COMPARATIVE STUDY OF MATLAB AND ITS OPEN SOURCE ALTERNATIVE SCILAB

OSSRC TECHNICAL REPORT OSS 0601

Srikanth S V
National Institute of Technology,
Calicut
email : srikanthssv@gmail.com
( on Summer Fellowship to OSSRC, Mumbai )
Contents

Acknowledgment 4

1 Background of this Study 6

2 Scilab and Matlab – Some Must know facts 7

3 Loading Time 9

4 GUI
   4.1 The Matlab GUI 10
   4.2 The Scilab GUI 11

5 Help
   5.1 The Matlab Help 12
   5.2 The Scilab Help 13

6 Some Common functions in Matlab and Scilab 14

7 Going Engineering Specific 16
   7.1 Control System Implementation 17
   7.2 Control System Implementation using Matlab 18
   7.3 Control System Matlab Output 19
   7.4 Control System Matlab Plots 21
   7.5 Control System Implementation using Scilab 24
   7.6 Control System on Scilab – Command Window 24
   7.7 Control System on Scilab – Rtool 26
   7.8 Control System Scilab plots 28
   7.9 Control System Matlab vs Scilab 30

8 Image Processing Implementation 31
   8.1 Image Processing using Matlab 32
   8.2 Image Processing Matlab Output 33
   8.3 Image Processing using Scilab 34
   8.4 Image Processing Scilab Output 35
   8.5 Image Processing Matlab vs Scilab 36

9 Summary 37
   9.1 Concluding Remarks and Directions for Improvement 39
      Appendix A – User defined Scilab functions 41
      Appendix B – Scilab must have tools 42
Acknowledgment

My sincere thanks to Mr. Hari R and Dr. Sasikumar M for having given me this opportunity to work on such a lucrative project. I would also like to thank the OSSRC team for having provided a great work environment.

This project would not have been complete without the guidance and timely inputs from Dr. Abraham T. Mathew (Ass. Professor, NIT Calicut), Mr. Cyril (OSSRC Team), Mr. Deepak Malani (for his help on Image processing), Mr. Thomas Jos (my dear friend, for his first hand comments and suggestions on the report).

Last but not the least, I thank the entire work force at C-DAC (Kharghar) for the part they played in making the period of work a joyful experience.
1. BACKGROUND OF THIS STUDY

The term “MATLAB” is familiar to every Engineering graduate. MATLAB is a scientific computational package that has been widely in use ever since its inception in the early nineties. In the beginning it was limited to the research arena but later it gained a prominent place in the Engineering course syllabus, especially the Electrical and Electronics branches.

The one great problem with MATLAB is that it is a proprietary software and thus involves a huge licensing fee which may even run down to thousands of dollars. Another problem with the package was that its source code was not public.

With the advent of Open source Softwares and increased usage of the same towards the late eighties, the need to come up with an Open source alternative to MATLAB rose. Thus the SCILAB project begun. The first version of SCILAB was launched in 1994 and since then it has been constantly updated and is available for download via the Internet.

Sadly the SCILAB package has not received the share of success it ought to have got. Though available for free, it has not been able to build a great user community. In this project an attempt has been made to compare the SCILAB software with the MATLAB package and identify the strengths and weaknesses of the SCILAB package. The project also suggests some probable solutions to make the package better.
2. SCILAB and MATLAB – SOME MUST KNOW FACTS

2.1 SCILAB HISTORY

It's an open source software package developed at INRIA (France), for system control and signal processing applications. It also features a wide variety of tools for various Engineering and Mathematical applications. It was introduced as an Open source alternative to MATLAB. It is also a vector based program. It has constantly undergone vital changes ever since its inception in 1994.

2.2 HOW TO OBTAIN?

It can be freely downloaded at the home page (http://www.scilab.org/). The site offers the latest version for all the prominent Operating systems. The Windows version is a 13.6 MB zip file (Scilab 4.0). The user is free to modify the source which is also readily available at the same link. The package also includes a simulator called SCICOS which is an open source alternative to SIMULINK. The minimal package does not include subject specific toolbox’s and are to be downloaded separately.

2.3 INSTALLATION & REQUIREMENTS

The Scilab installation file has to be unzipped into a folder from which it can be easily installed into the desired location by double clicking the installation file.

In case of Linux, the setup file in binary version is a 15.6 MB .tar file which has to be extracted into a directory. Open the Linux terminal and then move to the directory into which the SCILAB .tar file has been extracted. Typing the “make” command from the SCILAB directory produces an executable file, which can be invoked by typing “scilab”. The standard package occupies only 130 MB of disk space upon installation.
2.4 MATLAB HISTORY

MATLAB is a high-level technical computing language and an interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas. Mathworks Inc. owns the proprietary software.

2.5 HOW TO OBTAIN?

Matlab – 7.2 is the latest version commercially available and can be purchased online for $99. This is exclusive of even a single toolbox and the total product along with all the toolboxes and SIMULINK can cost up to $2,812 for an individual edition. The purchase can also be made via their Sales division and more details on this is available at http://www.mathworks.com/.

2.6 INSTALLATION & REQUIREMENTS

The latest version (MATLAB 7.2) would require a disk space of 460 MB for MATLAB alone. The complete package could demand up to 2.5GB of disk space. The latest version would require a minimum of 512MB RAM. The installation process is quite simple and the Installer guide is excellent. The software is normally available in 2 CD’s. The version used throughout this report is MATLAB 7.0. The version does run on a system with 128 MB RAM but is really sluggish.

note

All the below mentioned values were obtained when both the packages were run on a system with 256 MB and running Windows XP Operating System.
3. LOADING TIME

The SCILAB software takes hardly five seconds to load, while the MATLAB package took around 15 seconds.

The MATLAB help is highly descriptive but takes around 10 seconds to load. On the contrary, the SCILAB help loads almost instantaneously.

The same is the case with the MATLAB editor, which consumes about 5 seconds to load while its open source competitor loads immediately.

The time taken in MATLAB to load a figure while using functions like plot(), is far more than the time taken by SCILAB to do the same.

Thus in terms of LOADING TIME it is evident that the Open source software SCILAB has an edge over MATLAB.

One other great feature of SCILAB is that it supports more than one instance without much compromise on performance. This means that it is possible for us to run two or more instances of Scilab simultaneously. Though this feature is also supported by MATLAB, running more than one instance loads the computer and the performance deteriorates heavily.
4. GUI

4.1 THE MATLAB GUI

The MATLAB desktop is excellent and comprises the Command history window, Workspace Window, Command Window and the Current Directory. The Menu is also highly featured and includes almost all the required tools. All the windows can be moved on to any area on the screen using the mouse. A screenshot of the MATLAB desktop is shown below.

![The MATLAB desktop](image)

**Figure 1.** The MATLAB desktop

All the windows such as Workspace, Command Window etc can be docked anywhere on the screen using the button shown below.
4.2 THE SCILAB GUI

SCILAB, however does not present the user a highly interactive environment to work with. When loaded in LINUX the GUI features are even less. The desktop includes only the command window and there is no Command history Window available. The featured Menu also looks very dull and many essentials tools are absent. A screenshot of the SCILAB desktop is shown below

```
-->s = poly(0, "s");
-->L = syslin('c', 3e4 *(0.05*s + 1)^2 / ((s+1)^3 *(0.01*s + 1)));
-->bode(L);
-->...
```

Figure 2. The SCILAB desktop

The desktop does not have a Command history window though the previously typed in commands can be accessed using the up/down arrow buttons. The Workspace Window is also absent so is the Current directory Window.

Thus from the GUI point of view MATLAB is definitely better than SCILAB.
5. HELP

5.1 THE MATLAB HELP

The Help available on MATLAB is detailed and includes alphabetical listing of all functions available. It also enlists the various toolboxes featured and on clicking them the user is guided to the required function. The demo’s provided are also nice and gives the user a glimpse into the world of Matrix programming. The help also supports a detailed search for any function, data type etc.

![MATLAB Help](image)

*Figure 3. The MATLAB help*

The MATLAB help also includes features such as categorical function list which are very much helpful for beginners to learn MATLAB. The Help layout is highly user friendly. The Search provided is excellent. The only problem with the MATLAB Help is that it takes 5-10 seconds to load and the search is also slow.
5.2 THE SCILAB HELP

The SCILAB Help is very poor. The examples provided are not satisfactory. It does not include categorical listing of functions in a toolbox and the documentation provided for the functions is also poor. The help documents are not available in portable document format (.pdf). It does not feature a "Getting Started “ link for beginners.

If a new toolbox is installed into SCILAB, its help is not available and does not form a part of the SCILAB help unless the toolbox is loaded on every start-up. This is very annoying because each time the user carries his program from one computer to another he has to carry the accompanying toolbox.

Concluding, MATLAB Help is much better than SCILAB help.
### 6. SOME COMMON FUNCTIONS IN MATLAB AND SCILAB

<table>
<thead>
<tr>
<th>SlNo</th>
<th>Function ...Scilab</th>
<th>What it does</th>
<th>Function....Matlab</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>abs</td>
<td>Absolute value and complex magnitude</td>
<td>abs</td>
</tr>
<tr>
<td></td>
<td>abs(1+%i)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>acosh</td>
<td>Inverse hyperbolic cosine similar : asinh, atanh etc</td>
<td>acosh</td>
</tr>
<tr>
<td></td>
<td>acosh(x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>acos</td>
<td>Inverse cosine (answer in radians) similar : asin, atan etc</td>
<td>acos</td>
</tr>
<tr>
<td></td>
<td>acos(x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>angle</td>
<td>Phase angle of a complex number</td>
<td>angle</td>
</tr>
<tr>
<td></td>
<td>atan(imag(x), real(x))</td>
<td></td>
<td>angle(A)</td>
</tr>
</tbody>
</table>

Comment: the “angle” command is not available on SCILAB and the operation is to be performed using the statement shown.

| 5    | cd                  | Change/get working directory | cd                  |
|      | cd(“..”)            |                            | cd..               |
| 6    | clc                 | Clear Command Window        | clc                |

Comment: the option clc([number of lines]) deletes the mentioned no. of lines above the cursor.

<p>| 7    | clear               | Remove items from workspace, freeing up system memory | clear              |
| 8    | clf                 | Clear figure window          | clf                |
| 9    | conj                | Complex conjugate            | conj               |
|      | conj(a)             |                              | conj(a)            |
| 10   | cos                 | Cosine of angle expressed in radians similar sin tan etc | cos                |
|      | cos(A)              |                              | cos(w)             |
| 11   | date                | Current date string          | date               |
| 12   | det                 | Determinant of a square matrix | det                |
|      | det(a)              |                              | det(a)             |
| 13   | dir                 | Directory listing            | dir                |
| 14   | exit/quit           | Ends current session         | exit/quit          |
| 15   | help/F1             | Display help                  | help/F1            |
| 16   | mtlb_hold           | Keep drawing on the same graph | hold               |
| 17   | imag                | Imaginary part of a complex number | imag              |
|      | imag(a)             |                              | imag(a)            |
| 18   | inv                 | Inverse of a square matrix   | inv                |
|      | inv(a)              |                              | inv(a)             |</p>
<table>
<thead>
<tr>
<th>SNo</th>
<th>Function ...Scilab</th>
<th>What it does</th>
<th>Function....Matlab</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>log10</td>
<td>Logarithm to the base 10</td>
<td>log10</td>
</tr>
<tr>
<td></td>
<td>log10(a)</td>
<td></td>
<td>log10(a)</td>
</tr>
<tr>
<td>20</td>
<td>log</td>
<td>Logarithm to the natural base e</td>
<td>log</td>
</tr>
<tr>
<td></td>
<td>log(a)</td>
<td></td>
<td>log(a)</td>
</tr>
<tr>
<td>21</td>
<td>real</td>
<td>Real part of a complex number</td>
<td>real</td>
</tr>
<tr>
<td></td>
<td>real(a)</td>
<td></td>
<td>real(a)</td>
</tr>
<tr>
<td>22</td>
<td>round</td>
<td>Round to the nearest integer place</td>
<td>round</td>
</tr>
<tr>
<td></td>
<td>round(7.66)</td>
<td></td>
<td>round(8.33)</td>
</tr>
<tr>
<td>23</td>
<td>sqrt</td>
<td>Square root of a number</td>
<td>sqrt</td>
</tr>
<tr>
<td></td>
<td>sqrt(a)</td>
<td></td>
<td>sqrt(a)</td>
</tr>
</tbody>
</table>

**note**

1. Some other MATLAB functions such as “hold” (to keep drawing on the same plot) can be used in SCILAB using the “mtlb” prefix.
   Eg: mtlb_hold // does the same function of “hold” in MATLAB

2. The use of “%i” in SCILAB for a complex number is cumbersome. This is because SCILAB allows the character “i” to be used as a variable name while MATLAB does not allow the same.
MATLAB is heavily used by students pursuing their Engineering degree in Electrical/Electronics streams. Thus, the toolboxes most widely used by students include the Control Systems toolbox and the Image processing toolbox. We will discuss the two in this section.

The default MATLAB package is exclusive of any toolbox and thus they have to be purchased separately. Some basic control system functions are a part of the MATLAB standard package but that is insufficient for the analysis of Control systems. The Control System toolbox and the Image processing toolbox cost $59 each.

The default Scilab package includes neither the Image processing toolbox nor the Control Systems toolbox. Though the Control System toolbox does not exist separately almost all the required functions are either available in the general package or can be easily incorporated. The SCILAB home page also provides the user with a handy graphical tool to help create simple control systems and find its parameters. The tool called “RLtool” can downloaded at this link http://www.ee.iitb.ac.in/uma/~ishan/scilab/rltool.htm.

In case of Image processing we use the ready made functions available in the MATLAB Image processing toolbox which contains all the required functions.

The standard SCILAB package does not have an image processing toolbox but the SIP( Signal and Image processing ) toolbox can be freely downloaded from this link http://siptoolbox.sourceforge.net/. The package uses the .dll (dynamic link library) files of the ImageMagick software (an open source image processing tool) and thus requires the package to be installed on the system. The binary file for a Windows/Linux installation is available at its link http://www.imagemagick.org/script/index.php.
7.1 CONTROL SYSTEMS IMPLEMENTATION

In the following section, a general control system is built on MATLAB and the system parameters such as Bode plot, Nyquist plot etc are evaluated. The same system is then created using SCILAB and the functions used are compared. In case of SCILAB we shall employ both the standard commands as well as the graphical tool “RLtool” to create and analyze the control system.

7.2 CONTROL SYSTEMS using MATLAB

The code given below was saved in a .m (the matlab executable file format) file and was run using MATLAB.

```matlab
% this code creates a simple control system and studies its properties
% written by srikanth s. v
1   num=[2 0];  % the numerator of the transfer function in decreasing power of 's' - 2s
2   den=[1 2 4 ];  % similar to the above line but for the denominator – s^2+2s+4
3   sys=tf(num,den)  % the 'tf' function defines the variable sys to be the transfer function defined
% by the numerator and denominator and displays sys
4   A=[-2 -4;1 0];  % entering the same system using its state space representation
5   B=[1;0];  % notice the “;” operator is given at the end of a statement to prevent
6   C=[2,0];  % MATLAB from echoing the answer soon after the execution of the line
7   D=[0];
8   sys1=ss(A,B,C,D) % defining sys1 to be the system with A B C D as its state space
% representation and displays it
9   sys2=tf(sys1);  % defines sys2 in the transfer function format created from the system sys1
10  sys2  % displays the system sys2 in transfer function format
11  [A1,B1,C1,D1] = tf2ss(num,den)  % finds the state space equivalents of system created from num & den
12  [num,den]=ss2tf(A1,B1,C1, D1,1)
% this function ss2tf converts a system from state space representation
% into transfer function format
13  bode(sys)  % plots the bode plot of the linear system sys
14  margin(sys)  % plots the phase and gain margins on the bode plot previously plotted
15  pzmap(sys)  % plots the pole zero plot of the system sys
16  rlocus(sys)  % plots the root locus plot of the system sys
17  nyquist(sys)  % plots the nyquist plot of the system sys
18  impulse(sys)  % plots the pole impulse response of the system sys
19  step(sys)  % plots the step response of the system sys
```
note

1. The above statements can be written in any text editor and saved in a file with a .m extension.

2. It must also be noted that MATLAB is an interpreter and checks for errors in every line and only then executes the next line.

3. The MATLAB interpreter ignores any statement given after the % percentile special character.

4. The MATLAB figure window also provides an option to save the figure in many popular formats such as .bmp, .jpeg etc.

5. The plots open up in a separate window and there are zoom in and zoom out options on the window (a screenshot of the figure window is given below).

Figure 5. The MATLAB figure window
The output of the above code is shown in the box below. The MATLAB command window fonts can be changed in the File->Preferences->Fonts menu.

Transfer function:
\[
\frac{2s}{s^2 + 2s + 4}
\]
a =
\[
\begin{bmatrix}
  x1 & x2 \\
  x1 & 0 & 1 \\
  x2 & -4 & -2
\end{bmatrix}
\]
b =
\[
\begin{bmatrix}
  u1 \\
  x1 & 0 \\
  x2 & 2
\end{bmatrix}
\]
c =
\[
\begin{bmatrix}
  x1 & x2 \\
  y1 & 1 & 0
\end{bmatrix}
\]
d =
\[
\begin{bmatrix}
  u1 \\
  y1 & 0
\end{bmatrix}
\]
Continuous-time model.
Transfer function:
\[
\frac{2}{s^2 + 2s + 4}
\]
A1 =
\[
\begin{bmatrix}
  -2 & -4 \\
  1 & 0
\end{bmatrix}
\]
B1 =
\[
\begin{bmatrix}
  1 \\
  0
\end{bmatrix}
\]
C1 =
\[
\begin{bmatrix}
  2 & 0
\end{bmatrix}
\]
note

1. The plots have to be plotted one by one by making the other plot statements inactive by using the "%" operator.

2. If the "%" operator is not used MATLAB shows only the last plot.
7.4 CONTROL SYSTEM - MATLAB PLOTS

Figure 6. The Bode plot also indicating the phase and gain margin on MATLAB

Figure 7. The pole zero plot indicating the position of the poles and zeros on MATLAB
Figure 8. The MATLAB root locus plot of the system

Figure 9. The MATLAB nyquist plot of the system
**Figure 10.** The Step response of the system on MATLAB

**Figure 11.** The Impulse response of the system on MATLAB
7.5 CONTROL SYSTEMS using SCILAB

The standard SCILAB package comes with many of the essential functions used for the study of control systems though it does not have a separate control system toolbox. It however does not contain functions such as “tf” and “impulse” which are commonly used in this field of study. Also some functions have a slightly different syntax in comparison to MATLAB. The SCILAB homepage also provides a graphical tool “RLtool” to aid Control System studies. We will also employ the tool to make the plots for the same.

The “tf” (tf- transfer function) function which is not available in the standard SCILAB package can be easily incorporated by copying the function code given in Appendix A into any text editor and saving the file with a .sci extension.

In this section, we shall employ both the graphical tool “RLtool” as well as the standard command window in creating and studying a simple control system.

7.6 CONTROL SYSTEM ON SCILAB - COMMAND WINDOW IMPLEMENTATION

Some of the control system plots can be drawn in the SCILAB Command Window itself using the statements shown below.

```scilab
// this code will create a simple control system . written by srikanth
// copy this into any text editor and save the file with a .sci extension and open the file using SCILAB
1  num=[2 0];
2  den=[1 2 4];
3  sys=tf(num,den); // the tf function is not a part of the standard package and can be easily implemented
   // by saving the function given in Appendix A in a file with .sci extension and then
   // executing it from Scilab ( File -> Execute )
4  bode(sys)  // the bode plot
5  clf // clear the figure window
6  nyquist(sys) // the nyquist plot
7  clf

// the output of the code given above can be obtained using RLtool and are given at the end of the next section
```
note

1. The SCILAB plots are shown below. The Graphic editor tool (GED) available along with the zoom in/out button provides the user with various options to change the appearance of the plot. Using this tool it is possible to change properties of the plot such as background color, the axes titles etc.

2. The SCILAB homepage also features another tool to help the user plots look the same as in MATLAB. This tool called the “plot lib” can be downloaded at this link http://www.dma.utc.fr/~mottelet/myplot.html

Figure 12. The SCILAB graph window
7. 7 CONTROL SYSTEM USING SCILAB - RLtool IMPLEMENTATION

RLtool is a highly interactive graphic tool which can be added on to SCILAB and can be used in Control System Design and Analysis. The RLtool .zip can be downloaded at this link http://www.ee.iitb.ac.in/uma/~ishan/scilab/rltool.htm

1. The .zip file has to be extracted into a folder and the builder.sci and loader.sci executed from the SCILAB window. type rlt() in the command window to run RLtool. The start window of RLtool is shown below

![Figure 13. The SCILAB RLtool start window](image)

2. Choose New and enter the system transfer function numerator and denominator at the spaces provided as shown and press Finish

![Figure 14. The SCILAB RLtool system transfer function window](image)
3. The default setting of the RLtool is to set the feedback path (sensor) gain to one which will change the system plots from the desired values. By default the control system formed is a closed loop system (left figure), We need to change the loop configuration to forward path (right figure) alone from the last drop down “Settings” menu. The tool also plots the bode plot by default as soon as we press the finish button.

![Figure 15. The SCILAB RLtool default (left) and modified (right) window respectively.](image)

4. The Gain of the system has to be set to one at the “Type Gain here” prompt.

5. To assure that the plant has the required transfer functions, click on the plant block and check the parameters.

6. Now to plot the Bode plot select Response-> Bode plot. Similarly the other plots can be plotted and are given below.
7.8 CONTROL SYSTEM - SCILAB PLOTS

Figure 16. The Bode plot on SCILAB

Figure 17. The Nyquist plot on SCILAB
Figure 18. The Impulse response on SCILAB

Figure 19. The Step response on SCILAB
7. 9 CONTROL SYSTEMS MATLAB vs SCILAB

I. The other MATLAB commands such as ss2tf, tf2ss etc are also available on SCILAB. They however operate on closed loop systems and do not produce the same results as on MATLAB.

II. The default SCILAB package does not contain various commonly used MATLAB functions such as \texttt{tf} (to create a linear system in transfer function format)

The \texttt{tf} function which is not available in the standard SCILAB package can be easily incorporated by copying the function code given in Appendix A into any text editor and saving the file with a \texttt{.sci} extension

\texttt{pzmap} (to plot the pole zero plot)

\texttt{impulse} (to plot the impulse response of the system)

\texttt{step} (to plot the step response of the system)

\texttt{rlocus} (to plot the root locus of the open loop system)

III. Some of the SCILAB plots are not auto-scaled. The user has to manually scale the plot using the zoom in/out buttons

Thus the SCILAB package integrated with the RLtool package does provide the user the essential tools to aid the study of Control Systems. It would be better if the standard package also included the functions mentioned above (\texttt{tf,impulse,step,rlocus}). But SCILAB on the other hand provides the user with quick plots when MATLAB takes several seconds to plot each figure.
The Image processing toolbox is primarily used by Engineering students and professionals. This section deals with comparing the features of both MATLAB and SCILAB in performing certain commonly used operations such as Averaging and Slicing of an image. These can be easily done using the MATLAB Image processing Toolbox.

The SIP (Signal and Image processing) toolbox which is required for the implementation of Image processing functions on SCILAB can be freely downloaded from this link http://siptoolbox.sourceforge.net/.

The package uses the .dll (dynamic link library) files of the ImageMagick software (an open source image processing tool) and thus requires the package to be installed on the system. The binary file for a Windows/Linux installation is available at its link http://www.imagemagick.org/script/index.php.
8. 1 IMAGE PROCESSING using MATLAB

The MATLAB implementation of the following operations is compiled in two .m files which are given below. Copy this code into a text editor and save the file with a .m extension

Average.m (produces the average of an image)

% this code gives the averaged image as the output
% written by srikanth
a = imread('geo.jpg'); % the file must be in the current directory
a = im2double(a(:,:,1)); % converts the binary image BW to a double-precision intensity image
for i = 2:119
    for j = 2:159
        c(i,j) = a(i,j) + a(i-1,j) + a(i+1,j) + a(i,j+1) + a(i-1,j+1) + a(i+1,j+1) + a(i,j-1) + a(i-1,j-1) + a(i+1,j-1);
        c(i,j) = c(i,j)/9;
    end
end
imshow(a); % display the original image
figure, imshow(c); % display the averaged image

Bitplane Slicing (bitsli.m)

% this code gives the darkened image produced by bit slicing
% written by srikanth
a = imread('geo.jpg');
a = a(:,:,1);
b = zeros(120,160,8);
for k = 1:8
    for i = 1:120
        for j = 1:160
            b(i,j,k) = bitget(a(i,j),k); % get the pixel color and change accordingly
            if b(i,j,k) == 1;
                b(i,j,k) = 255;
            end
        end
    end
end
figure, imshow(b(:,:,k));
8.2 IMAGE PROCESSING - MATLAB OUTPUT

Figure 20. The original picture file - MATLAB

Figure 21. The averaged picture - MATLAB

Figure 22. The sliced picture - MATLAB
The SIP toolbox once installed on the system can be activated by choosing SIP toolbox on the toolbox menu. The SIP toolbox gets loaded and also enlists the functions on the command window. Two basic operations involved in Image processing studies are the Image averaging and Image slicing. These operations can easily performed on SCILAB as shown below. Copy the code given below into a text editor and save with a .sci extension.

### Averaging

// The same operation can be accomplished using the imaver function given in the appendix A
// written by srikanth

```scilab
a=imread('d.jpg');
si=size(a);
for i=2:(si(1)-1)
    for j=2:(si(2)-1)
        c(i,j)=a(i,j)+a(i-1,j)+a(i+1,j)+a(i,j+1)+a(i-1,j+1)+a(i+1,j+1)+a(i,j-1)+a(i-1,j-1)+a(i+1,j-1);
        c(i,j)=c(i,j)/9;
    end
end
imshow(c);
```

### Slicing

// this operation of bit slicing can be performed on SCILAB using the im2bw function where the threshold value can be used to control the degree of darkness

```scilab
im=gray_imread('d.jpg');
bwim=im2bw(im,0.4);
imshow(bwim);
```
1. The launch of the SIP toolbox caused the SCILAB program to crash after a few functions were executed

2. It was also observed that the Execution time for Image processing functions was almost the same as that taken by MATLAB.
8. 5 IMAGE PROCESSING – MATLAB vs SCILAB

A  Most of the MATLAB Image processing toolbox commands can be used as such in SCILAB with the SIP (Signal and Image Processing) Toolbox.

B  The SCILAB program on Windows XP crashed after 4-5 commands were used when with the SIP toolbox loaded.

C  The operations such as Image Averaging and Image Slicing took almost the same time on both MATLAB and SCILAB.

D  Functions such as IMAVER can be easily incorporated into the SIP toolbox as given in Appendix A.

Concluding, the Image Processing Operations can be performed with equal ease both on MATLAB and SCILAB.
The table grades both the software packages with respect to the criterion on a 10 point scale (0 – 10)

<table>
<thead>
<tr>
<th>Criterion</th>
<th>MATLAB</th>
<th>SCILAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>System requirements</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>requirements are very high both on the RAM and Harddisk for optimum performance</td>
<td>functions well on a system with minimal requirements and has minimal dependencies</td>
</tr>
<tr>
<td>Cost</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>the standard package costs 99$ while to buy any other specific toolbox may cost 59$ to 69$</td>
<td>Its free and can be downloaded at <a href="http://www.scilab.org">www.scilab.org</a></td>
</tr>
<tr>
<td>GUI</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>the GUI is extensive and contains all essential features</td>
<td>the GUI is poor and requires many more features</td>
</tr>
<tr>
<td>Help</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>the help is highly resourceful and contains categorical listing of functions in all the toolbox's tutorials, demos, extensive search etc. The online help is also extensive and excellent tutorials are available</td>
<td>the help contains neither a categorical listing of functions nor an extensive search. It does not feature tutorials or a Getting Started tool. The online help available is also limited and not many tutorials are available.</td>
</tr>
<tr>
<td>Loading Time</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>It is very sluggish during the launch and is slow in computations and plotting graphs</td>
<td>It loads very fast and the computations are also quick. Even multiple instances run quite well. Image processing functions are slower</td>
</tr>
<tr>
<td>Control System Implementation#</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>the study of control systems is very easy and highly accurate. The number of functions available in the control system toolbox is very high</td>
<td>the package along with RLtool does aid in the study of Control system but the available functions are insufficient to complete the study</td>
</tr>
<tr>
<td><strong>Criterion</strong></td>
<td><strong>MATLAB</strong></td>
<td><strong>SCILAB</strong></td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>Image Processing Implementation#</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>The implementation is quite simple but lacks some of the ready made functions</td>
<td>The implementation is again quite simple and ready made functions are available. The program crashes after 4-5 function implementations</td>
</tr>
<tr>
<td>User Community (Popularity)</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>It is the widely used simulation software in the World today and is a part of the syllabus in almost all Engineering colleges worldwide. Used by reputed organizations such as NASA.</td>
<td>It is limited to the developer community and open source users.</td>
</tr>
<tr>
<td>Toolbox's</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>It deals to a wide range of topics including topics such as Financial Computations, Biomedical Engineering etc</td>
<td>The spectrum of topics dealt with, even in the contributors page is not large.</td>
</tr>
<tr>
<td>GRAND TOTAL</td>
<td>60/90 = 66.66%</td>
<td>50/90 = 55.55%</td>
</tr>
</tbody>
</table>

**NOTE**

# - The toolbox's considered include the two most widely used toolbox's by Engineering students.

Thus the implementation of a specific toolbox may be efficient on MATLAB where as some other toolbox might be better on SCILAB.

Thus the final scores obtained may change according to the required field of application.
9.1 CONCLUDING REMARKS and DIRECTIONS FOR IMPROVEMENT

The SCILAB package does contain all the basic features needed for a scientific computational program. The Open Source venture does have the caliber to pose a serious threat to proprietary softwares like MATLAB but lacks certain key features.

If these features are incorporated into the SCILAB package it can claim to be “Second to none” in terms of content and accuracy. Some of them are given below.

1. INSTALLATION

The SCILAB homepage must include a list of available toolbox's on a separate page. The page must also rank the toolbox's in each category if more than one are present (the ranking may be according to the maximum downloads). The page must also prompt the user to enter his field of study and recommend the appropriate toolbox's to be downloaded.

2. GUI and VISUAL APPEAL

The SCILAB GUI requires a complete makeover. One of SCILAB's most disturbing features is its lack of visual appeal. The Desktop needs to incorporate many essential windows such as Workspace and Command History. The Current Directory Window on MATLAB is not quite essential.

3. DOCUMENTATION

The SCILAB Help needs to be supplemented with more examples and demo's. It also requires an extensive “Search”. Many more tutorials also have to be integrated into the Help feature. The section must also include “Getting Started” tutorials for beginners. It must also include Categorical listing of functions.
4. FUNCTIONALITY

The package does not include some of the basic functions very commonly used in Control system analysis. These functions include

a.) tf (transfer function)
   the code to implement the tf function on SCILAB is given in Appendix A and can be incorporated by copying the code into any text editor and saving the file with as tf.sci (the extension must be “.sci” for SCILAB to read the file)

b.) pzmap (function to plot the pole zero map of the system)

c.) impulse (function to plot the impulse response of the system)

d.) step (function to plot the step response of the system)

e.) rlocus (function to plot the root locus of the open loop system)

5. MINOR FEATURES

Some of the minor changes to be incorporated into the SCILAB package are

a.) The use of %i in case of complex numbers is very annoying and must be changed into “i” (i stands for square root of -1)

b.) The SCILAB figure window must allow users to save files in any format directly rather than exporting it to some other format and then saving it.

6. TOOLBOX's

The SCILAB package include many more subject specific toolbox's such as those on Fuzzy logic, Communications, Neural Networks, Filter design etc. The user contributed toolbox's must be categorically listed and they must also be ranked (if more than one is available per topic)
APPENDIX-A

THE TF FUNCTION FOR SCILAB
// the function takes in the numerator and denominator matrices in the decreasing order of s and gives
the linear system in the transfer function format as the output
// same as tf on MATLAB
// written by Srikanth S V , NITC, email : srikanthssv@gmail.com
function [sys]=tf(num,den)
s = poly(0, "s");

for i = 1:length(num)
    revnum(length(num)-i+1)=num(i);
end

for i = 1:length(den)
    revden(length(den)-i+1)=den(i);
end

numtf=poly([revnum],’s’,’c’)
dentf=poly([revden],’s’,’c’)
numtf;
dentf;
sys=syslin(’c’,numtf,dentf)
endfunction

THE IMAVER FUNCTION FOR SCILAB
// the function takes in an image matrix as the input and gives the averaged image at the output
// written by Srikanth S V , NITC, email : srikanthssv@gmail.com
function [c]=imaver(a)
si=size(a);

for i = 2:(si(1)-1)
    for j = 2:(si(2)-1)
        c(i,j)=a(i,j)+a(i-1,j)+a(i+1,j)+a(i,j+1)+a(i-1,j+1)+a(i+1,j+1)+a(i,j-1)+a(i-1,j-1)+a(i+1,j-1);
        c(i,j)=c(i,j)/9;
    end
end

imshow(c);
endfunction

// NOTE : copy this into any text editor and save it with the corresponding function-name.sci
and then execute the file using the Exec function on SCILAB. Now the function is ready for use.
APPENDIX- B

SCILAB EXTRA's - MUST DOWNLOADs

(The descriptions given below of the softwares are available at their respective homepage.)

RLtool – graphical tool for control system Design and Analysis

Description:

RLTOOL is a graphics based tool for designing compensators for SISO continuous-time plants. An attempt has been made to provide most of the features available in a utility of similar name which is available with a well known commercial software.

Developer:

Rltool 1.7 has been developed by Ishan Pendharkar at the Indian Institute of Technology Bombay, India, with inputs from Jose Paulo Vilela Soares da Cunha, State University of Rio de Janeiro - UERJ - Brazil. The following people provided valuable help and comments: Prof. John Bechoefer, Peter Auer.

Targetted users:

Students and Control System designers using Scilab.

Features:

• Handle Single input Single output continuous time transfer functions.
• Choose between Time Domain and Frequency Domain Design.
• Utility for easy editing of poles and zeros at the click of a mouse.
• Interface for viewing various plots.
• Various settings for plots are accessible to the user.
• Save and Load your plants as required.

The latest version of the tool is available at the link

http://www.ee.iitb.ac.in/uma/~ishan/scilab/rltool.htm
SIP – SIGNAL AND IMAGE PROCESSING TOOL BOX

Description:

SIP stands for Scilab Image Processing toolbox. SIP intends to do imaging tasks such as filtering, blurring, edge detection, thresholding, histogram manipulation, segmentation, mathematical morphology, color image processing, etc. These operations are useful for problem solving in real-world applications ranging from car motion planning to automatic diagnosis of medical images.

Current main authors:

- Zhang Cheng (zhangcheng.johnchain@gmail.com)
- Ricardo Fabbri (rfabbri@if.sc.usp.br).

Features

- I/O of image files in many formats, including BMP, JPEG, GIF, PNG, TIFF, XPM, PCX, and more.
- Numerous functions with flexible interface
- Help pages with examples for all the functions
- Demos

The latest version can be downloaded at this link

http://siptoolbox.sourceforge.net/
ImageMagick – The .dll files of this program are used by SIP toolbox

Description: The SIP toolbox makes use of the .dll files of this open source program. Thus its required to install ImageMagick to perform any Image processing Operation

ImageMagick®, version 6.2.7, is a software suite to create, edit, and compose bitmap images. It can read, convert and write images in a large variety of formats. Images can be cropped, colors can be changed, various effects can be applied, images can be rotated and combined, and text, lines, polygons and ellipses can be added to images and stretched and rotated.

ImageMagick is free software delivered with full source code and can be freely used, copied, modified and distributed. Its license is compatible with the GPL. It runs on all major operating systems.

ImageMagick supports many image formats (over 95) including formats like GIF, JPEG, JPEG-2000, PNG, PDF, PhotoCD, TIFF, and DPX.

Features and Capabilities

Here are just a few examples of what ImageMagick can do:

- Convert an image from one format to another (e.g. PNG to JPEG)
- Resize, rotate, sharpen, color reduce, or add special effects to an image
- Create a montage of image thumbnails
- Create a transparent image suitable for use on the Web
- Turn a group of images into a GIF animation sequence
- Create a composite image by combining several separate image
- Draw shapes or text on an image
- Decorate an image with a border or frame
- Describe the format and characteristics of an image

The latest version of the software can be downloaded at this link

http://www.imagemagick.org/script/download.php
PlotLib - A must have tool if you need your plots to look like MATLAB plots

Description:

This program can be loaded into SCILAB if the user requires the SCILAB plots to look like those on MATLAB.

Developer:

Mottelet Stephane

The latest version of the tool can be downloaded at this link

http://www.dma.utc.fr/~mottelet/myplot.html