

MatPECS—A MATLAB-based Power Electronic Circuit Simulation Package with GUI for Effective Classroom Teaching*

M. VARADARAJAN and SIMI P. VALSAN

Research Scholars, Electrical Engineering, IIT Madras, Chennai, India. E-mail: pvsimi@yahoo.co.uk

Power electronics devices and circuits have revolutionized the concept of power control for power conversion and electrical motor drives. Power Electronics course is essential for electrical engineers and is included in the undergraduate-level syllabus. With various circuit configurations like single-phase/three-phase circuits with R/RL/RLC/RLE loads and associated voltage and current waveforms to be dealt with, it is a tough subject to teach as well as to understand. This has led to a general disinterest in this field among the students. If all the circuit configurations and associated waveforms can be visualized for different operating conditions, the concepts become clearer and interesting. The objective of this paper is to introduce a MATLAB-based simulation package intended to make the classroom teaching of power electronics more effective and easy. MATLAB has proved itself to be a powerful tool for simulation. It is a software package available in almost all the technical institutions. This paper presents a simple menu-driven program that students can use to explore the behavior of power electronic circuits. The paper also explains how the graphical user interface (GUI) feature of MATLAB can be effectively used for making interactive software packages.

INTRODUCTION

MATLAB is a widely used tool in the electrical engineering community. The original concept of a small and handy tool has evolved to become an engineering workhorse widely used for research and design purposes. It is now accepted that MATLAB and its numerous toolboxes can replace and/or enhance the usage of traditional simulation tools for advanced engineering applications. With all these features MATLAB, with or without toolboxes, has come to occupy an indisputable position in the list of available software in all technical institutions. In fact, it has become a *de facto* standard. Since it is already easily available in institutions, in this paper it is proposed to use MATLAB for creating a GUI-based package for effective teaching of power electronics.

Teachers are constantly in search of better methods to teach students. In earlier days, the learning process was a teacher-centered one, whereas, the recent trend is to adopt a student-centered learning process. To make teaching highly efficient and successful, it is required to combine traditional teaching approaches with new learning techniques [1]. In the last few years, slideshows have become more attractive for teaching. In this method of teaching, the teachers do not have to write down anything or draw the complicated waveforms on the blackboard and can concentrate more on his lecture and explanations. But the

disadvantage is that as most of the materials presented are given in the form of handouts, the student here has a very passive role to play. If along with the slide show or lecture the student is provided with a tool to simulate and verify the things he has already learned, then the teaching-learning process becomes interesting.

Surveys of teaching techniques have shown that the traditional blackboard–chalk teaching methods are slowly giving way to modern approaches like slide shows with sound, audio text and movies, numerical simulations, interactive animation, web-based simulations, e-learning, etc. These audio and video files are typically large in size. Students may not have latest computers at home and modem connection to the Internet may have a low transfer rate [2]. But the rapid progress in computer technology as such can help the instructors to teach more effectively using new methods and appropriate software tools and environments. As pointed out in [3], computers by itself cannot educate, but with properly trained educators, they can be efficiently used in the classroom teaching. This paper proposes to tap this potential for teaching of power electronics in technical universities.

Authors firmly believe that a good teacher can never be replaced by a computer. But in the engineering education scenario in South India, there is an acute shortage of well qualified educators. On the other hand, most of the institutions are willing to make large investments in infrastructure development. As a result, there are a lot of engineering institutions with adequate

* Accepted 2 April 2005.

infrastructure, but lacking qualified teaching experts. It is in view of such a scenario that the authors have come up with an idea to put the available infrastructure to better use, thereby aiding the teachers in their classroom lectures. One typical area in which this idea can be efficiently utilized is in the teaching of power electronics.

Power electronics technology encompasses the use of electronic components, the application of circuit theory and design techniques, and the development of analytical tools toward efficient electronic conversion, control, and conditioning of electric power. The typical undergraduate syllabus will have topics like:

- uncontrolled and controlled rectifiers with R, RL, RLE loads;
- choppers;
- single-phase and 3-phase inverters;
- AC voltage controllers, etc.

Power electronics, power electronics systems and especially the associated applications represent a good opportunity for waking new interests in younger generations provided the basics are made clear. There are numerous power electronic simulation packages commercially available such as SABER, CASPOC, PSPICE, SIMPLORER, etc. Some of the packages like PSPICE have the student versions, which are freely downloadable. But the problem is that it requires considerable time and effort from the part of the user to master these packages. Also in the case of these simulation packages, the user is assumed to have good general understanding of power electronic devices and circuits.

In this paper, a specialized teaching software package for Power Electronic circuit simulation based on MATLAB, and GUIDE (Graphical User Interface Development Environment), 'MatPECS' (MATLAB based Power Electronic Circuit Simulation), is presented. Particular factors that support the selection of MATLAB are:

1. MATLAB is a de-facto standard that already dominates the university environment.
2. The flexible software structure of MATLAB comprising libraries, models and programs.
3. MATLAB and GUIDE feature, create a friendly system.
4. Fast-development with MATLAB using powerful calculation and visualization abilities, enable one to expand the software quickly and efficiently without developing any extra programming tools.

It is generally believed that students learn better when they are given an opportunity to implement what they learn. In the traditional method of teaching power electronics, laboratory courses are designed to accompany the theory courses for making the students get better insight in the subject. But in most cases when the students go for laboratory work, they do not have an idea what results to expect and which waveforms to observe.

Because of this, they may sometimes harm themselves or the equipment they use. With the help of this package, the students can simulate the laboratory experiments prior to actually doing the experiments in lab and fully understand the concepts without any danger. MatPECS aims at providing the user with:

- Teaching software using a general-purpose simulation tool—MATLAB.
- A tool to help the teacher in his or her lectures in the classroom, complementing the traditional teaching approach.
- A tool that is user friendly and does not require extra time for mastering.
- A tool that can be used by the students, individually, outside the classroom.
- A tool that helps the students to effectively utilize the laboratory course.

The ultimate aim is to relieve the teacher from drawing the complicated waveforms and to help him/her in concentrating more on the lecture part. It gives the teacher more free time to interact with the students.

MATLAB AND GUIDE

There are essentially three ways in which a user can communicate with the computer via MATLAB [4]: through the Command Window, through the use of scripts and functions, and through GUIs. The *Command Window* is MATLAB's default I/O technique. As the name implies, it is a window into which any standard MATLAB commands or user-defined commands fit. Although the Command Window is adequate for accomplishing simple tasks, it is often useful to create a file containing a list of commands. These files are called *scripts* if they simply run a list of commands, and *functions* if they accept input arguments and/or return output arguments. Both scripts and functions can be executed either directly from the Command Window or from within other scripts or functions. The third communication device, a GUI, provides an intuitive interface between the user and the programming language. A GUI allows the user to bypass MATLAB commands altogether and, instead, to execute programming routines with a simple mouse click or key press. No knowledge of MATLAB or computer programming is necessary for a user to successfully navigate a well-designed GUI, indeed from the user's perspective, the language underlying the GUI is irrelevant. GUIs can range from simple question boxes prompting the user for a Yes/No response, to more complex interfaces. MATLAB provides the user with intuitive tools for easy construction of GUIs.

MATLAB definition for GUIDE is 'an environment having tools that allow FIG-files to be edited interactively'. Using GUIDE is very straightforward and easy—just type GUIDE at

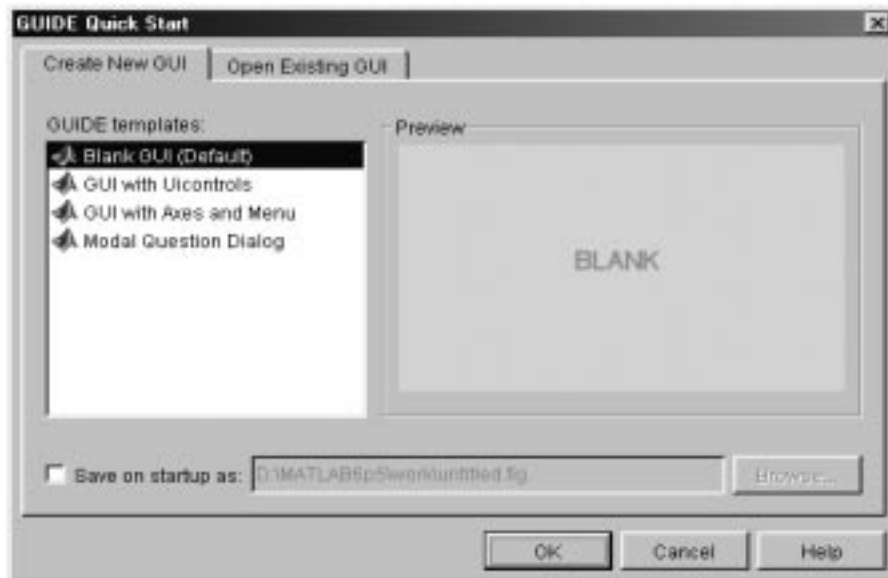


Fig. 1. Blank GUI window.

the prompt in the MATLAB command window and a new GUI window as shown in Fig. 1 will come up. There are four main GUI templates available. To start with, the user can select a 'blank GUI' from the menu [5].

A blank GUI window has a tool palette on the left side of the screen, which consists of tools like push button, text, check box, slider, plot, toggle switch, etc. According to the user's need any tool can be picked and placed on to the blank GUI window in the appropriate places. The properties of each of the items placed on the blank GUI can be edited using a 'property inspector'. Once all the items are ready, the GUI file can be saved. A background program window comes up when the new GUI file is saved. The necessary program statements to be executed behind the various tools are written in this file at the appropriate places. The GUI file is now said to be adapted as per the user's need and can be executed.

MATPECS FOR THE USER

MatPECS is a package developed to facilitate the classroom teaching of power electronics. As a first stage, only the single-phase circuits are introduced. The front end or the user interface is developed using the GUIDE feature of MATLAB. Normally simulators can be divided into two types: circuit simulators and equation solvers. In the case of circuit simulators, the user has to supply the circuit topology and the component values. The simulator internally generates the circuit equations. Equation solvers, on the other hand, require the circuit to be described by means of differential and algebraic equations. The equations should be developed for all the possible states in which the circuit may operate. The 'equation solver' simulation technique adopted here gives

more flexibility and total control over the simulation process and takes lesser execution time [6, 7]. The amount of data input required is a minimum. For example the coding for single phase inverter in square wave mode is given below:

```
global vdc R L fs h theta v0 . . .
i0 f vh v0rms v0lrms . . .
time THD v0rms v01
dt = pi/100;
omega = 2*pi*fs;
thetaz = atan( (omega*L)/R );
impedance = sqrt( R^2 + (omega*L)^2 );
theta1 = 0:dt:pi; % for mode:1
theta2 = pi:dt:2*pi; % for mode:2
theta = [theta1 theta2];
time = theta/omega;
v0 = [vdc*ones(size(theta1)) . . .
-vdc*ones(size(theta2))];
i0 = (4*vdc) / (pi*abs(impedance)) . . .
*sin(theta-thetaz);
v0rms = sqrt(mean(v0.^2));
[f vh ph] = harmonic(v0, h);
v01 = vh(2)*sin(theta);
```

From a user's point of view, MatPECS is a very friendly package. There is no need for the user to understand the programming techniques behind but just follow the directions given and click at the appropriate places and enter the parameter values as the case may be. The user can study the effect of variation in parameters just by changing the appropriate data field values. To start with, the command 'main_gui' has to be typed at the MATLAB prompt to open the main window of MatPECS, which is shown in Fig. 2.

The user can select any one of the listed standard topologies and click on the corresponding push button 'OK'. The window corresponding to the selected topology will pop up. In the window for the selected circuit the required parameter values

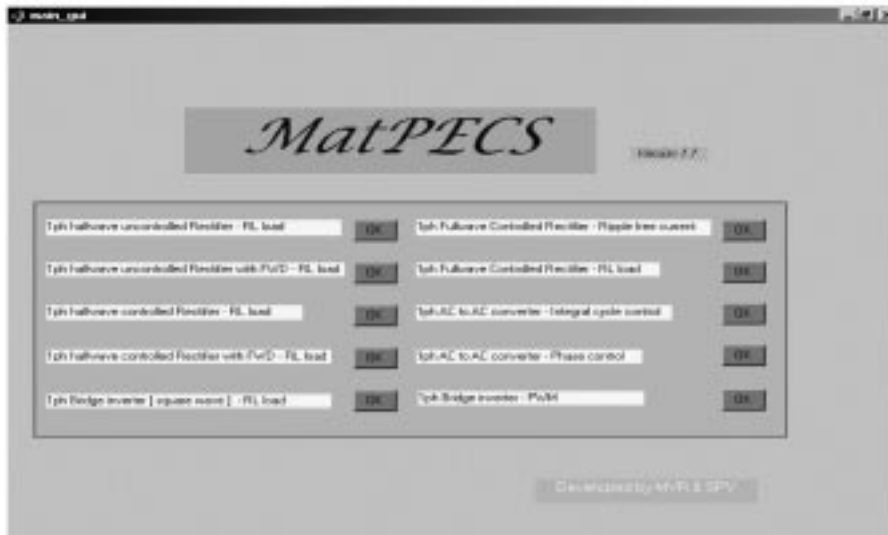


Fig. 2. MatPECS—main window.

have to be given. Clicking on the ‘simulate’ button simulates the circuit. After simulation the various waveforms can be seen in the plot window. The window for the selected circuit with all the parameter values entered and a graph plotted on the plot window is represented in Fig. 3. A typical data read operation for reading load inductance value has the background commands given below:

```
function Load_inductance_Callback
(hObject, eventdata, handles)
global vdc R L fs h theta v0 v01 i0
f vh time
L=str2double(get(hObject,'String'))
```

The simulate button causes the following commands to be executed, where ‘spinverter_sq.m’

is the equation solver-based simulation file for single-phase inverters with square wave operation:

```
function Simulate_Callback
(hObject, eventdata, handles)
spinverter_sq
```

Finally, the results are tied to the buttons as shown below. The example shown is for displaying the output voltage waveform:

```
function output_voltage_Callback
(hObject, ventdata, handles)
global vdc R L fs h theta v0 v01
i0 f vh time v0rms v01rms THD
plot(time, v0, time, v01);
grid
zoom on
```

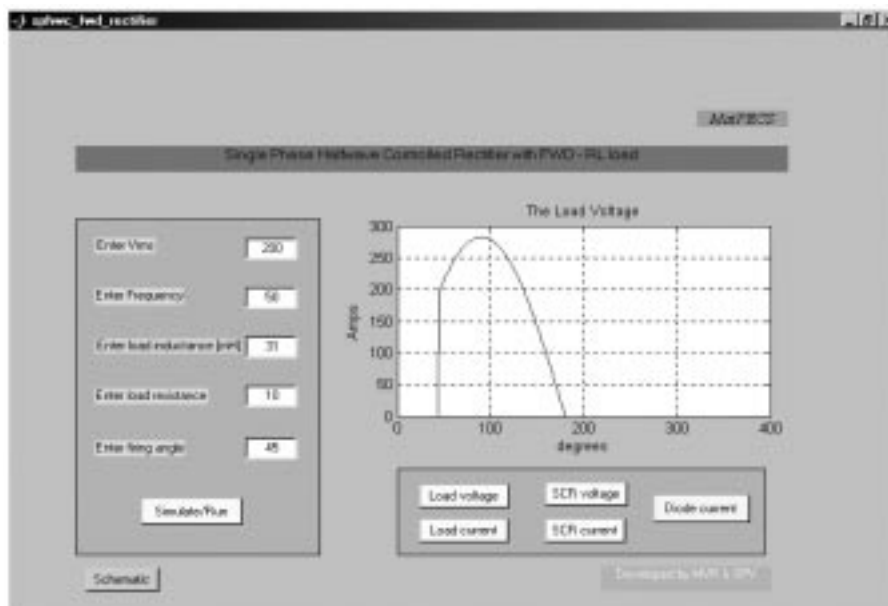


Fig. 3. Typical simulation data and plot.

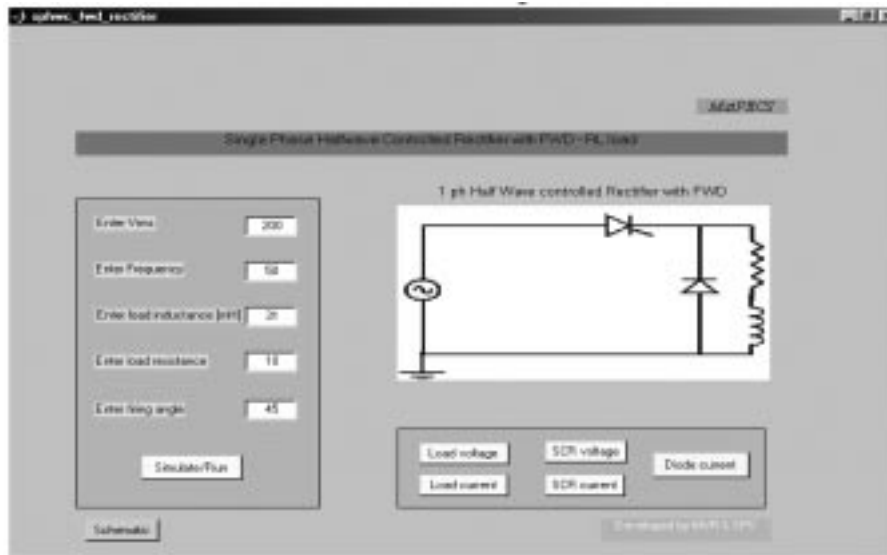


Fig. 4. Circuit details.

```
title('Single phase bridge inverter-  
square wave')  
xlabel('Time')  
ylabel('Output voltage')
```

It is possible to zoom the graph by selecting the required portion with the mouse or by a left click after placing the mouse at the correct position on the plot. A right click brings the plot back to original size.

In case the user wants to get an idea about the circuit arrangement, the circuit can also be seen in the plot window as shown in Fig. 4 by pressing on the 'schematic' button. The MatPECS window for a single-phase bridge inverter with pulse width modulation is given in Fig. 5.

There is provision to do the harmonic analysis and to calculate the total harmonic distortion, THD, of the output voltage. The harmonic analysis is

performed with the help of a function file utilizing the built-in function 'fft(x)', which finds the Discrete Fourier Transform of a given input vector, (x). The harmonic profile along with the display of THD value is shown in Fig. 6.

FUTURE WORK

The package explained here takes only the single-phase circuits into account. A new version including three-phase circuits is under construction, the details of which are not given here, as only the basics were intended to be conveyed. As of now, the circuits displayed are JPEG files loaded for each case given in the main window. Developing a package where the user can draw the circuit and then simulate the same after filling in the



Fig. 5. Bridge inverter with PWM.

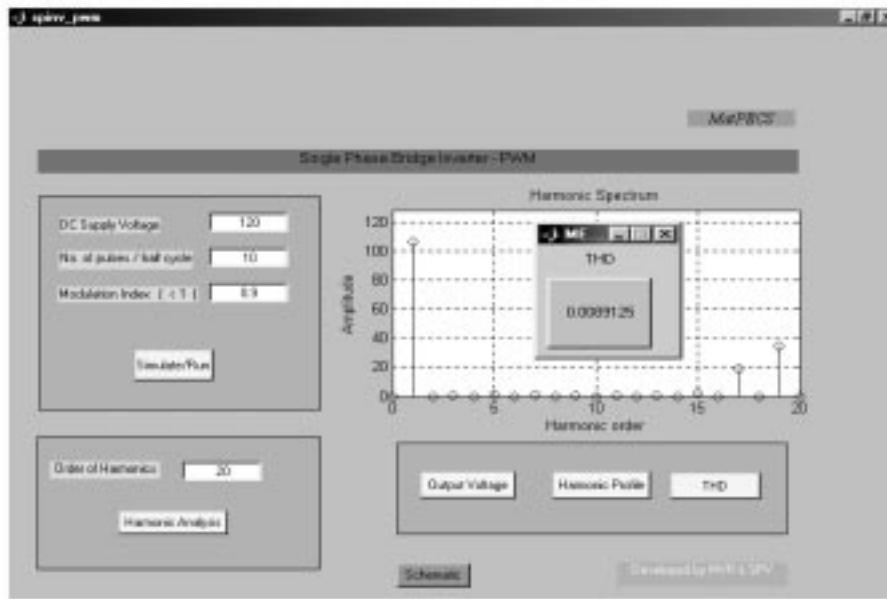


Fig. 6. Harmonic analysis.

parameters in MATLAB GUI environment is aimed to be the future work in this area.

CONCLUSIONS

MatPECS is the outcome of the realization that students need environments that facilitate learning and conceptual understanding of the underlying principles. The aim of this paper is to make use of generally available software and infrastructure in technical institutions and develop application-

specific packages. MatPECS acts as an efficient interface between the teacher and student by making the ideas put forward by the teacher completely clear to the students. The program encourages the students to explore the behavior of the power electronic circuits as the various parameters are changed, to gain insight into the operation of the circuits. It is planned to make the software available in the public domain after incorporating modifications suggested by the users.

REFERENCES

1. M. Mernik and V. Zumer, An educational tool for teaching compiler construction, *IEEE Trans. Education*, **46**(1), Feb 2003, pp. 61–68.
2. Uwe Drogenik and Johann W. Kolar, Survey of Modern Approaches of Education in Power Electronics, Power Electronic Systems Laboratory, Swiss Federal Institute of Technology Zurich. www.ipes.ethz.ch/ipes/paper/iPES_APEC02.pdf
3. C. Roger Schank, Educational Technology: The Promise and The Myth. www1.worldbank.org/education/lifelong_learning/pdf/educational_technology.pdf
4. MATLAB, *Using MATLAB, Version 6.5*.
5. MATLAB, *Using MATLAB Graphics, Version 6.5*.
6. Ned Mohan, Tore M. Undeland and William P. Robbins, *Power Electronics: Converters, Applications and Design*, John Wiley & Sons (1996).
7. M. H. Rashid, *Power Electronics, Circuits, Devices, and Applications*, 2nd Edn, Prentice-Hall (1993).

M. Varadarajan received his BE Degree in Electrical and Electronics Engg. in 1993 from Mepco Schlenk Engg. College, Sivakasi and ME Degree in Power Systems in 1995 from Annamalai University, Tamil Nadu, India. He is currently a research scholar at IIT Madras, prior to which he was working as Senior Lecturer in the School of EEE, SASTRA (Deemed University), Thanjavur, India. His research interests are in the field of Power Quality, FACTS, Voltage Security, Power Electronics and Drives.

Simi P. Valsan is currently a research scholar at IIT Madras. She received her B.Tech Degree in Electrical and Electronics Engg. in 1999 from Govt. Engg. College, Trichur, Kerala and M.Tech. Degree in Energy Systems in 2001 from IIT Madras. From 2001 to 2004, she was a faculty member of the School of EEE, SASTRA (Deemed University), Thanjavur, India. Her research interests are Artificial Intelligence Application to Power Systems, FACTS, Microprocessors, Power Electronics and Drives.