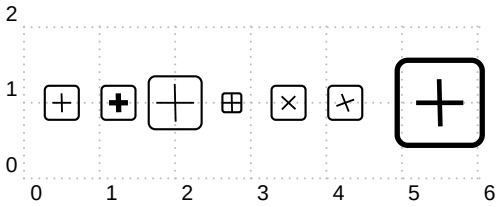



```
\begin{pspicture}[showgrid=true](6,2)
  \pscircleop(.5,1){op1}
  \pscircleop[opwidth=2pt](1.25,1){op2}
  \pscircleop[oplength=.25](2,1){op3}
  \pscircleop[opsep=0](2.75,1){op4}
  \pscircleop[operation=times](3.5,1){op5}
  \pscircleop[angle=20](4.25,1){op6}
  \pscircleop[scale=2.5](5.5,1){op7}
\end{pspicture}
```

5.10 psframeop

\psframeop[*keys*](*coor*){*node*}

This macro is very similar to the \pscircleop macro. The only difference is that the operation is enclosed inside a square frame rather than a circular one.




```
\begin{pspicture}[showgrid=true](6,2)
  \psframeop(.5,1){op1}
  \psframeop[opwidth=2pt](1.25,1){op2}
  \psframeop[oplength=.25](2,1){op3}
  \psframeop[opsep=0](2.75,1){op4}
  \psframeop[operation=times](3.5,1){op5}
  \psframeop[angle=20](4.25,1){op6}
  \psframeop[scale=2.5](5.5,1){op7}
\end{pspicture}
```

5.11 psdisk

\psdisk[*keys*](*coor*){*radius*}

It draws a solid disk centered at *coor* with radius *radius* as depicted in Figure 9. The fill color is specified by the `fillcolor` key. This macro is used to shade the region of convergence of a system in the z plane.

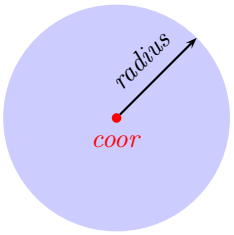
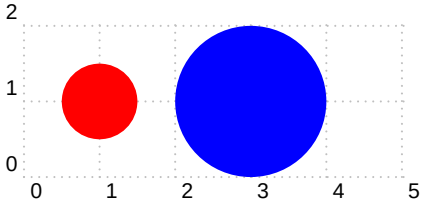


Figure 9. \psdisk macro




```
\begin{pspicture}[showgrid=true](5,2)
  \psdisk[fillcolor=red](1,1){.5}
  \psdisk[fillcolor=blue](3,1){1}
\end{pspicture}
```

5.12 psring

\psring[*keys*](*coor*){*inner-radius*}{*outer-radius*}

This macro draws a solid ring centered at *coor* with inner radius *inner-radius* and outer radius *outer-radius* as shown in Figure 10. The fill color is specified by the `fillcolor` key. This macro is used to shade the region of convergence of a system in the z plane.

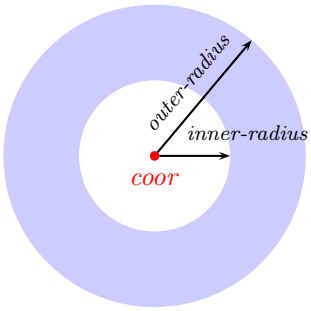
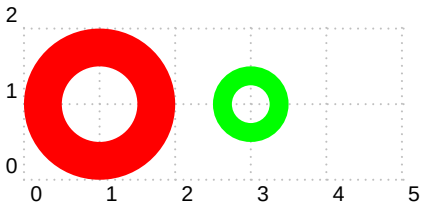


Figure 10. `\psring` macro




```
\begin{pspicture}[showgrid=true](5,2)
  \psring[fillcolor=red](1,1){.5}{1}
  \psring[fillcolor=green](3,1){.25}{.5}
\end{pspicture}
```

5.13 psdiskc

\psdiskc[*keys*](*coor*)(x_0, y_0){*radius*}

As shown in Figure 11, this macro shades the area confined between a circle centered at *coor* with radius *radius* and a rectangle centered at *coor* and side lengths $2x_0$ and $2y_0$. The fill color is specified by the `fillcolor` key. This macro is used to shade the region of convergence of a system in the z plane.

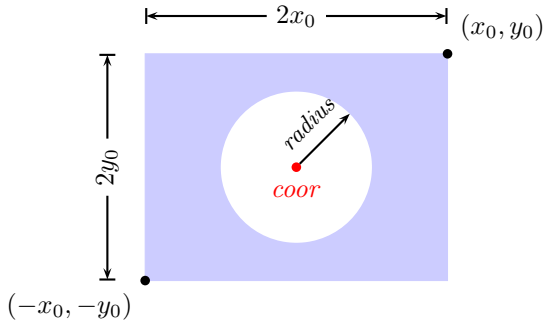
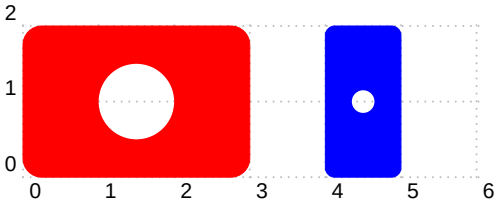


Figure 11. `\psdiskc` macro




```
\begin{pspicture}[showgrid=true](6,2)
  \psdiskc[fillcolor=red](1.5,1)(1.5,1){.5}
  \psdiskc[fillcolor=blue](4.5,1)(.5,1){.15}
\end{pspicture}
```

5.14 psldots

\psldots[*keys*]

As depicted in Figure 12, this macro draws three dots each with diameter `ldotssize` on the same straight line. Every two consecutive dots are separated by `ldotssep`. The angle of the line on which the dots lie with the horizontal axis is controlled by the key `angle`. The key `scale` can be used to scale up or down the dot diameter (`ldotssize`) and the dot separation (`ldotssep`). The keys corresponding to `\pslodots` are summarized in Table 6.

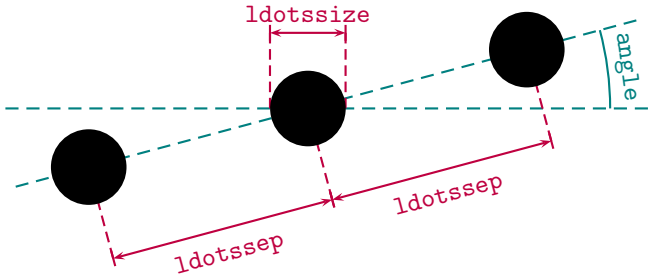
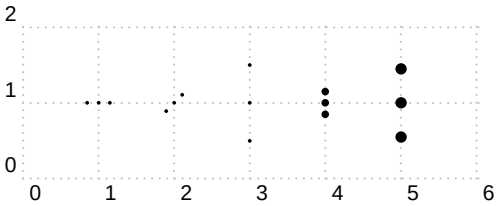


Figure 12. `\psldots` macro

Table 6. `\psldots` Keys

Key	*Value*	*Default*	*Description*
`ldotssize`	*num/dimen/*	0.05	Dot diameter
`ldotssep`	*num/dimen/*	0.15	Distance between consecutive dots
`angle`	*angle*	0	Dots angle
`scale`	*num*	1	Scale factor

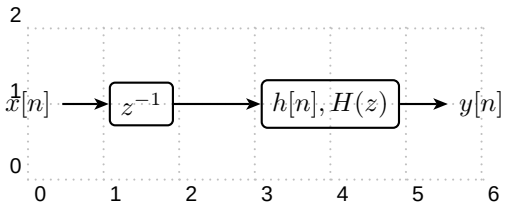



```
\begin{pspicture}[showgrid=true](6,2)
  \rput(1,1){\psldots}
  \rput(2,1){\psldots[angle=45]}
  \rput(4,1){\psldots[angle=90,ldotssize=.1]}
  \rput(3,1){\psldots[angle=90,ldotssep=.5]}
  \rput(5,1){\psldots[angle=90,scale=3]}
\end{pspicture}
```

5.15 psblock

\psblock[*keys*](*coor*){*node*}{*stuff*}

This macro places *stuff* at coordinate *coor*, encloses it in a rectangular frame, and turns that into a node labeled *node*. The separation between the *stuff* and the frame is controlled by the `framesep` key.

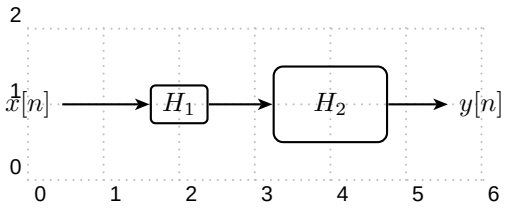



```
\begin{pspicture}[showgrid=true](6,2)
  \rput(0,1){\rnode{x}{$x[n]$}}
  \psblock(1.5,1){a}{$z^{-1}$}
  \psblock(4,1){b}{$h[n], H(z)$}
  \rput(6,1){\rnode{y}{$y[n]$}}
  %-----------------
  \psset{style=Arrow}
  \ncline[nodesepA=.15]{x}{a}
  \ncline{a}{b}
  \ncline[nodesepB=.15]{b}{y}
\end{pspicture}
```

5.16 psfblock

`\psfblock`[*keys*](*coor*){*node*}{*stuff*}

This macro is very similar to the `\psblock` macro except that the size of the frame is controlled by the key `framesize`. The frame size is specified as `framesize=`*num1[dimen] num2[dimen]*. Note that *num1* and *num2* are separated by a space, not by comma. If *num2* is absent, then a square frame is created.

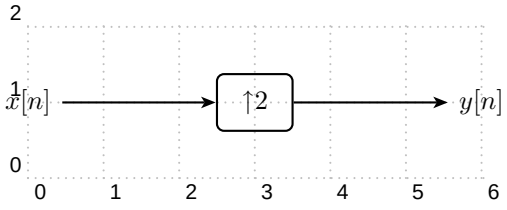



```
\begin{pspicture}[showgrid=true](6,2)
  \rput(0,1){\rnode{x}{$x[n]$}}
  \psfblock[framesize=.75 .5](2,1){a}{$H_1$}
  \psfblock[framesize=1.5 1](4,1){b}{$H_2$}
  \rput(6,1){\rnode{y}{$y[n]$}}
  %-----------------
  \psset{style=Arrow}
  \ncline[nodesepA=.15]{x}{a}
  \ncline{a}{b}
  \ncline[nodesepB=.15]{b}{y}
\end{pspicture}
```

5.17 psusampler

`\psusampler`[*keys*](*coor*){*node*}{*stuff*}

This macro is similar to the `\psfblock` except that *stuff* is placed next to an up-arrow in math mode representing an up-sampler. *Notice that stuff must be in text mode, not in the math mode, i.e., do not put $ around stuff.*

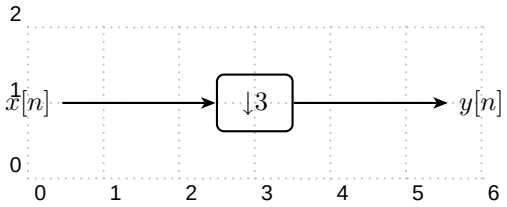



```
\begin{pspicture}[showgrid=true](6,2)
  \rput(0,1){\rnode{x}{$x[n]$}}
  \psusampler[framesize=1 .75](3,1){a}{2}
  \rput(6,1){\rnode{y}{$y[n]$}}
  %-----------------
  \psset{style=Arrow}
  \ncline[nodesepA=.15]{x}{a}
  \ncline[nodesepB=.15]{a}{y}
\end{pspicture}
```

5.18 psdsampler

\psdsampler[*keys*](*coor*){*node*}{*stuff*}

This macro is similar to the \psfblock except that *stuff* is placed next to a down-arrow in math mode representing a down-sampler. *Notice that stuff must be in text mode, not in the math mode, i.e., do not put $ around stuff.*

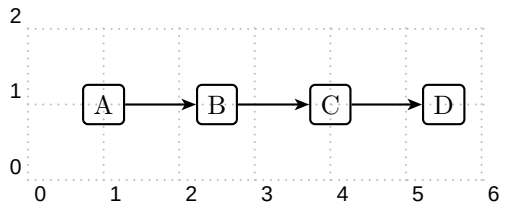



```
\begin{pspicture}[showgrid=true](6,2)
  \rput(0,1){\rnode{x}{$x[n]$}}
  \psdsampler[framesize=1 .75](3,1){a}{3}
  \rput(6,1){\rnode{y}{$y[n]$}}
  %-----------------
  \psset{style=Arrow}
  \ncline[nodesepA=.15]{x}{a}
  \ncline[nodesepB=.15]{a}{y}
\end{pspicture}
```

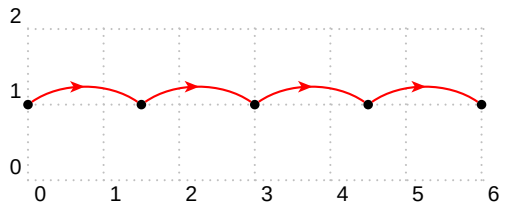
5.19 nclist

\nclist[*keys*]{*nc-macro*}{*node-list*}

This macro is very useful when connecting several nodes using a single node-connecting macro. The *node-list* must be a comma-separated list of nodes. If *node-list* is $n_1, n_2, n_3, \ldots$ is a list of nodes, then \nclist connects n_1 to n_2, n_2 to n_3, and so forth all using the macro *nc-macro*.




```
\begin{pspicture}[showgrid=true](6,2)
  \psblock(1,1){a}{A}
  \psblock(2.5,1){b}{B}
  \psblock(4,1){c}{C}
  \psblock(5.5,1){d}{D}
  \nclist[style=Arrow]{ncline}{a,b,c,d}
\end{pspicture}
```



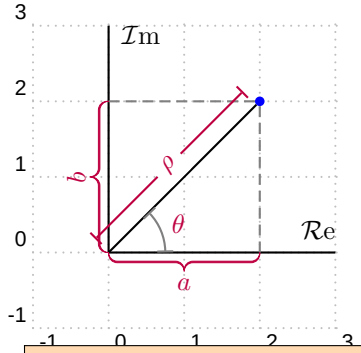

```
\begin{pspicture}[showgrid=true](6,2)
  \dotnode(0,1){a}
  \dotnode(1.5,1){b}
  \dotnode(3,1){c}
  \dotnode(4.5,1){d}
  \dotnode(6,1){e}
  \psset{style=ArrowIn,arcangle=35}
  \psset{linecolor=red}
  \nclist{ncarc}{a,b,c,d,e}
\end{pspicture}
```

6 Examples

In this section, we provide some examples to illustrate the benefits and usage of macros defined in Section 5. Note that some of these examples require the use of additional packages. In that case, additional packages are mentioned next to the example number.

6.1 Complex Number

Example 1.(use `pstricks-add`) Show the complex number $c = a + jb = \rho e^{j\theta}$ as a point in the complex plane.



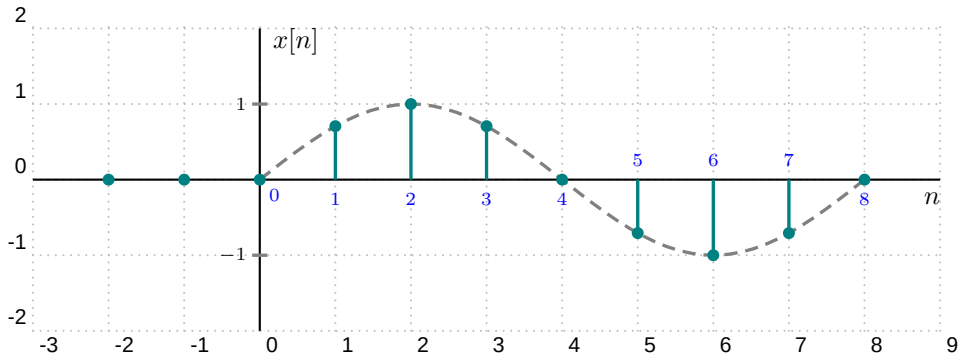

```
\begin{pspicture}[showgrid=true](-1,-1)(3,3)
  %--- Drawing axes ---
  \psaxeslabels[xlpos=t](0,0)(0,0)(3,3){\RE}{\IM}

  %--- Defining some useful nodes ---
  \dotnode[linecolor=blue](2,2){c}
  \pnode(0,0){org}
  \pnode(2,0){a}
  \pnode(0,2){b}

  %--- Connecting nodes ---
  \ncline{org}{c}
  \ncline[linecolor=gray,style=Dash]{c}{a}
  \ncline[linecolor=gray,style=Dash]{c}{b}

  %--- Labeling ---
  \color{purple}
  \psset{linecolor=purple,arrows=|-|,nrot=:U}
  \psbrace*[style=BraceDown](org)(a){$a$}
  \psbrace*[style=BraceLeft](b)(org){$b$}
  \ncline[offset=.25]{org}{c} \ncput*{$\rho$}
  \psarc[linecolor=gray](org){.75}{0}{45}
  \rput(1;22.5){$\theta$}
\end{pspicture}
```

6.2 Plotting

Example 2. (use `pst-plot`) Consider the continuous-time signal

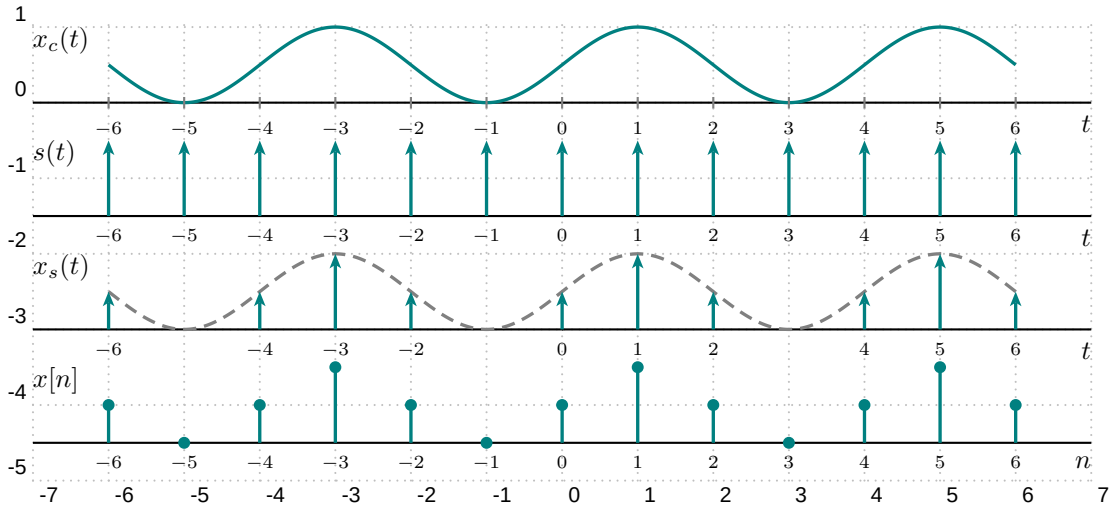
$$x_c(t) = \begin{cases} \sin(t) \ , & t \geq 0 \\ 0 \ , & t < 0 \ . \end{cases}$$

Draw the sampled sequence $x[n] = x_c(\pi n/4)$.




```
\begin{pspicture}[showgrid=true](-3,-2)(9,2)
  %--- Drawing axes ---
  \psaxeslabels(0,0)(-3,-2)(9,2){$n$}{$x[n]$}

  %--- x_c(t) ---
  \psplot[style=Graph,linecolor=gray,linestyle=dashed]{0}{8}{x 45 mul sin}

  %--- x[n] ---
  \psset{style=Stem,linecolor=teal,stemtagformat={\color{blue}\scriptstyle}}
  \psstem(-2,1){0,0,0}
  \psstem[stemtag](1,1){.707107,1,.707107,0,-.707107,-1,-.707107,0}

  %--- Labeling the origin ---
  \uput[-45](0,0){$\color{blue}\scriptstyle 0$}

  %--- Horizontal ticks ----
  \pshtick[linecolor=gray](0,1){.1}
  \pshtick[linecolor=gray](0,-1){.1}
  \uput[180](0,1){$\scriptstyle 1$}
  \uput[180](0,-1){$\scriptstyle -1$}
\end{pspicture}
```

Example 3. (use `pst-plot`) Consider the process of sampling a continuous-time signal $x_c(t)$ with period T as follows: (1) Multiply $x_c(t)$ by the impulse train $s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$ to obtain $x_s(t) = x_c(t)s(t)$, and (2) Convert every delta in $x_s(t)$ into a sequence to obtain the sampled sequence $x[n]$. Demonstrate this process for the continuous-time signal $x_c(t) = 0.5\sin(\pi t/2) + 0.5$ and $T = 1$.

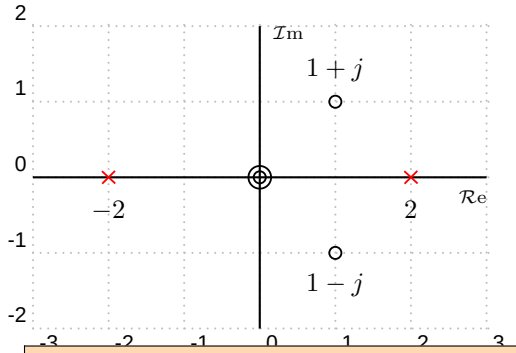



```
\begin{pspicture}[showgrid=true](-7,-5)(7,1)
  %--- Some settings and definitions ---
  \psset{plotpoints=500,ylpos=l,stemtag}
  \def\plotsin[#1]{\psplot[#1]{-6}{6}{x 90 mul sin .5 mul .5 add}}

  %--- x_c(t) ---
  \psaxeslabels(0,0)(-7,0)(7,0){$t$}{}
  \rput[tl](-7,1){$x_c(t)$}
  \plotsin[style=Graph,linecolor=teal]
  \multips(-6,0)(1,0){13}{\psline[linecolor=gray](0,-.075)(0,.075)}
  \multido{\nn=-6+1}{13}{\rput[t](\nn,-.25){$\scriptstyle\nn$}}

  %--- s(t) ----
  \rput(0,-1.5){%
    \psaxeslabels(0,0)(-7,0)(7,0){$t$}{}
    \rput[tl](-7,1){$s(t)$}
    \psset{style=Stem,style=Arrow,stemhead=>,linecolor=teal}
    \psstem(-6,1){1,1,1,1,1,1,1,1,1,1,1,1,1}}

  %--- x_s(t) ---
  \rput(0,-3){%
    \psaxeslabels(0,0)(-7,0)(7,0){$t$}{}
    \rput[tl](-7,1){$x_s(t)$}
    \plotsin[style=Graph,linecolor=gray,linestyle=dashed]
    \psset{style=Stem,style=Arrow,stemhead=>,linecolor=teal}
    \psstem(-6,1){.5}  \psstem(-4,1){.5,1,.5}
    \psstem{.5,1,.5}   \psstem(4,1){.5,1,.5}}

  %--- x[n] ----
  \rput(0,-4.5){%
    \psaxeslabels(0,0)(-7,0)(7,0){$n$}{}
    \rput[tl](-7,1){$x[n]$}
    \psset{style=Stem,style=Arrow,linecolor=teal}
    \psstem(-6,1){.5,0,.5,1,.5,0,.5,1,.5,0,.5,1,.5}}
\end{pspicture}
```

6.3 Pole-Zero Diagram

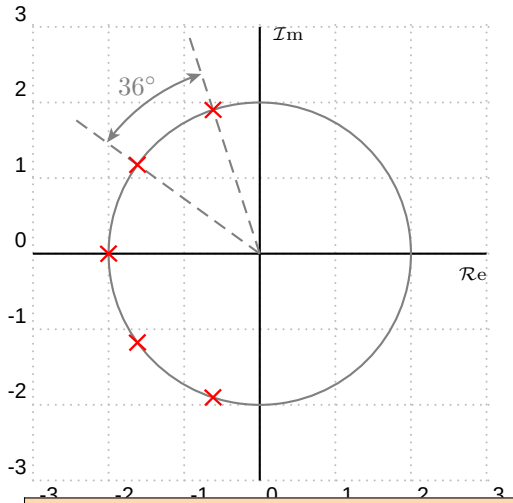
Example 4. Draw the pole-zero diagram of a system with the following system function.

$$H(z) = \frac{z^4 - 2z^3 + 2z^2}{z^2 - 4}$$




```
\begin{pspicture}[showgrid=true](-3,-2)(3,2)
  %--- Drawing axes ---
  \psaxeslabels(0,0)(-3,-2)(3,2){$\sRE$}{$\sIM$}

  %--- Marking zeros ---
  \pszero[order=2](0,0){z1}
  \pszero(1,1){z2}   \nput{90}{z2}{$1 + j$}
  \pszero(1,-1){z3}  \nput{-90}{z3}{$1 - j$}

  %--- Marking poles ---
  \psset{linecolor=red}
  \pspole(2,0){p1}   \nput{-90}{p1}{$2$}
  \pspole(-2,0){p2}  \nput{-90}{p2}{$-2$}
\end{pspicture}
```

Example 5. (use `multido`) Draw the pole-zero diagram of a fifth-order Butterworth filter.



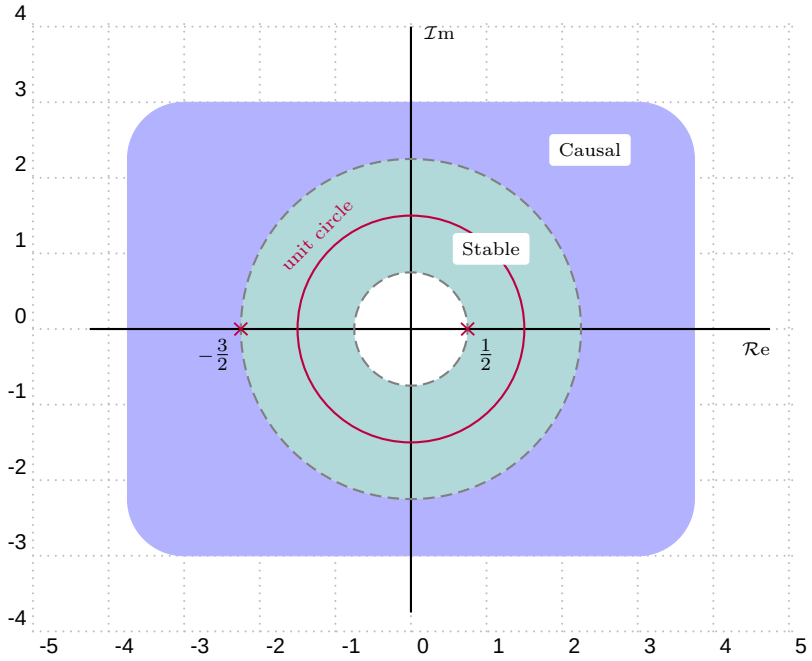

```
\begin{pspicture}[showgrid=true](-3,-3)(3,3)
  %--- Drawing axes ---
  \psaxeslabels(0,0)(-3,-3)(3,3){$\sRE$}{$\sIM$}
  \pscircle[linecolor=gray](0,0){2}

  %--- Angle between poles ---
  \psset{linecolor=gray}
  \psline[linestyle=dashed](3;108)
  \psline[linestyle=dashed](3;144)
  \psarc[style=Arrow,arrows=<->](0,0){2.5}{108}{144}
  \rput(2.75;126){\textcolor{gray}{$36^\circ$}}

  %--- Placing poles ---
  \psset{linecolor=red,scale=1.25}
  \multido{\np=108+36}{5}{\pspole(2;\np){p}}
\end{pspicture}
```

6.4 Region of Convergence

Example 6. Shade the region of convergence (ROC) of a system with the following system function assuming it is: (1) causal, and (2) stable.

$$H(z) = \frac{1}{z^2 + z - \frac{3}{4}}$$

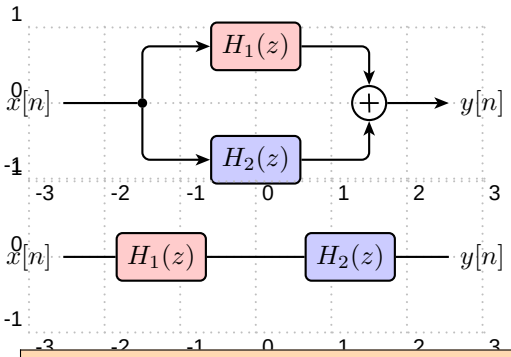
Since the poles of the system are at $z = \frac{1}{2}$ and $z = -\frac{3}{2}$, the ROC of the system with the given assumptions is as follows.




```
\begin{pspicture}[showgrid=true](-5,-4)(5,4)
  %--- Shading ROCs ---
  \psring[fillcolor=teal!30](0,0){.75}{2.25}
  \psdiskc[fillcolor=blue!30](0,0)(3.75,3){2.25}

  %--- Drawing axes ---
  \psaxeslabels(0,0)(-4.25,-3.75)(4.75,4){$\sRE$}{$\sIM$}

  %--- Marking poles ---
  \psset{linecolor=purple,labelsep=.05}
  \pscircle(0,0){1.5}
  \rput[b]{45}(1.68;135){{\scriptsize\color{purple}unit circle}}
  \pscircle[linecolor=gray,linestyle=dashed,style=Dash](0,0){.75}
  \pspole(.75,0){p1}    \nput{-45}{p1}{$\tfrac{1}{2}$}
  \pscircle[linecolor=gray,linestyle=dashed,style=Dash](0,0){2.25}
  \pspole(-2.25,0){p2}  \nput{225}{p2}{$\scriptstyle-\tfrac{3}{2}$}

  %--- Labeling the stable and causal ROCs ---
  \rput*(1.5;45){{\scriptsize Stable}}
  \rput*(3.35;45){{\scriptsize Causal}}
\end{pspicture}
```

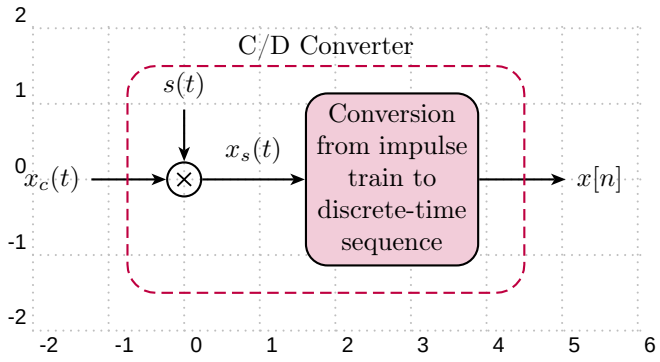
6.5 Block Diagrams

Example 7. Draw the block diagram of two systems $H_1(z)$ and $H_2(z)$ in both parallel and series combinations.




```
%============================
%   Parallel Combination
%============================
\begin{pspicture}[showgrid=true](-3,-1)(3,1)
  %--- Defining blocks ---
  \rput(-3,0){\rnode{x}{$x[n]$}}
  \dotnode(-1.5,0){dot}
  \psblock[fillstyle=solid,fillcolor=red!20](0,.75){H1}{$H_1(z)$}
  \psblock[fillstyle=solid,fillcolor=blue!20](0,-.75){H2}{$H_2(z)$}
  \pscircleop(1.5,0){oplus}
  \rput(3,0){\rnode{y}{$y[n]$}}

  %--- Connecting blocks ---
  \psset{style=Arrow}
  \ncline[nodesepA=.15]{-}{x}{dot}
  \ncangle[angleA=90,angleB=180]{dot}{H1}
  \ncangle[angleA=-90,angleB=180]{dot}{H2}
  \ncangle[angleB=90]{H1}{oplus}
  \ncangle[angleB=-90]{H2}{oplus}
  \ncline[nodesepB=.15]{oplus}{y}
\end{pspicture}
%
%========================
%   Series Combination
%========================
\begin{pspicture}[showgrid=true](-3,-1)(3,1)
  %--- Defining blocks ---
  \rput(-3,0){\rnode{x}{$x[n]$}}
  \psblock[fillstyle=solid,fillcolor=red!20](-1.25,0){H1}{$H_1(z)$}
  \psblock[fillstyle=solid,fillcolor=blue!20](1.25,0){H2}{$H_2(z)$}
  \rput(3,0){\rnode{y}{$y[n]$}}

  %--- Connecting blocks ---
  \ncline[nodesepA=.15]{x}{H1}
  \ncline{H1}{H2}
  \ncline[nodesepB=.15]{H2}{y}
\end{pspicture}
```

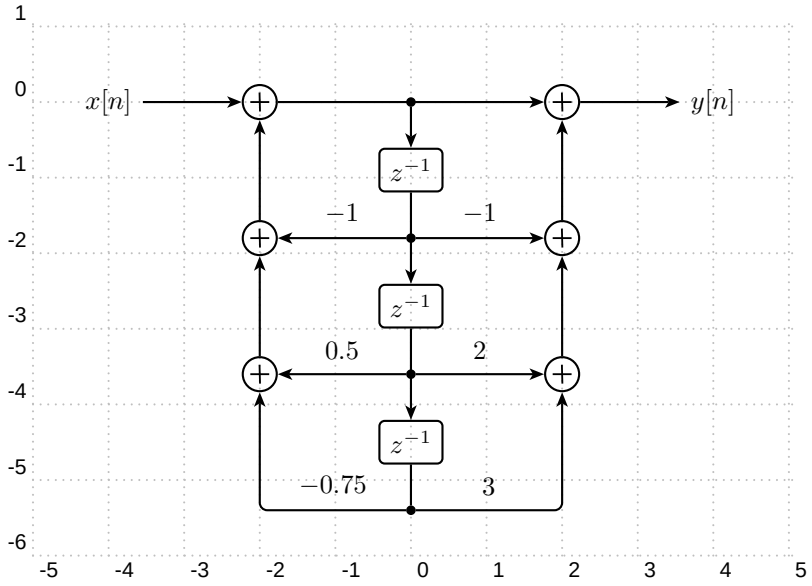
Example 8. Draw the block diagram of a continuous-to-discrete-time (C/D) converter.




```
\begin{pspicture}[showgrid=true](-2,-2)(5.5,2)
  %--- Defining blocks ---
  \rput(-1.75,0){\rnode{xc}{$x_c(t)$}}
  \pscircleop[operation=times](0,0){otimes}
  \rput(0,1.25){\rnode{s}{$s(t)$}}
  \psblock[fillstyle=solid,fillcolor=purple!20](2.75,0){conv}{\parbox[c]{2\psunit}%
  {\centering Conversion from impulse train to discrete-time sequence}}
  \rput(5.5,0){\rnode{x}{$x[n]$}}

  %--- Connecting blocks ---
  \psset{style=Arrow}
  \ncline[nodesepA=.15]{xc}{otimes}
  \ncline[nodesepA=.15]{s}{otimes}
  \ncline{otimes}{conv}   \naput{$x_s(t)$}
  \ncline[nodesepB=.15]{conv}{x}

  %--- Drawing the dashed frame ---
  \psframe[linecolor=purple,linestyle=dashed,style=Dash](-.75,-1.5)(4.5,1.5)
  \rput(1.875,1.75){C/D Converter}
\end{pspicture}
```

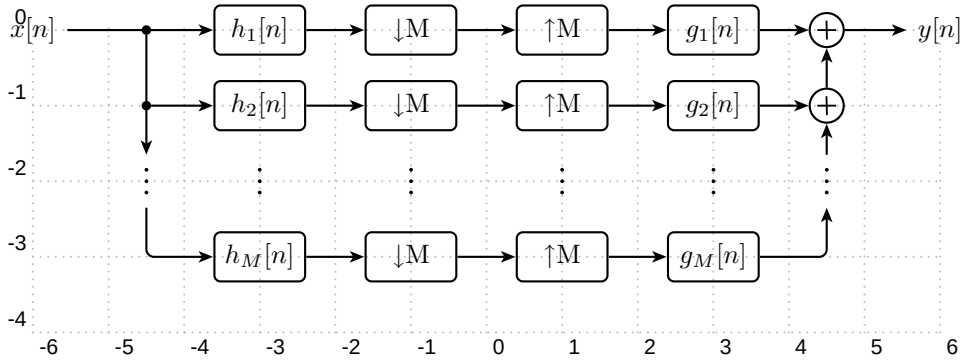
Example 9. (use `multido`) Draw the direct-form II block diagram of a discrete-time LTI system with the following system function.

$$H(z) = \frac{1 - z^{-1} + 2z^{-2} + 3z^{-3}}{1 + z^{-1} - 0.5z^{-2} + 0.75z^{-3}}$$




```
\begin{pspicture}[showgrid=true](-5,-6)(5,1)
  %--- Some settings ---
  \psset{style=Arrow}

  %--- Defining blocks ---
  \dotnode(0,0){dot1}
  \multido{\nA=1+1,\nB=2+1,\ryA=-.9+-1.8,\ryB=-1.8+-1.8}{3}{%
    \psblock(0,\ryA){D\nA}{$z^{-1}$}
    \dotnode(0,\ryB){dot\nB}}
  \multido{\nn=1+1,\ry=0+-1.8}{3}{\pscircleop(-2,\ry){oplusL\nn}}
  \multido{\nn=1+1,\ry=0+-1.8}{3}{\pscircleop(2,\ry){oplusR\nn}}
  \rput(-4,0){\rnode{x}{$x[n]$}}
  \rput(4,0){\rnode{y}{$y[n]$}}

  %--- Connecting blocks ---
  \psset{style=Arrow}
  \ncline[nodesepA=.15]{x}{oplusL1}
  \ncline{oplusL1}{oplusR1}
  \ncline[nodesepB=.15]{oplusR1}{y}
  \nclist{ncline}{dot1,D1,D2,D3}
  \ncline{-}{D3}{dot4}
  \nclist{ncline}{oplusL3,oplusL2,oplusL1}
  \nclist{ncline}{oplusR3,oplusR2,oplusR1}
  \ncline{dot2}{oplusL2}   \nbput{$-1$}
  \ncline{dot2}{oplusR2}   \naput{$-1$}
  \ncline{dot3}{oplusL3}   \nbput{$0.5$}
  \ncline{dot3}{oplusR3}   \naput{$2$}
  \ncangle[angleA=180,angleB=-90]{dot4}{oplusL3}   \nbput[npos=.5]{$-0.75$}
  \ncangle[angleB=-90]{dot4}{oplusR3}   \naput[npos=.5]{$3$}
\end{pspicture}
```

Example 10. (use `pstricks-add`) Draw the block diagram of an M-channel maximally decimated filter bank.




```
\begin{pspicture}[showgrid=true](-6,-4)(6,.5)
  \psset{framesize=1.2 .65}
  \rput(-6,0){\rnode{x}{$x[n]$}}
  \rput(6,0){\rnode{y}{$y[n]$}}
  \dotnode(-4.5,0){dot1}
  \dotnode(-4.5,-1){dot2}
  \newcount\cnt

  %--- First and second channels ---
  \cnt=0
  \psforeach{\ry}{0,-1}{%
    \advance\cnt by 1
    \psfblock(-3,\ry){h\the\cnt}{$h_{\the\cnt}[n]$}
    \psdsampler(-1,\ry){ds\the\cnt}{$M$}
    \psusampler(1,\ry){us\the\cnt}{$M$}
    \psfblock(3,\ry){g\the\cnt}{$g_{\the\cnt}[n]$}
    \pscircleop(4.5,\ry){oplus\the\cnt}}

  %--- Placing dots ---
  \cnt=0
  \psforeach{\rx}{-4.5,-3,-1,1,3,4.5}{%
    \advance\cnt by 1
    \rput(\rx,-2){\rnode{dots\the\cnt}{\psldots[angle=90]}}}

  %--- M-th channel ---
  \psfblock(-3,-3){hM}{$h_M[n]$}
  \psdsampler(-1,-3){dsM}{$M$}
  \psusampler(1,-3){usM}{$M$}
  \psfblock(3,-3){gM}{$g_M[n]$}

  %--- Connecting blocks ---
  \psset{style=Arrow}
  \ncline[nodesepA=.15]{x}{h1}
  \nclist{ncline}{h1,ds1,us1,g1,oplus1}
  \ncline[nodesepB=.15]{oplus1}{y}
  \nclist{ncline}{dot2,h2,ds2,us2,g2,oplus2}
  \ncline[nodesepB=.35]{dot1}{dots1}
  \ncangle[nodesepA=.35,angleA=-90,angleB=180]{dots1}{hM}
  \nclist{ncline}{hM,dsM,usM,gM}
  \ncangle[nodesepB=.35,angleB=-90]{gM}{dots6}
  \ncline[nodesepA=.35]{dots6}{oplus2}
  \ncline{oplus2}{oplus1}
\end{pspicture}
```

References

[1] Hendri Adriaens. xkeyval package. CTAN:/macros/latex/contrib/xkeyval, 2004.

[2] Dominique Rodriguez and Herbert Voß. pstricks-add package. CTAN:/graphics/pstricks/contrib/pstricks-add, 2008.

[3] Timothy Van Zandt. *PSTricks - PSTricks macros for generic TEX*. http://www.tug.org/application/PSTricks, 1993.

[4] Timothy Van Zandt. pst-node package. CTAN:/graphics/pstricks/base/pst-node, 1999.

[5] Timothy Van Zandt. pst-plot package. CTAN:graphics/pstricks/base/pst-plot, 1999.