ANNALS of the ORADEA UNIVERSITY.
Fascicle of Management and Technological Engineering, Volume X (XX), 2011, NR1

AUTOMOTIVE SUSPENSION TEACHING USING MATLAB

Dumitru Nicoara
Transilvania University of Brasov, tnicoara@unitbv.ro

Keywords: vehicle suspension, vibration control, engineering educations, MATLAB

Abstract: In this paper, we have described a successful approach to enhancing an advanced engineering automotive course via the addition of numerical simulation using Matlab. The main objective of the paper is to show the usefulness and power of Matlab in modeling and analyzing dynamic behavior of vehicle subsystems. The second purpose of this paper is to present the advantages of using Matlab to simulate these systems. Tutorial problems in arrears of analysis and control of suspension responses for the quarter car models are used.

1. INTRODUCTION

An advanced engineering mechanics course teaches students to analyze a variety of dynamical systems using Newtonian and Lagrangean mechanics approaches. The modeling task typically produces linear and nonlinear differential equations that are best solved numerically. The systems with more than two degrees of freedom require considerable analytical and computational effort. In order to prepare students to competently solve these systems numerically, the students must master a suitable programming environment. The user must develop computer programs in FORTRAN or C++. These programming languages are high level languages which require the user to have advanced knowledge and ability to develop subroutines that can resolve differential equations, curve-fitting, matrix manipulation, and graphics. These difficulties can be avoided by using friendly software programs, easy to learn and use.

For proper analysis, design and control of mechanical systems we have software like Matlab and Maple. Unfortunately, often using them in courses taught mechanical engineering students is limited to presenting some aspects of programming without being used for modeling, analysis and design of dynamic systems.

MATLAB® (Matrix Laboratory) is a commercially available software package originally developed in the seventies by Cleve Moler, [4], for convenient numerical computations, especially matrix manipulations. Today it has grown to a high-level technical computing language and interactive environment, for algorithm development, data visualization, data analysis, and numerical computation, widely used in the academic world and in industry, [2]-[5].

MATLAB® is a software package for high performance computation and visualization [3]. The combination of analysis capabilities, flexibilities, reliability and powerful graphics makes MATLAB the premier software package for engineers and scientists [2], [3]. MATLAB provides an iterative environment with more than hundreds of reliable and accurate built-in mathematical functions. These functions provide solutions to a broad range of mathematical problems including: Matrix Algebra, Complex Arithmetic, Linear Systems, Differential Equations, Signal Processing, Optimization, Control and other types of scientific computations [3], [8], [10], and [11].

In this paper, the need for MATLAB as a pedagogical tool in engineering research and teaching is highlighted. For example, the studies of an automobile suspension system, automobile steering system, car lateral and longitudinal control and vibrations control can be easily done in MATLAB environment.

Tutorial problems in arrears of analysis and control of suspension responses for the quarter car models are used to validate the enormous potentials of MATLAB.

2. MATHEMATICAL SIMULATION IN MATLAB

An important component of an advanced dynamics course is solving the derived equations of motion analytically and numerically. There are many approaches for implementing ODE solutions in MATLAB, and no doubt, many approaches to teaching the process. Matlab contains a complete suite of solvers for simulating ODE's and DAE's. The user can either implement his model in Simulink as a block diagram or directly in Matlab as mathematical equations.

The standard form used by all Matlab solvers is

$$\mathbf{M}(\mathbf{x},t)\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},t) \quad (1)$$

were *M* is a state and time dependent mass-matrix and *f* is a nonlinear state and time dependent vector valued function. This general formulation allows specifying a wide variety of differential equations into a standard form, [5], [10].

All Matlab solvers have the same calling interface, but the user should know the basic solver properties in order to select the most appropriate routine for a specific problem. The following paragraphs give some guidelines concerning the implementation of differential equations for the simulation of physical systems, following from the experience obtained by author in his didactic and research activity.

The following solvers are suitable for the simulation of non-stiff problems, i.e. of which the eigenvalues of the Jacobean matrix are all of the same order of magnitude. These solvers are ODE integrators, which require a nonsingular mass-matrix.

• ode45: based on an explicit Runge-Kutta formula. It is a one-step solver, which only requires a solution at the immediately preceding time instant. In general, ode45 is the best function to apply as a first try for most problems.

• ode23: based on an explicit Runge-Kutta. It may be more efficient than ode45 at crude tolerances and in the presence of mild stiffness. Like ode45, ode23 is a one-step solver.

• ode113: variable order Adams-Bashforth-Moulton solver. It may be more efficient than ode45 at stringent tolerances and when the ODE function is particularly expensive to evaluate. ode113 is a multistep solver, which requires solutions at several preceding time instants to compute the current solution.

We have found that at this academic level, it is adequate to portray ODE solvers as "black boxes" and defer discussion of the details of numerical ODE solution to other courses, Figure 1. The ODE solvers are presented as black boxes that take as input ODE and initial conditions and produce as output functions that satisfy the ODE and initial conditions.

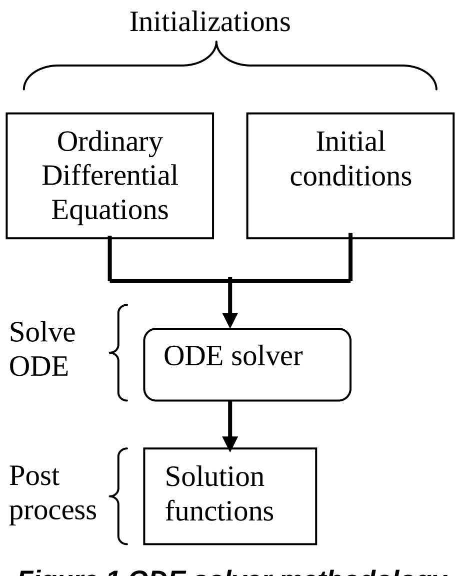

Figure 1 ODE solver methodology

3. TUTORIAL PROBLEMS

The main objective of the paper is to help students understand and appreciate principles and concepts of automotive vibrations, modeling and control, through an effective integration of software programs, MATLAB and SIMULINK, with theory.

An automobile is made up of many components. These components include suspension, engine and its components, chassis, transmission, brakes, etc. and represent the many subsystems in a multi-degree of freedom analysis. Although the treatment of automobile suspension system is a standard application of vibration theory, the application of MATLAB and SIMULINK to it is an original frame work.

3.1. Half car suspension model

3.1.1 Mathematical model

The model of a half-car suspension system is shown in Figure 2, [1], [6], [7].
The model is represented as a linear four DOF system. It consists of car body (single sprung mass) connected to front and rear wheels (two unsprung masses) at each corner. The sprung mass is free to heave and pitch. The sprung masses are free to bounce vertically with respect to the sprung mass. The suspensions between the sprung mass and unsprung masses are modeled as linear viscous dampers and spring elements, while the tire is modeled as simple linear spring without damping characteristic

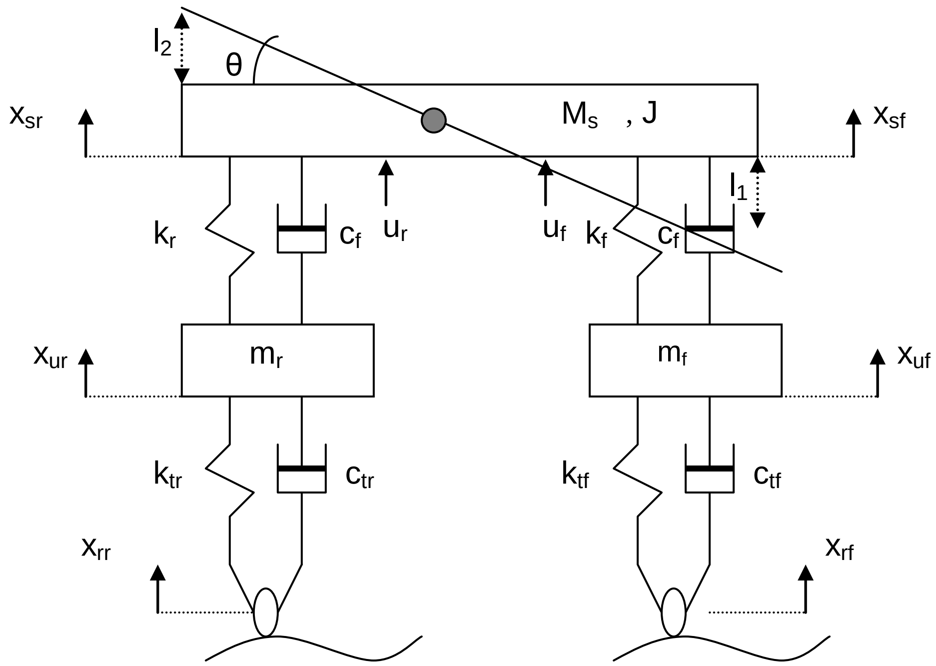

Figure 2 Half car passive suspension models

The system state variables can assign as:

$x_1 = x_{sf}$, front sprung mass displacement

$x_2 = \dot{x}_{sf}$, front sprung mass velocity

$x_3 = x_{uf}$, front unsprung mass displacement

$x_4 = \dot{x}_{uf}$, front unsprung mass velocity

$x_5 = x_{rf}$, rear sprung mass displacement

$x_6 = \dot{x}_{sf}$, rear sprung mass velocity

$x_7 = x_{uf}$, rear unsprung mass displacement
$x_8 = \dot{x}_{uf}$, rear unsprung mass velocity

The state vector is:

$$[\mathbf{x}] = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8]^T \tag{2}$$

The state equations with these assigned state variables for half –car suspension system is as follows:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} = \mathbf{Cx} + \mathbf{Du} \end{cases} \tag{3}$$

where matrices A, B, C, D are:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -a \cdot k_f & -a \cdot c_f & a \cdot k_f & a \cdot c_f & -b \cdot k_r & -b \cdot c_r & b \cdot k_r & b \cdot c_r \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{k_f}{m_f} & \frac{c_f}{m_f} & -\frac{k_f + k_{tf}}{m_f} & -\frac{c_f}{m_f} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -b \cdot k_f & -b \cdot c_f & b \cdot k_f & b \cdot c_f & -d \cdot k_r & -d \cdot c_r & d \cdot k_r & d \cdot c_r \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{k_r}{m_r} & \frac{c_r}{m_r} & -\frac{k_r + k_{tr}}{m_r} & -\frac{c_r}{m_r} \end{bmatrix} \tag{4}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ a & b & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & \frac{k_{tf}}{m_f} & 0 \\ 0 & 0 & 0 & 0 \\ b & c & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{m_r} & 0 & \frac{k_{tf}}{m_r} \end{bmatrix} ; \; \mathbf{C} = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \; ; \; \mathbf{D} = [0 \quad 0 \quad 0 \quad 0] \; ; \tag{5}$$

and *a*= (1/M_S) + ((l_1*l_2)/J), *b*= (1/M_S)-((l_1*l_2)/J), *d*= (1/M_S) + ((l_1*l_2)/J).

The matrix $\mathbf{u} = [u_f \quad u_r \quad x_{rf} \quad x_{rr}]^T$ is an input matrix, where u_f ,u_r are the inputs active forces u_f ,u_r and x_{rf} ,x_{rr} are road disturbances.
The response of all 8 - state variables can be obtained using Matlab and above mathematical model.

In the numerical simulations with MATLAB, the active forces have not been considered.

3.1.2 Time response analysis -MATLAB solution for passive suspension

The time response analysis is done by using Matlab software using Laplace Transformation and State Space with 4-DOF model given by equations(1)-(5).
The road surfaces vary from one place to another, the road surface is sometime bumpy and sometimes there are holes in it. When a passenger car passes through these bumps and holes the excitation input is "Step Response" which leads to vertical vibrations in car.

The damping coefficient can be varied from value zero up to the optimum value to suggest the range for optimized response.

The MATLAB analysis to the car suspension described above is obtained by generating a Matlab script. The MATLAB script can be used to study the sensitivity of the system to the variation of the parameters. The script plot the position of the car and the wheel after the car hits a "unit bump" (i.e. a unit step) for following values of suspension damping:

$c_f = c_r = 500;\ c_f = c_r = 1500;\ c_f = c_r = 3000;\ c_f = c_r = 4000\ N.s/m.$

MATLAB script **–**example for Amplitude Response of Front Sprung Mass

```
Ms=600 ; mf=45 ; mr=45 , J=730 ;l1 =1.29;l2 = 1.26;kr =18000;
kf=18000;ktf= 180000; ktr =180000;
a= (1/Ms)+((l1*l2)/J), b= (1/Ms)-((l1*l2)/J) ,d=(1/Ms) +((l1*l2)/J);
c=[0.5 1.5 3 4]*1e3;
Y=[];
for n=1:4
A=[ 0 1 0 0 0 0 0 0; -a*kf -a*c(n) a*kf a*c(n) -b*kr -b*c(n) b*kr b*c(n)
; 0 0 0 1 0 0 0 0; kf/mf c(n)/mf -(kf +ktf)/mf ...
-c(n)/mf 0 0 0 0; 0 0 0 0 0 1 0 0; -b*kf -b*c(n) b*kf b*c(n) -d*kr...
-d*c(n) d*kr d*c(n); 0 0 0 0 0 0 0 1;...
0 0 0 0 kr/mr c(n)/mr -(kr + ktr)/mr -c(n)/mr];
B=[ 0 0;0 0; 0 0;ktf/mf 0;0 0;0 0;0 0;
0 ktr/mr];
C=[1 0 0 0 0 0 0 0 ];
D= [0 0];
%G = ss(A,B,C,D);
G1= ss(A,B(:,1),C,D(:,1));
t=linspace(0,2,1000)';
y=step(G1,t);
Y=[Y y];
figure(1), plot(t,Y)
xlabel('Time (s)'); title('Front sprung masses')
ylabel('Amplitude');
legend('c=500', 'c=1500', 'c=3000','c=4000')
end
```

The MATLAB script can be used to study the effect of the system parameters on the performance of the system. From Figures 3-4, it is seen that a family of curves result as we vary the value of the damping. An important problem is to find the value of damping c so that the car passengers have maximum of comfort. Passenger is sensitive to sprung mass acceleration and to frequency oscillations.
For half car model we was suggested after time and frequency response analysis that damping value should be in range of 2000 N.s/m – 3000 N.s/m. To obtain the front

unsprung mass displacement, Figure 4, we substitute the matrix C, from above matlab script with matrix C=[0 0 1 0 0 0 0 0].

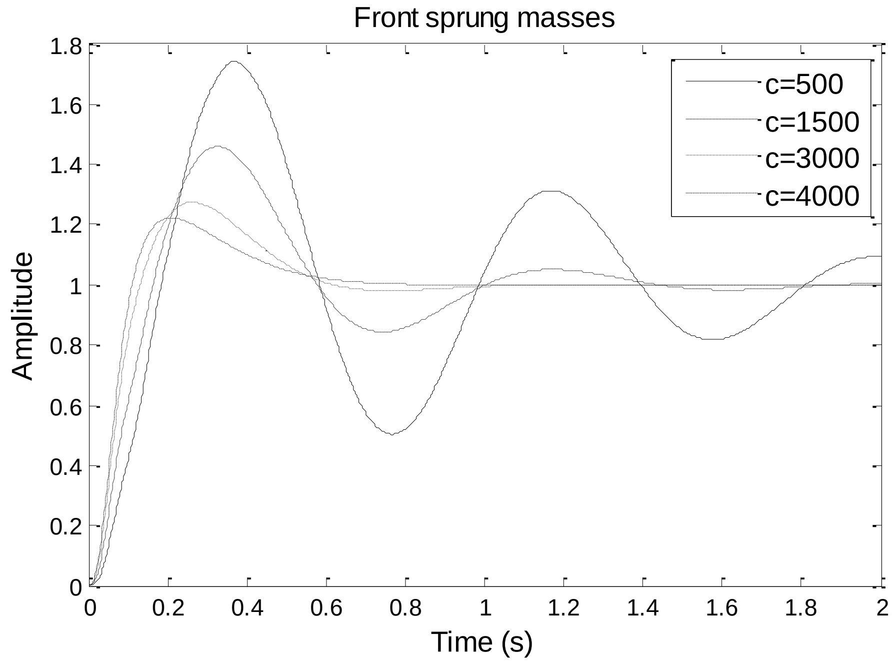

Figure 3 Amplitude Half car passive suspension model

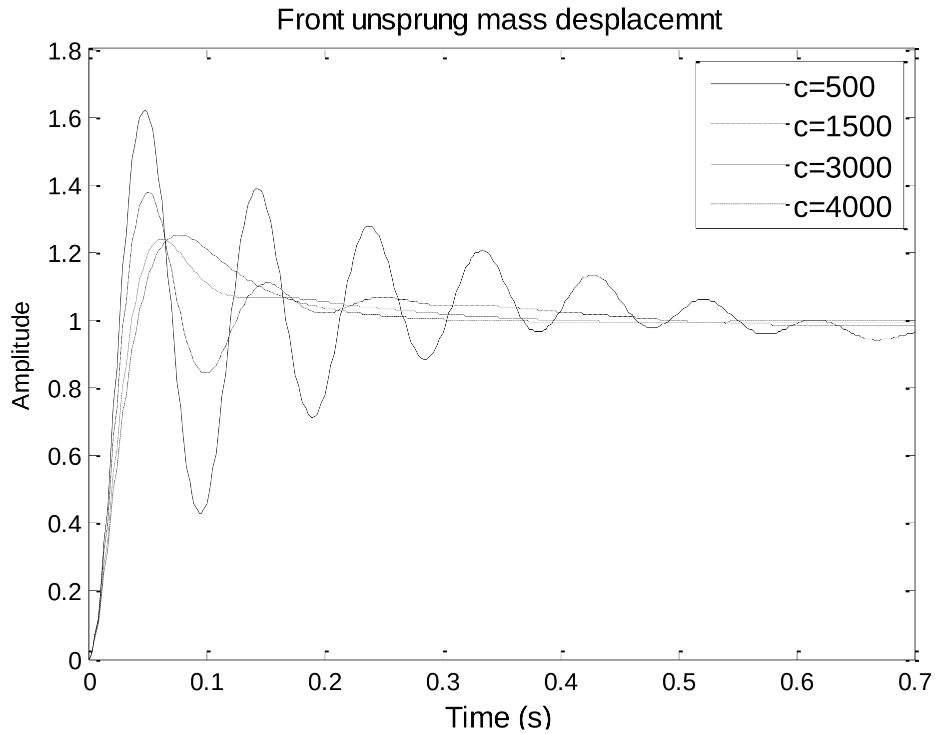

Figure 4 Front unsprung mass displacements

3.1.3 Frequency response analysis -MATLAB solution

The frequency response for a half car model has been analyzed in this section. The frequency response for front sprung and rear unprung masses has been obtained using Matlab sequence: Bode (G1*.1) and Bode (G2*.1).

From Figures 5 and 6, it is noted that for frequency response of front sprung mass the peak resonant frequency is about 250 db. However as the damping value is increased from 0 N.s/m to 1500 N.s/m, the peak is decreased up to 20 db which is acceptable, [1]. (

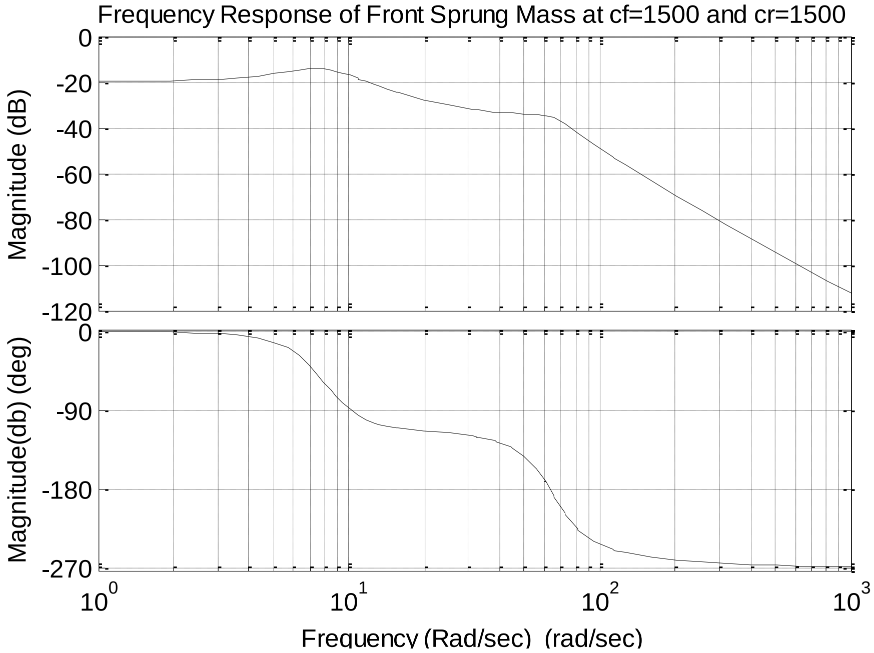

Figure 5 Frequency response of front sprung mass

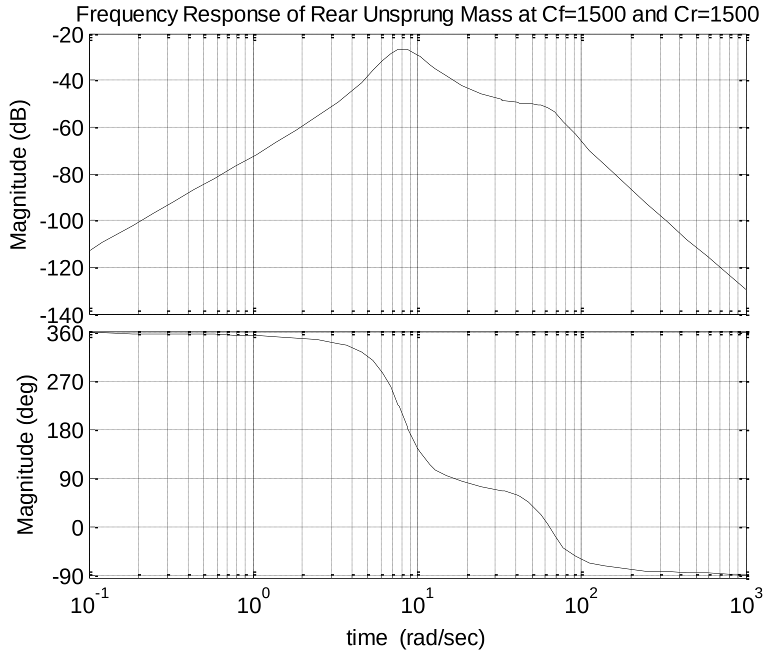

Figure 5 Frequency response of rear u sprung mass

4. CONCLUSION AND RECOMMENDATIONS

The main objective of the paper is to show how helpful is MATLAB to students to automotive engineering in order to understand problems of analysis and control of vibrations.

We have described the usefulness of MATLAB as a teaching and research aid in engineering automotive course via the addition of numerical simulation.

The analysis for the half car model considered in this paper is simulated using Matlab platform to optimize the damping value for better ride and passenger comfort. The displacement response and vertical acceleration of sprung and unsprung masses at different damping values are obtained.

The damping characteristic of the suspension damper has to be optimized in accordance with other vehicle dynamics parameter like sprung mass, unsprung mass, tire stiffness and damping, suspension stiffness and road input.

Matlab platform has been used to develop programs to obtain the system response.

In the engineering field, simulations are very powerful and useful tools used to catalyze the design process. The current numerical techniques in simulations approximate mathematical equations that are used to describe physical behavior of a dynamic system.

The tutorial problem presented in this paper shows the simplicity, efficiency, reliability, and accuracy of MATLAB as a pedagogical tool in engineering education. MATLAB should be introduced in engineering faculties and made compulsory for all engineering students.

References

[1] Gillespie, T. D.: Fundamentals of vehicle Dynamics, Society of Automotive Engineer, 1999.
[2] Hingham, D. J.: MATLAB Guide, Philadelphia Society for Industrial and Applied Mathematics, 2005.
[3] MathWorks, Inc.: MATLAB, the Language of Technical Computing, 2002.
[4] Mathworks, Inc., (1991) MATLAB User's Guide.
[5] Moore, H.: MATLAB for Engineers, Prentice Hall, 2007.
[6] Nicoară, D.: On Semi-Active Damping Control of Suspension Systems. In: Annals of the Oradea University, Fascicle of Management and Technological Engineering (2009) Vol. XVIII (VIII), p.413-418.
[7] Nicoară, D.: On the Design and Control of Active Vehicle Suspension. The Dual Objective Analysis. In: Proceedings of the 11th CONAT International Congress, Brasov, 2010, Vol.5, p.37-44.
[8] Ogata, K.: Modern Control Engineering, Prentice Hall International, 1997.
[9] Palm III. W. J.: Concise Introduction to MATLAB, McGraw-Hill, 2008.
[10] Sticklen, J., Taner Eskil, M., (2006) An Introduction to Technical Problem Solving with MATLAB v. 7, 2e, Great Lakes Press.
[11] Xue,.D., Chen, Y.Q., Atherton P. D., (2007) Linear Feedback Control Analysis and Design with MATLAB, Society for Industrial and Applied Mathematics (SIAM), ISBN 978-0-898716-38-2.