

SCILAB

An overview of Scilab's introduction, how to install, how to use and small demo program.

I. Introduction

SCILAB is a free scientific software for engineering and scientific applications. It bases on a numerical, programming and graphics environment; similar in operation but different syntax from MATLAB.

This is a self-contained package developed since 1990 by researchers from INRIA and ENPC (France); it is now maintained and developed by Scilab Consortium since its creation in May 2003.

Distributed freely and open source, Scilab is currently being used in educational and industrial environments around the world.

Scilab includes hundreds of mathematical functions with the possibility to add interactively programs from various languages (C, Fortran...). It has sophisticated data structures (including lists, polynomials, rational functions, linear systems...), an interpreter and a high level programming language.

Scilab can be run using a variety of OS like UNIX, Linux, Windows(9X/2000/XP), etc.

II. How to install

1. System Requirements

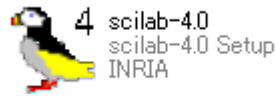
Scilab is distributed in source code format; binaries for Windows95/NT systems and several popular Unix/Linux-XWindow systems are also available: Dec Alpha (OSF V4), Dec Mips (ULTRIX 4.2), Sun Sparc stations (Sun OS), Sun Sparc stations (Sun Solaris), HP9000 (HP-UX V10), SGI Mips Irix, PC Linux. All of these binaries versions include tk/tcl interface.

The installation requirements are the following :

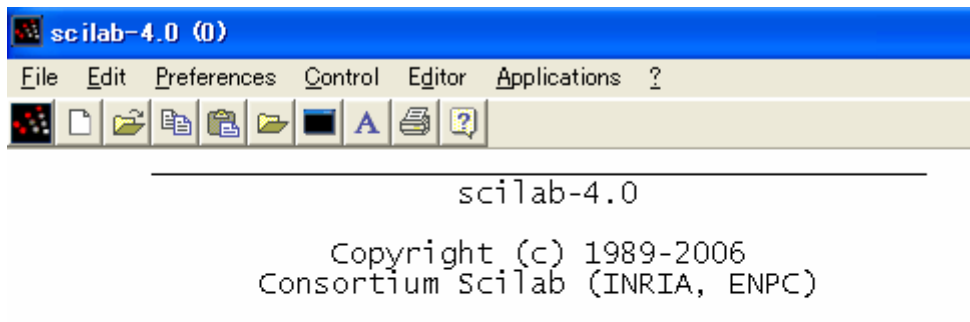
- for the source version: Scilab requires approximately 130Mb of disk storage to unpack and install (all sources included). You need X Window (X11R4, X11R5 or X11R6, C compiler and Fortran compiler (e.g. f2c or g77 or Visual C++ for Windows systems).
- for the binary version: the minimum for running Scilab (without sources) is about 40 Mb when decompressed. These versions are partially statically linked and in principle do not require a fortran compiler.

2. Install Scilab

The simplest way to install and use Scilab is download scilab-4.0.exe, run and follow its steps



After finish installing:



```
Startup execution:  
loading initial environment
```

```
-->_
```

II. How to use

- Editing a command line
 - enter a command line by typing after the prompt, or
 - using Scipad to edit
- The Scilab window has the following **Control** buttons.
 - Stop interrupts execution of Scilab and enters in pause mode
 - Resume continues execution after a pause entered as a command in a function or generated by the Stop button or Control C.
 - Abort aborts execution after one (or several) pause, and returns to top-level prompt
 - Restart clears all variables and executes startup files
 - Quit quits Scilab
 - Kill kills Scilab shell script
 - Demos for interactive run of some demos
 - File Operations facility for loading functions or data into Scilab, or executing script files.
 - Help : invokes on-line help with the tree of the man and the names of the corresponding items. It is possible to type directly help <item> in the Scilab window.

- Graphic Window : select active graphic window
- New buttons can be added by the addmenu command.

● Sample Session for Beginners

```
-->a=1;

-->A=2;

-->a+A
ans =

    3.

-->//Two commands on the same line

-->c=[1 2];b=1.5
b =

    1.5

-->//A command on several lines

-->u=1000000*(a*sin(A))^2+...
-->    2000000*a*b*sin(A)*cos(A)+...
-->    1000000*(b*cos(A))^2
u =

    81268.994

-->who
your variables are...

ans      b      a      bugmes  %helps  scicos_pal
MSDOS    home    PWD     TMPDIR  percentlib  soundlib
xdesslib utllib  tdcslib siglib  s2flib   roplib   optlib
metalib  elemLib commLib polylib  autolib  armalib  alglib
intlLib  mtlblib SCI     %F      %T      %z      %s
%nan     %inf    $      %t      %f      %eps    %io
%i       %e
using    4997 elements out of 1000000.
and      43 variables out of 1791
```

- ★Notes: Scilab is case-sensitive.
- // is not interpreted (it is a comment line).

● Matrix:

```

-->I=1:3
I =
    1.    2.    3.
-->W=rand(3,4)
W =
    0.6856896    0.8415518    0.8784126    0.5618661
    0.1531217    0.4062025    0.1138360    0.5896177
    0.6970851    0.4094825    0.1998338    0.6853980
-->W(1,I)
ans =
    0.6856896    0.8415518    0.8784126
-->W(:,I)
ans =
    0.6856896    0.8415518    0.8784126
    0.1531217    0.4062025    0.1138360
    0.6970851    0.4094825    0.1998338
-->W($,$-1)
ans =
    0.1998338
-->Q=[1 2 3;4,5,6]
Q =
    1.    2.    3.
    4.    5.    6.

```

★Notes: The \$ symbol stands for the last row or last column index of a matrix or vector.
The colon symbol stands for “all rows” or “all columns”.

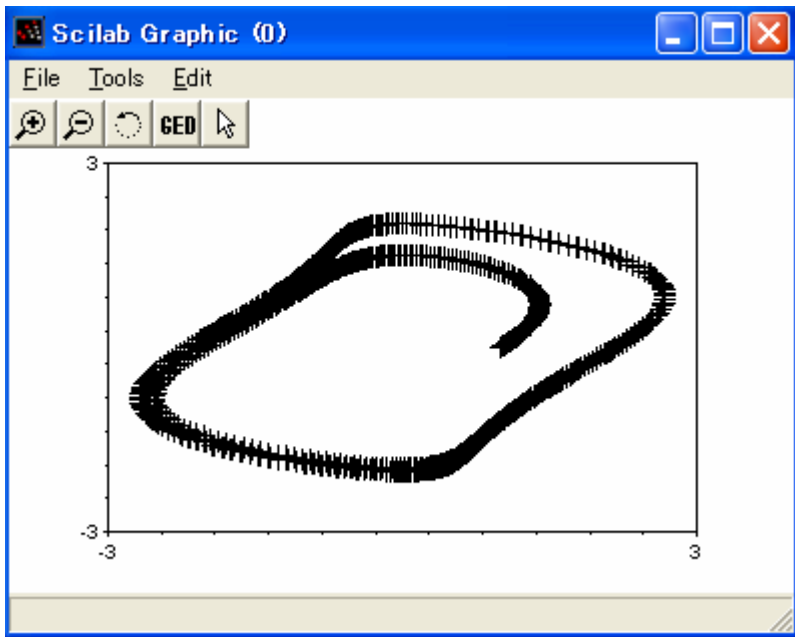
- **Function**

```

-->function ydot=f(t,y)
-->ydot=[a-y(2)*y(2) -1;1 0]*y
-->endfunction
-->a=1;y0=[1;0];t0=0;instants=0:0.02:20;
-->y=ode(y0,t0,instants,f);
-->plot2d(y(1,:),y(2,:),style=-1,rect=[-3,-3,3,3],max=[10,2,10,2])
-->xtitle('Van der Pol')

```

We have the result:



- **Boolean Matrices**

Boolean constants are %t and %f. They can be used in boolean matrices. The syntax is the same as for ordinary matrices i.e. they can be concatenated, transposed, etc...

Operations symbols used with boolean matrices or used to create boolean matrices are == and ~ =

```
-->%t
```

```
%t =
```

```
T
```

```
-->[1,2]==[1,3]
```

```
ans =
```

```
T F
```

```
-->[1,2]==1
```

```
ans =
```

```
T F
```

```
-->a=1:5; a(a>2)
```

```
ans =
```

```
3. 4. 5.
```

```
-->A=[%t,%f,%t,%f,%f,%f];
```

```
-->B=[%t,%f,%t,%f,%t,%t]
```

```
B =
```

```
T F T F T T
```

```
-->A|B
```

```
ans =
```

T F T F T T

-->A&B

ans =

T F T F F F

- Integer matrices:

There are 6 integer data types defined in Scilab,

- 32 bit signed integers (int32)
- 32 bit unsigned integers (uint32)
- 16 bit signed integers (int16)
- 16 bit unsigned integers (uint16)
- 8 bit signed integers (int8)
- 8 bit unsigned integers (uint8)

e.g.

```
-->x=[1.1 2.2 3.3 4.4 5.5]
```

```
x =
```

```
1.1 2.2 3.3 4.4 5.5
```

```
-->int8(x)
```

```
ans =
```

```
! 1 2 3 4 5 !
```

- Matrix Operations

SYMBOL	OPERATION
[]	matrix definition, concatenation
;	row separator
()	extraction $m = a(k)$
()	insertion: $a(k) = m$
'	transpose
+	addition
-	subtraction
*	multiplication
\	left division
/	right division
^	exponent
.*	elementwise multiplication
.\	elementwise left division
./	elementwise right division
.^	elementwise exponent
.*.	kronecker product
./.	kronecker right division
.\.	kronecker left division

- Exit from Scilab.

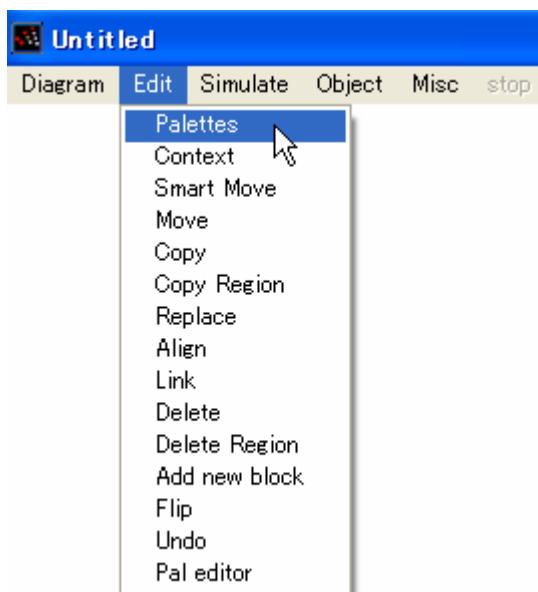
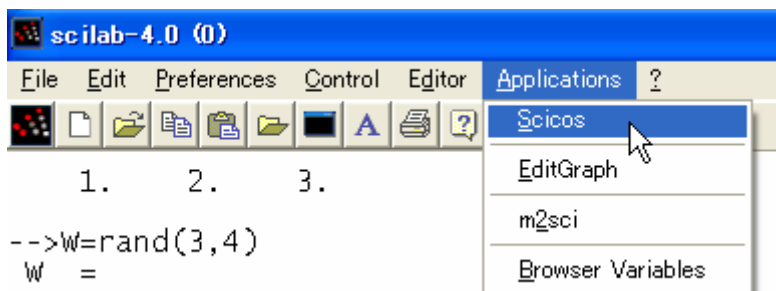
-->quit

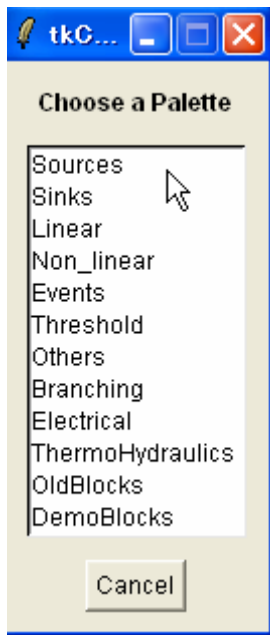
- **Programming:** You can open Scipad to write your program



- **Programming Tools:** Scilab supports a full list of programming tools
 - ✧ loops, (for, while)
 - ✧ conditionals, (==, <=, >=, <> or ~=)
 - ✧ case selection, and
 - ✧ creation new functions.
- **Special Function Commands:** Scilab has several special commands which are used almost exclusively in functions. These are:
 - ✧ argn: returns the number of input and output arguments for the function
 - ✧ error: used to suspend the operation of a function, to print an error message, and to return to the previous level of environment when an error is detected.

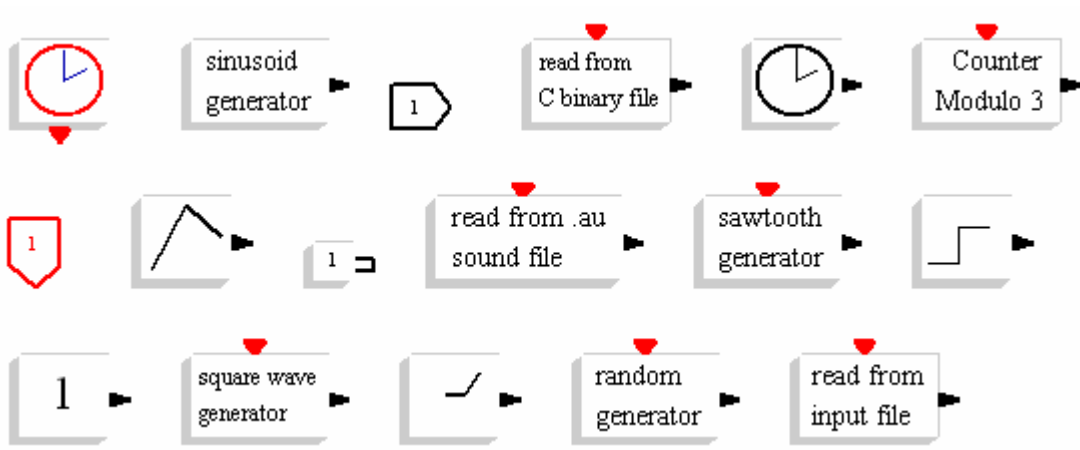
- ✧ **warning,**
 - ✧ **pause:** temporarily suspends the operation of a function.
 - ✧ **break:** forces the end of a loop
 - ✧ **return or resume :** used to return to the calling environment and to pass local variables from the function environment to the calling environment.
- **SCICOS:** Scicos is a [Scilab](#) toolbox included in the Scilab Package. Written in Fortran, C and Scilab language, it comes with complete source code. Scicos provides many functionalities available in *Simulink* and *SystemBuild*.



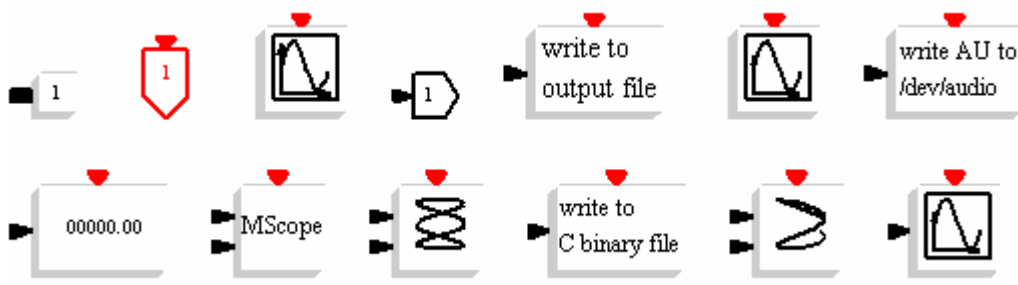


Palettes:

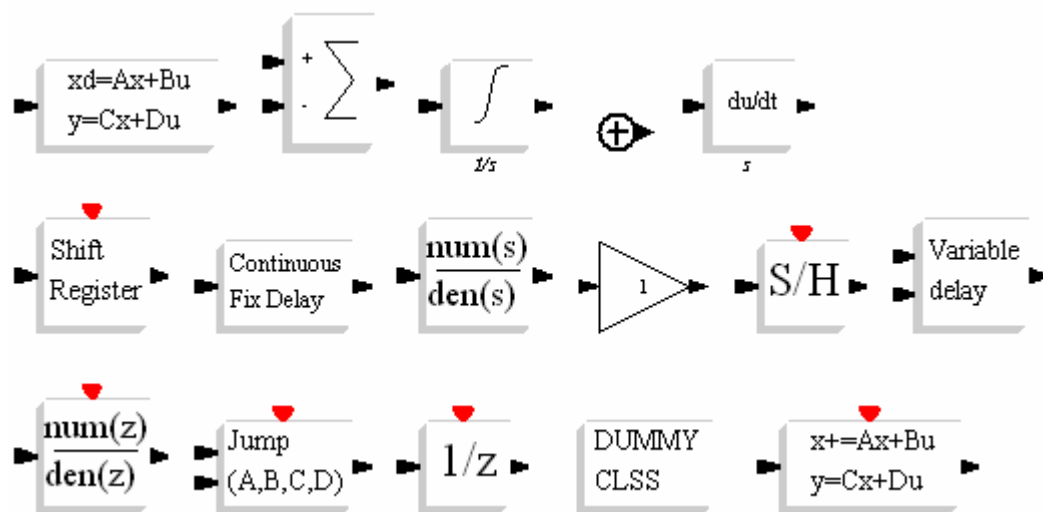
Sources



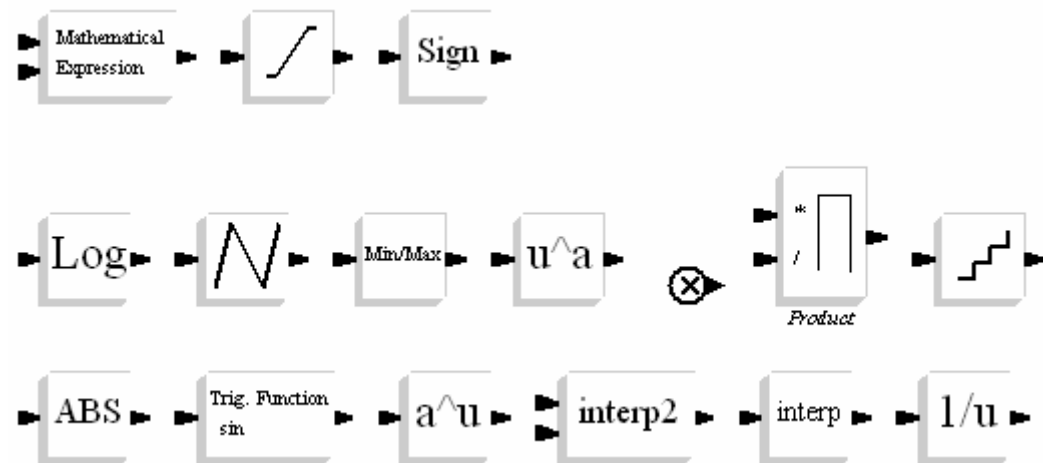
Sinks



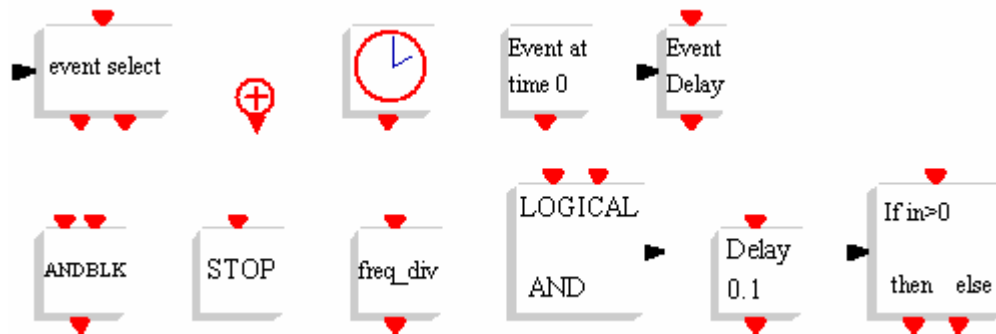
Linear



Non-linear



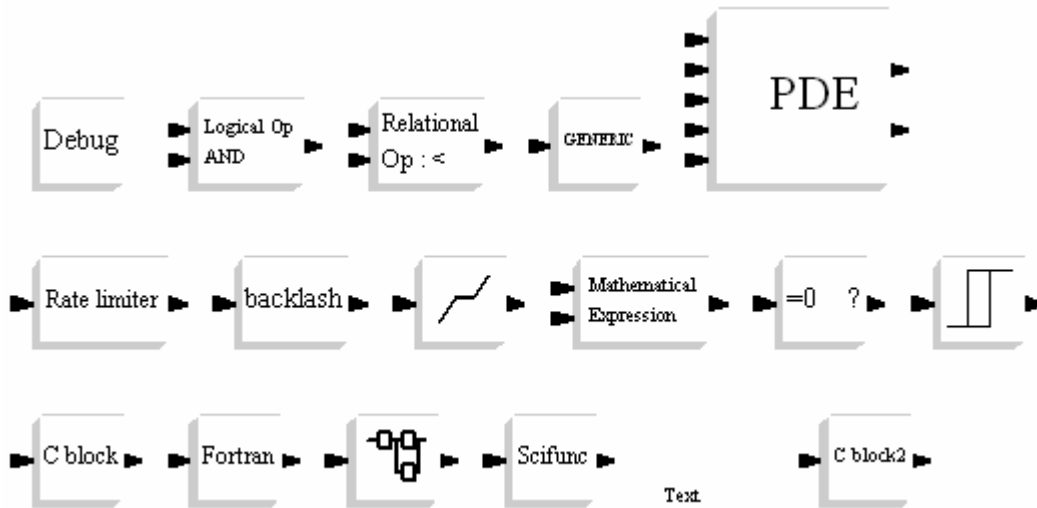
Events



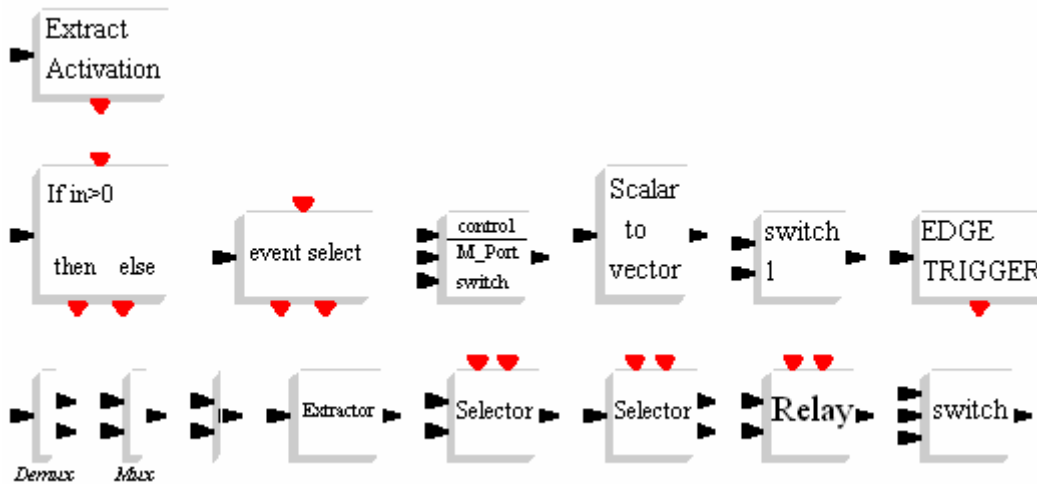
Threshold



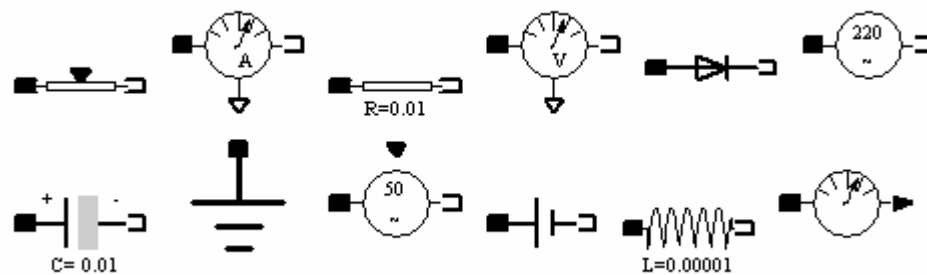
Others



Branching



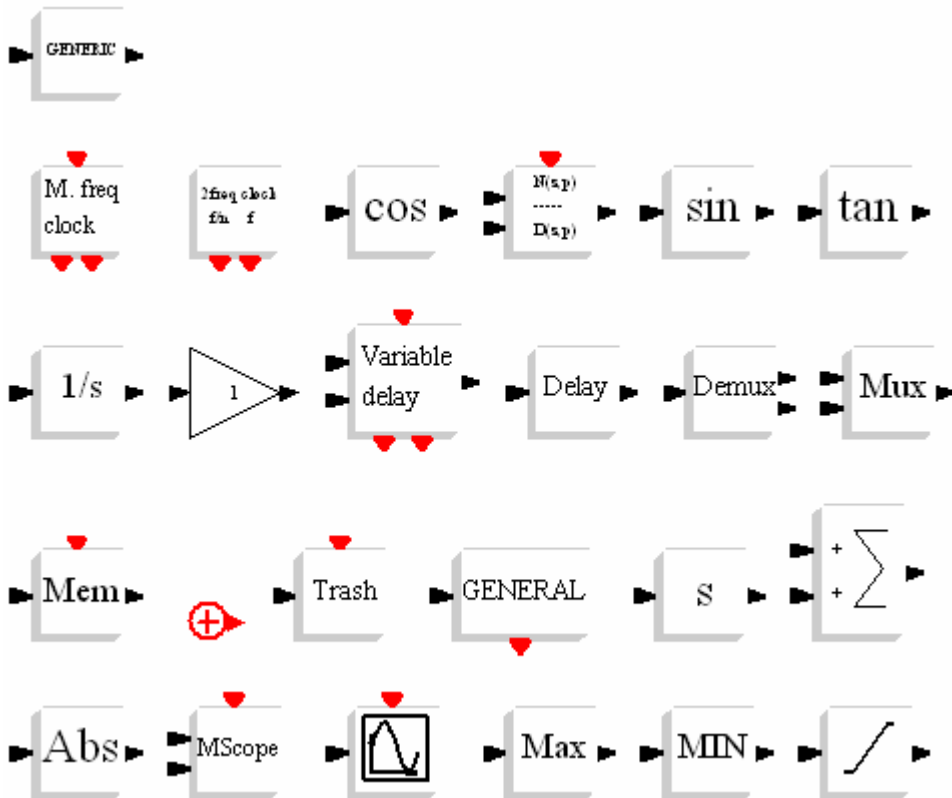
Electrical



ThermoHydraulics



Old Blocks

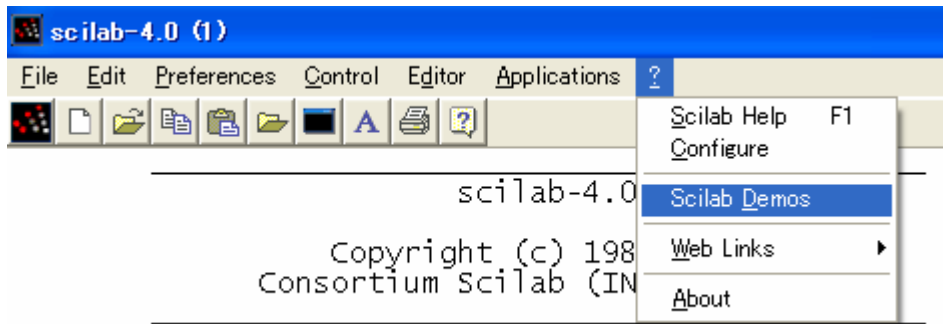


DemoBlocks



Notes

- Scilab doesn't have as many blocks as Matlab.
- Scilab doesn't have Visualize GUI as Matlab.
- Scilab has many demos for us to self-study:



Although both types of file (.sce files and .sci files) contain code, they are different. .sce files are executed while .sci files are loaded to be used when we need them.

III. Useful Links

<http://www.scilab.org/>

<http://scilabsoft.inria.fr/>

<http://www.scicos.org/>

http://www.infoclearinghouse.com/Merchant2/merchant.mv?Screen=CTGY&Category_Code=S