

Using Xcos as a Teaching Tool in a Simulation course

A. LEROS^{1,2} and A. ANDREATOS²

1) Department of Automation
School of Technological Applications
Technological Educational Institute of Chalkis
34400 Psachna, Evia
GREECE

lerosapostolos@gmail.com,
2) Div. of Computer Engineering & Information Science
Hellenic Air Force Academy
Dekeleia Air Force Base
Dekeleia, Attica, TGA-1010
GREECE
aandreatos@gmail.com

Abstract: - Simulation is a powerful tool for understanding and assessing the operation of systems, as well as, natural phenomena. At the Hellenic Air Force Academy a Simulation course is offered to engineering cadets during the 2nd year of studies; it is a fundamental and prerequisite course for future specialisation courses. The ability to model systems --either existing or under design-- in order to better understand their operation under various conditions is mandatory for Engineers. Simulation is a very important learning process because it enables learners to easily experiment with natural concepts, phenomena and systems. This paper proposes Xcos as a teaching tool in our Simulation course.

Key-Words: - Simulation course, educational policy, Matlab, Simulink, Open Source, Octave, Scilab, Xcos.

1 Introduction

Simulation is a powerful tool for understanding and assessing the operation of systems. At the Hellenic Air Force Academy (HAFA), Simulation is offered as a background course and a prerequisite for specialisation courses to the specialisation of Telecommunications and Electronics Engineers. Such courses include: Signal Processing, Radar Systems, Telecommunications, Antennas and Propagation, Queuing Theory and Computer Networks, etc.

In this paper we initially discuss the principles, as well as, the pedagogical value of Simulation. Then we focus on the Simulation course of the Computer Engineering and Information Science Division.

Next, we present the educational policy of our Division and the criteria used for selecting the most appropriate software for the Simulation course. Then we describe a successful gradual migration from proprietary software (Matlab) to a Free and Open Source Software (FOSS) alternative, i.e., Octave. In this process, a small gap remained: the lack of a graphical simulation environment like Matlab's companion, Simulink. Thus, the purpose of this paper is to examine Xcos, a tool embedded in Scilab, another Open Source alternative to Matlab.

2 Basic concepts of Simulation

The simulation course is useful for engineering students because it offers fundamental concepts, methods and tools for simulating the behaviour of the various transistor models. Simulation means to view an object from a specific viewpoint where certain parameters count more. Let us now briefly define the fundamental principles of Simulation.

System is an object or a collection of organically interconnected objects, whose properties we would like to study.

1. **Experiment** is the process of applying signals at the inputs of a system in order to extract information by studying its outputs or by observing and measuring its behaviour.
2. **Model** of a system is an abstract mathematical or simplified physical construct which we consider that it represents the system, i.e. behaves like it under specific circumstances or conditions. The use of a model is to give us relatively easy indications of the behaviour of the real system. The study of the behaviour of a

system is achieved by controlling specific parameters. The creation of models is simplified significantly with the use of special modelling software. The mathematical models and the simulation techniques are presented in the following Fig. 1:

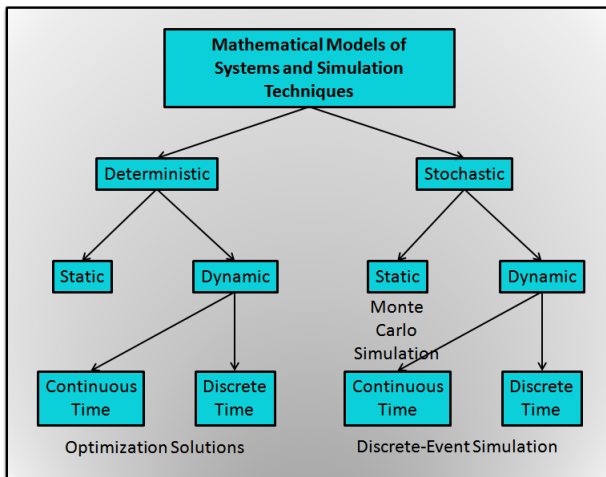


Fig. 1: Models and simulation techniques

Usually the simulation is applied to stochastic mathematical models with the Monte Carlo method if these are static or with the form of Discrete Events if these are dynamic. In case the models are not stochastic but deterministic, then we use optimisation instead of simulation.

3. **Simulation** is a controllable experiment performed on a model of a (stochastic) system.
4. **Simulation with computers** is the process of designing a model of a real system, the implementation of this model with a computer program, and the performance of experiments on this model with the aim to understand the behaviour of the system or for the valuation of strategies for its operation.

We study systems for various purposes, such as:

- a) To understand the physical world and to build our knowledge.
- b) To satisfy our scientific curiosity, i.e. understand better the nature (natural science viewpoint).
- c) To build them or to improve them (engineering point of view).

The major reasons for performing a simulation are:

- a) Pedagogical reasons: Effective learning through

the “construction of knowledge” (Andreatos & Leros, 2010a). The simulation gives us the ability to experiment with the model of a system in the classroom and/or at home, in order to understand better the physical meanings.

b) Economic reasons: The experiment is economically very expensive. Example: The determination of the endurance of airplanes or spacecrafts by constructing them and then by crashing them through experimentation is an uneconomical method to obtain information.

c) Security reasons: The experiment sometimes is dangerous: The training of nuclear plants operators for crisis management by driving the nuclear reactor in risky operating conditions is not a safe method.

d) Design of new systems: The system in need for experiments might not yet exist: This is part of the process to design new systems yet under construction.

e) Suppression of disturbances: In a simulation of a model it is possible to suppress the disturbances which might not be avoided in measurements in the real system.

f) Simulation for redesign and validation of analytical models of systems (Fig. 2): Using the results of simulation it is possible to modify a model of the system in order to incorporate in it details which originally were not included. It is also possible to conduct various types of analyses.

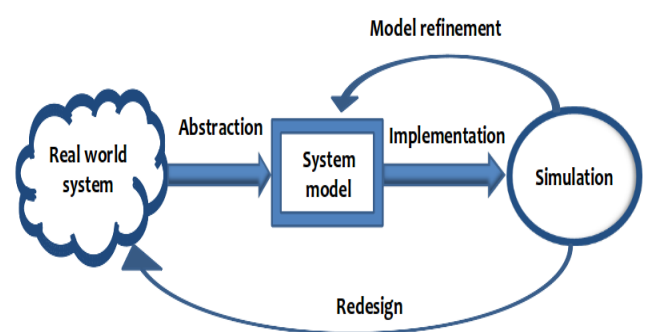


Fig. 2: The role of simulation in redesign

g) Validation of analytic modelling: The analytic modelling of systems often demands assumptions for simplification which make the results suspicious until these have been verified by other techniques, such as simulation. Fig. 3 shows the role of simulation to the validation of an analytical model.

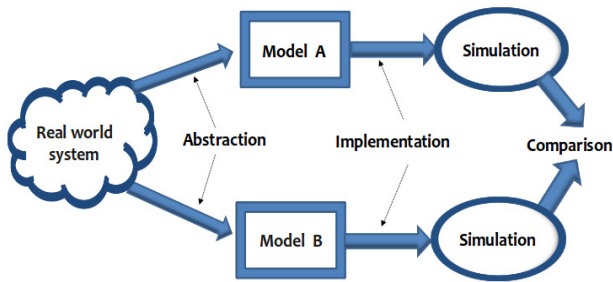


Fig.3: Simulation for validating analytical models

Examples of experiments or simulations:

- Simulation of the behaviour of vehicles, e.g., an airplane or automobile, for the purpose of operator training.
- Simulation of a manufacturing process, such as metal or paper, for learning the behaviour of the process under various operating conditions and for its improvement.
- Simulation of a simplified model of a computer network, for studying its behaviour of performance under different demanding loads.

Two representative sites containing a lot of simulation experiments are Interactive Physics [6] and Interactive Science Simulations [7].

The pedagogical value of Simulation is its ability to implement modern learning theories in education such as constructivism, collaborative learning, social learning and learning by discovery.

3 Simulation software and engineering education in HAFA

HAFA simulation course syllabus needs a software package with the following specifications (Andreatos & Leros 2010b):

- Full coverage of the Simulation course syllabus.
- Existence of specific toolboxes supporting specialisation courses.
- Operation on Windows as well as Linux platforms. Most courses of the Computer Engineering and Information Science Division are offered on Linux platform.
- The power of the programming language. All Matlab-like languages share this advantage. Thus, high Matlab compatibility matters.
- Wide user community and technical support.
- Steep learning curve.
- Compliance with software compatibility rules.
- User-friendly Graphical User Interface (GUI).

- Existence of bibliography in Greek and English.
- Cost effectiveness.

Taking all of the above into account, as well as, the existing instructor's asset, we have chosen GNU Octave as the best software for our needs (Andreatos & Leros, 2010a; 2010b). In (Leros, Andreatos & Zagorianos, 2010) we demonstrate the high compatibility of Octave-Matlab using three examples which use the same code.

4 Educational policy of the Computer Engineering Division

The use of Free and Open Source Software (FOSS) has gained considerable attention recently in many universities (Coll et al., 2008). Since 2008, the Computer Engineering and Information Science Division of the Hellenic Air Force Academy has begun introducing FOSS in an effort to cover educational and infrastructural needs. Computer Simulation is one of our courses in which we have been experimentally using FOSS software during the past two academic years. Simulation is a fundamental course for Engineers, as well as, a prerequisite for future courses; a good engineer should be able to model various systems, either existing or under design.

Two additional issues of our policy are: a) student learning experience with different software tools and programming styles, b) independence of specific products and suppliers (Andreatos & Leros 2010b). For the above reasons, the current trend in our educational policy is to use many different software products in our courses, and FOSS is a convenient option.

5 The migration from Matlab to Octave

During the past ten years, following international trends, Matlab software was introduced for use in the Simulation course; however, evolving educational policy of the Computer Engineering and Information Science Division offering this course as well as other reasons such as cost, dictated the examination of Open Source alternatives as the principal course software (Andreatos & Leros, 2010a). After a literature review (Andreatos & Leros, 2010a; 2010b), we decided to test Octave.

One of the reasons for this decision was the high compatibility of Octave to Matlab, as far as the command set is concerned. This was very important because it would: a) achieve instructors' consensus;

b) allow instructors to get acquainted and start using the new software in a minimal amount of time; c) re-use notes, problems, projects and code developed for Matlab during those years, what we define as *the instructor's asset*; d) use books written for Matlab, some of which in Greek, the official language of studies; e) achieve smooth transition in the class and student acceptance.

Starting from the fall 2008 semester, we began using both Matlab and Octave in the simulation course. This way we could achieve at least three things: a) compare their possibilities and features in relation to the course syllabus; b) check their compatibility; c) check student acceptance.

By the end of the academic year 2010-2011 (July 2011), we had successfully fulfilled the migration plan from Matlab to Octave, described in (Andreatos & Leros, 2010b). However, a small gap remained: we had no substitute for Matlab's companion, Simulink, a powerful graphical simulation environment. Simulink incorporates special modules for various disciplines, such as Simevents, SimPowersystems, SimMechanics and Stateflow. These are based on the "G" visual programming language which enables the user to easily and quickly design a system without the knowledge of a typical programming language. The gap was small because there was very little time left at the end of the course. However, a graphical simulation tool might prove useful in coming courses as well as, the thesis, and is a great teaching tool (Yousuf, Luo & Huang, 2004).

Thus, after further research we discovered that the Xcos tool of Scilab, another FOSS alternative to Matlab, could nicely substitute Simulink. So the new educational plan of the Simulation course which we are running this semester (fall 2011) is: a) to teach the basic theory in Octave/ QtOctave on Linux platform and then: b) jump to the Xcos tool of Scilab by the end of the course.

We did not migrate totally to Scilab for three reasons: a) a considerable amount of educational material on Octave has been developed in the meantime (Leros & Andreatos, 2010); b) we are currently investigating the compatibility of Scilab commands to Octave/ Matlab commands; c) we need the Octave toolboxes for future courses, to which Scilab has no substitute. We also note that a student familiar with the Octave command window can easily migrate to the Scilab command window.

6 Xcos simulation tutorial

This section presents a tutorial demonstrating a small part of the possibilities of the Xcos. The purpose of this tutorial is twofold: first, to demonstrate the power of a graphical simulation tool; second, to compare Xcos to Simulink.

Many common systems are presented by a linear differential equation of second order. Examples: a) An electrical circuit (two-ports) consisting of an inductor, a resistance, and a capacitor in series. b) A sphere tied to a spring on a level (plane) with friction, on which a force is applied to it. c) A locomotive (kinetic) system of a motor engine (Yousuf, Luo & Huang, 2004).

It can be determined (Yousuf, Luo & Huang, 2004) that the equations of all the above systems can finally be presented by equation (1):

$$\frac{d^2 y(t)}{dt^2} + 2\zeta\omega_n \frac{dy(t)}{dt} + \omega_n^2 y(t) = \omega_n^2 u(t) \quad \text{Eq. 1}$$

where the variables $y(t)$ and $u(t)$ represent the output and the input of the system respectively, ω_n is its natural frequency and ζ is the damping coefficient.

In the sequel, for pedagogical reasons, the mechanical systems such as (b) and (c) above are examined. For the numerical solution of equation (1) in Xcos Scilab the system is designed as in Fig. 4. For the sake of comparison we also provide the corresponding Simulink diagrams.

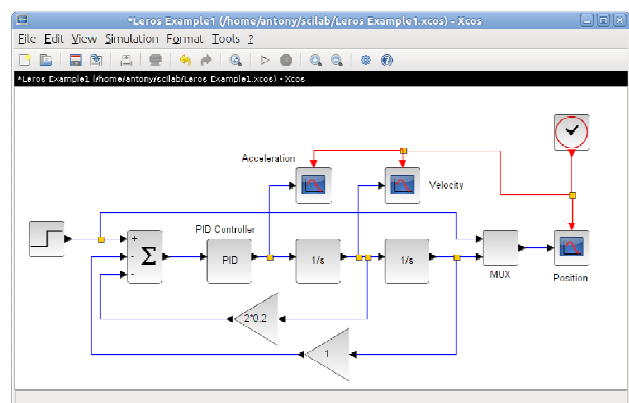


Fig. 4: Model of a second order linear differential equation in Xcos

In the above figure, with reference the mechanical system, $y(t)$ is the displacement, $dy(t)/dt$ is its velocity and $dy^2(t)/dt^2$ is its acceleration. Choosing $\zeta = 0.2$ and $\omega_n = 1$ and placing scopes at appropriate points (see Fig. 4) we obtain the following results (Fig. 5, 6 and 7):

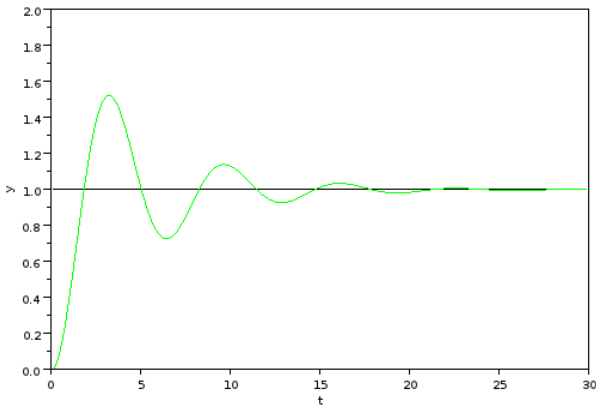


Fig. 5a: Position versus time (Xcos)

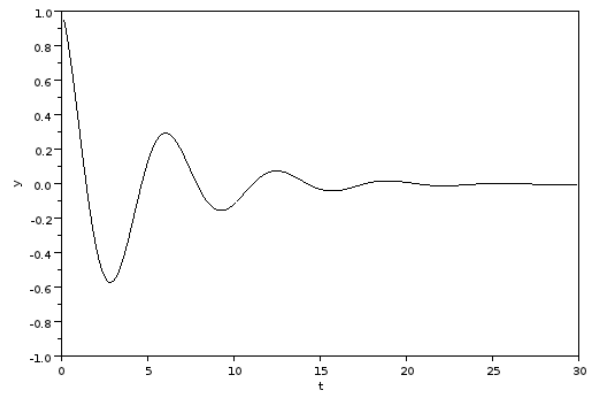


Fig. 7a: Acceleration versus time (Xcos)

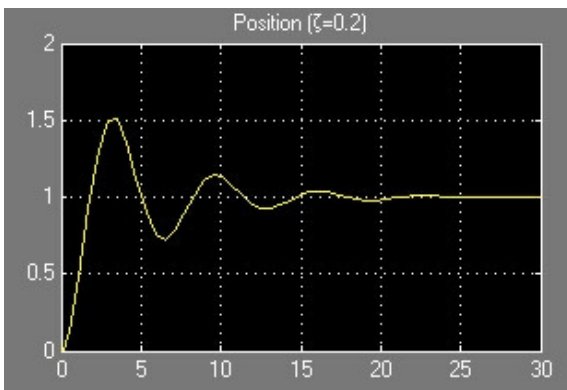


Fig. 5b: Position versus time (Simulink)

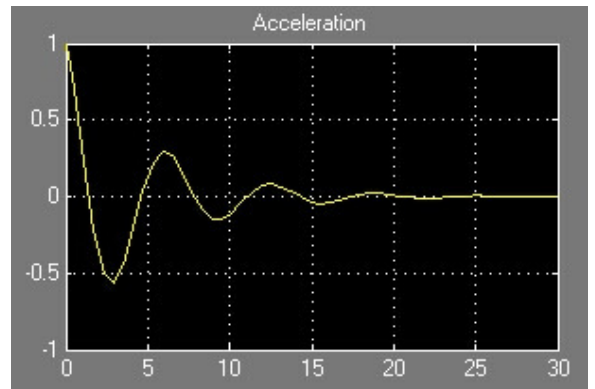


Fig. 7b: Acceleration versus time (Simulink)

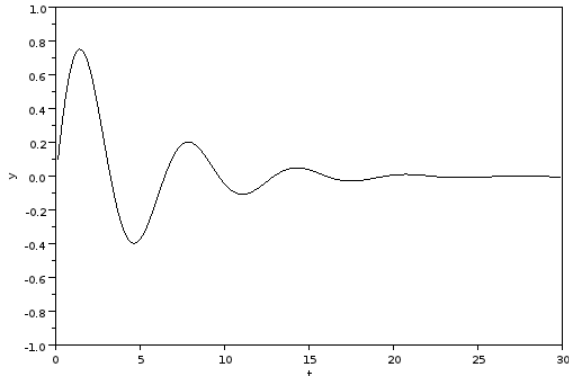


Fig. 6a: Velocity versus time (Xcos)

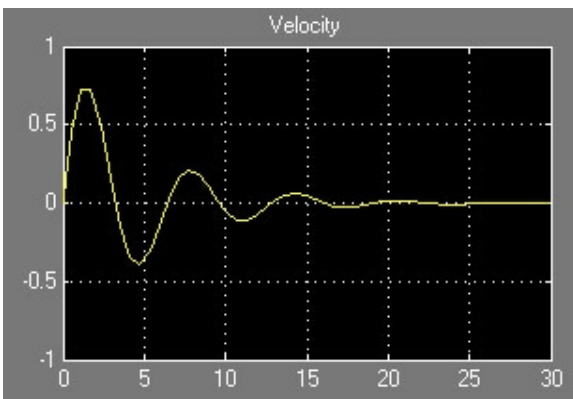


Fig. 6b: Velocity versus time (Simulink)

The results verify the theory of oscillations of the high school or college physics books. The major advantages of simulating such systems with the aid of Xcos are the following: a) The design of the model in Xcos is easy (visual) and does not require programming knowledge. b) As soon as the trainees acquire the model, they can very easily experiment with it, to build their personal knowledge; thus, a visual simulation tool such as Xcos or Simulink constitutes a great learning and teaching tool, (Yousuf, Luo & Huang, 2004) according to modern learning theories used in education, such as constructivism and learning by discovery.

Indeed, the regulation of the two Gain components in Figure 4 above is equivalent to changing the coefficients $2\zeta\omega_n$ και ω_n^2 of equation (1). Double clicking the gain blocks, the gain coefficient appears and we can easily enter any value. Consequently, the possibility is provided to the trainee to experiment and study the impact of the coefficients of the system on the position, the velocity and the acceleration of the object. According to the theory of oscillations, when ζ approaches 0, the system falls in oscillations. For pedagogical purposes, we give the position of the object for two more (small) values of ζ , specifically: 0.05 and 0.01 in Figure 8:

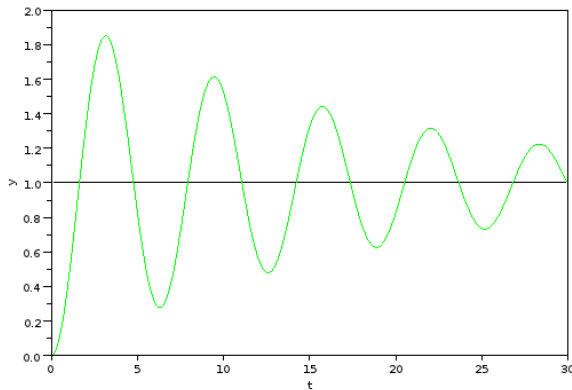


Fig. 8a: Making ζ smaller drives the system to oscillations (Xcos)

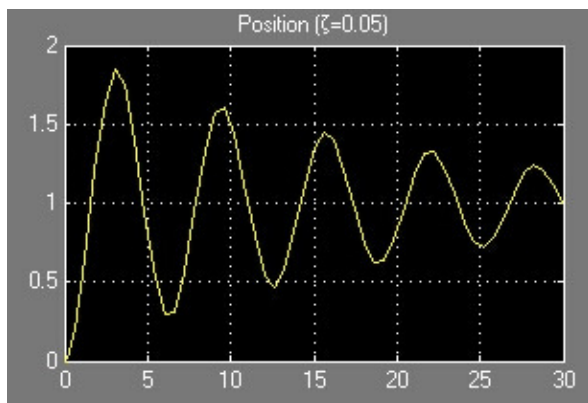


Fig. 8b: Oscillations in position (Simulink)

7 Conclusions

In our effort to examine the use of FOSS in education, we designed and implemented a 3-year migration plan for the Simulation course, starting September 2008.

Taking into account the syllabus of the Simulation course, as well as, the specialisation courses taught at the next years, we decided to use Octave plus the QtOctave GUI.

The transition from Matlab to Octave presented no difficulties for the HAFA students. Octave was a good alternative because of its toolboxes, high programming language compatibility and the QtOctave GUI. An evaluation questionnaire verified student acceptance (Andreatos & Leros, 2010b).

The next step, described in this paper, is the substitution of Simulink with the Xcos tool of Scilab. Evaluation results presented here show that this migration will also be successful because of the high compatibility of Simulink – Xcos function and libraries. Neither Octave nor Scilab can fully substitute Matlab with Simulink, because: a) Octave language is highly compatible to that of Matlab and

this was considered an advantage for the reasons explained in (Andreatos & Leros, 2010a; 2010b). Octave has a huge set of toolboxes but no tool like Simulink. b) Scilab has a good visual programming tool, namely Xcos, which looks comparable to Simulink, although less “polished”. The tutorial has shown not only that Xcos suffices for our purposes, but also, that it constitutes a great teaching and learning tool, for the following reasons: it uses visual programming, hence it does not require advanced programming skills; it enables the design and simulation of systems in a few minutes; we can easily test the effect of various parameters on system behaviour and performance.

The use of more than one software products in a single course is not considered a problem but rather an advantage, which, according to our educational policy, makes future engineers eager to test and use new tools, instead of being stuck to only what they have used in the past.

References:

- [1] A. Andreatos and A. Leros, Use of GNU Octave in the Simulation course of the Hellenic Air Force Academy, *Proceedings of the 1st International Conference on the use of FOSS in Education*, Chania, Crete, Apr. 16-18, 2010a (in Greek). Available online from: <http://www.foss4edu.gr/praktika/papers/paper05-teliko.pdf>.
- [2] A. Andreatos and A. Leros, Simulation course redesign and educational software selection, *Proceedings of the 9th European Conference on e-Learning (ECEL 2009 Conference)*. 4-5 November 2010b, Porto, Portugal.
- [3] H. Coll, D. Bri, M. Garcia and J. Lloret, Free software and open source applications in higher education, *Proceedings of the 5th WSEAS / IASME International Conference on Engineering Education, Heraklion, Greece*, July 22-24, 2008.
- [4] A. Leros, A. Andreatos and A. Zagorianos, Matlab - Octave science and engineering benchmarking and comparison. *Proceedings of the 14th CSCC*, Corfu, Greece, July 22-24, 2010.
- [5] A. Leros and A. Andreatos, Introduction to the Octave arithmetic software package. Athens, Greece, May 2010 (free eBook in Greek). Available online from: <http://t-h.wikispaces.com/octave>.
- [6] Interactive Physics: <http://www.design-simulation.com/ip/index.php>.
- [7] Interactive Science Simulations : <http://phet.colorado.edu/en/simulations/category/new>
- [8] A., Yousuf, J. Luo and C. L. Huang, Using the SIMULINK as a Teaching Tool. *Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition*.