

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,800

Open access books available

142,000

International authors and editors

180M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Introductory Chapter: Matlab and Simulink Applications

Constantin Volosencu

1. Generalities and publications

In the scientific and technical field there are a multitude of numerical calculation programs. Some examples of these programs can be given as follows. Analytica, created by Lumina Decision Systems, is a numerical modeling environment with a visual programming language based on influence diagrams. LabView, created by National Instruments, is a graphical and textual through formula nodes software, for process monitoring and control. Mathcad, created by Parametric Technology Corporation, is a computer software for the verification, validation, documentation and re-use of mathematical calculations. Matlab is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. It allows numerical computation and simulation with extended 2D/3D visualization with vector manipulation. Matlab allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages. Simulink is a Matlab-based graphical programming environment for modeling, simulating and analyzing dynamical systems from different domains.

According to MathWorks presentation, the Matlab language fundamentals consists in syntax, operators, data types, array indexing and manipulation. Some of mathematics domain supported are linear algebra, differentiation and integrals, Fourier transform and other. The users may presents graphic results in two and three dimensional plots, images, animation and visualization. Data can be imported and exported, analyzed, preprocessed and visually explored. The language has many functions and assures programming scripts with program files, control flow, editing and debugging. The users may develop applications using App Designer, Guide or a programmatic workflow. Advanced software development is supported with object-oriented programming, code performance, unit testing, external interfaces to Java and Web services, C/C++, .NET and other languages. The language is a desktop environment with preferences and settings and platform differences. It assures support for third-party hardware, such as webcam, Arduino, and Raspberry Pi hardware. Simulink is a block diagram environment for multidomain simulation and model-based design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with Matlab, enabling users to incorporate Matlab algorithms into models and export simulation results to Matlab for further analysis. It allows modeling of time-varying systems and of large-scale architectures, running systems, reviewing results, validate system behavior, optimizing performance for specific goals. The users may extend the existing Simulink modeling functionality using Matlab, C/C++, and Fortran code.

It assures hardware support for third-party hardware, such as Arduino, Raspberry Pi, and Beagle Board.

The software has many applications in practice, which can be mentioned: signal processing, image processing and computer vision, control systems, test and measurement, radio-frequency and mixed signals, wireless communication, robotics and autonomous systems, automotive, aerospace, FPGA, ASIC, and SoC Development, computational finance, computational biology and the number of application is increasing.

Over the years, numerous books have been published that present applications of the Matlab and Simulink programs. Some examples from the last years can be highlighted, as follows. Some books dedicated to students and engineers, which presents fundamentals of Matlab may be mentioned. This books make introduction in basic programming, fundamentals consisting in data, statement structures, control structures, functions, algebraic computation, variables, complex numbers, vectors and matrices, data processing, examples of solving problems, examples in chemistry and physics, but also some advanced techniques for object-oriented programming, graphical user interface design and web applications [1–7]. More advanced issues as model predictive control or deep learning application are presented in [8] and respectively [9].

Extensive collections of works in the field of Matlab and Simulink applications, from the last years, can be cited as follows [10–20]. These collections, which can be used for educational, scientific and engineering purposes, include applications of: programming, developing graphical user interfaces, power system analysis, control systems design, system modeling and simulation, parallel processing, optimizations, signal and image processing, computer graphic visualization, electric machines, power electronics, genetic programming, digital watermarking, artificial networks, algebraic computation, data acquisition, image processing, seismology, meteorology, natural environment, interconnected power grids, antennas, underwater vehicles, models and data identification in biology, fuzzy logic, and discrete event systems.

Papers using the Matlab and Simulink programs have appeared and continue to appear in the literature. Here are some examples from the last years. A Matlab processing toolbox for analytical spectral devises field spectroscopy data, for generation of consistent and comparable ground spectra that have been corrected for viewing and illumination geometries as well as other factors such as the individual characteristics of the reference panel used during acquisition [21]. A software development platform is used in [22] for speedy evaluation and implementation of image processing options on the automatic guided vehicles. A program code written in Matlab, designed to be used inside of a Simulink model in [23], allows a fuel cell model to be used in a wide variety of 1D simulation platforms by exporting the code as C/C++.

2. Examples

2.1 A hyperbolic partial differential equation

The following example is realized using the *PDE modeler* toolbox. With this application the users can analyze elliptic, parabolic and hyperbolic Eqs. A hyperbolic equation case study, for wave propagation in square domain in plane, is presented in this example [24, 25]. The equation used in analysis is:

$$\frac{\partial u^2}{\partial t} = c_1 \nabla(c_2 \nabla u) + c_3 u + c_4 \quad (1)$$

where the parameter have the following values: $c_1 = 1, c_2 = 1, c_3 = 0, c_4 = 10$.

The space on which is made the analysis is a square with unitary dimension $l = 1$. Boundary conditions were imposed as follows: on the left, right and front Dirichlet conditions: $h = 1, r = 0$. On the square's base Neumann conditions: $q = 0, g = 0$.

The discrete optimized number and position of meshes are presented in **Figure 1**. The contour solution is presented in **Figure 2**. For these meshes the approximated solution is presented in 3D in **Figure 3**.

2.2 Modeling and simulation of a control system for a second order process

The second example presents a simple case of modeling and simulation of a basic control system for a second order process.

The process has the transfer function:

$$H_p(s) = \frac{y(s)}{u(s)} = \frac{K_L}{(T_1s + 1)(T_s + 1)} \quad (2)$$

where u is process input and y is process output.

The following values are chosen for the process parameters: $T_1 = 0,4 \text{ s}, T_s = 0,04 \text{ s}$ and $K_L = 2$.

The process has a disturbance v at its input.

A linear PI controller is chosen, with the transfer function:

$$H_R(s) = \frac{u_c(s)}{e(s)} = K_R \left(1 + \frac{1}{T_i s} \right) \quad (3)$$

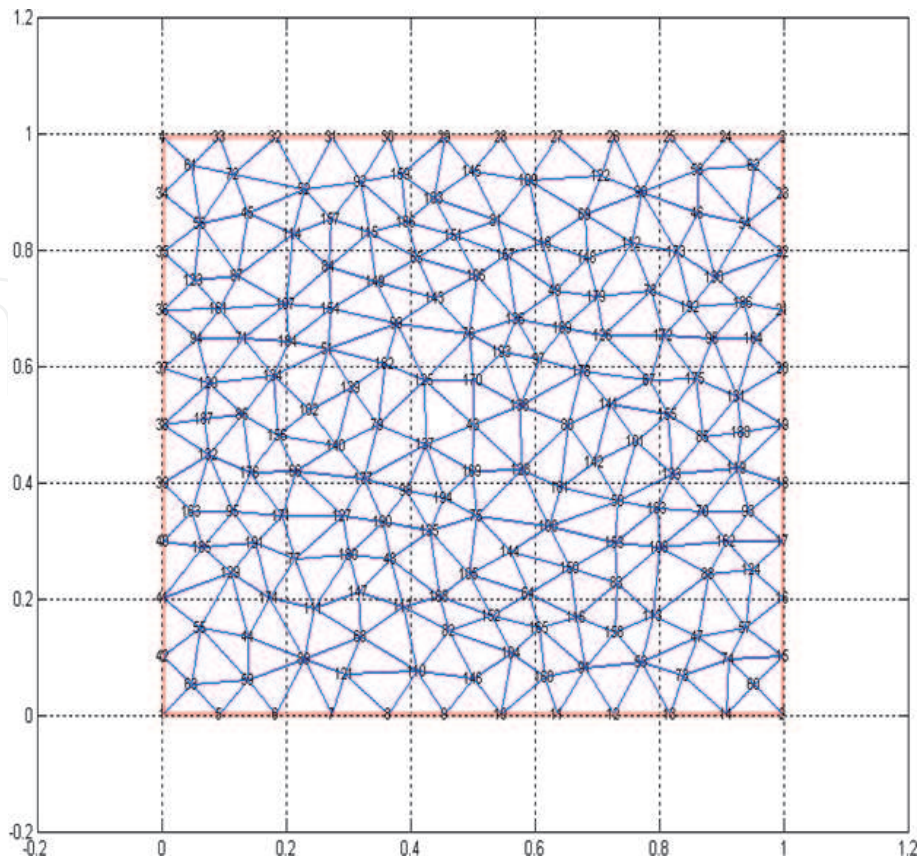


Figure 1.
 The optimized meshes.

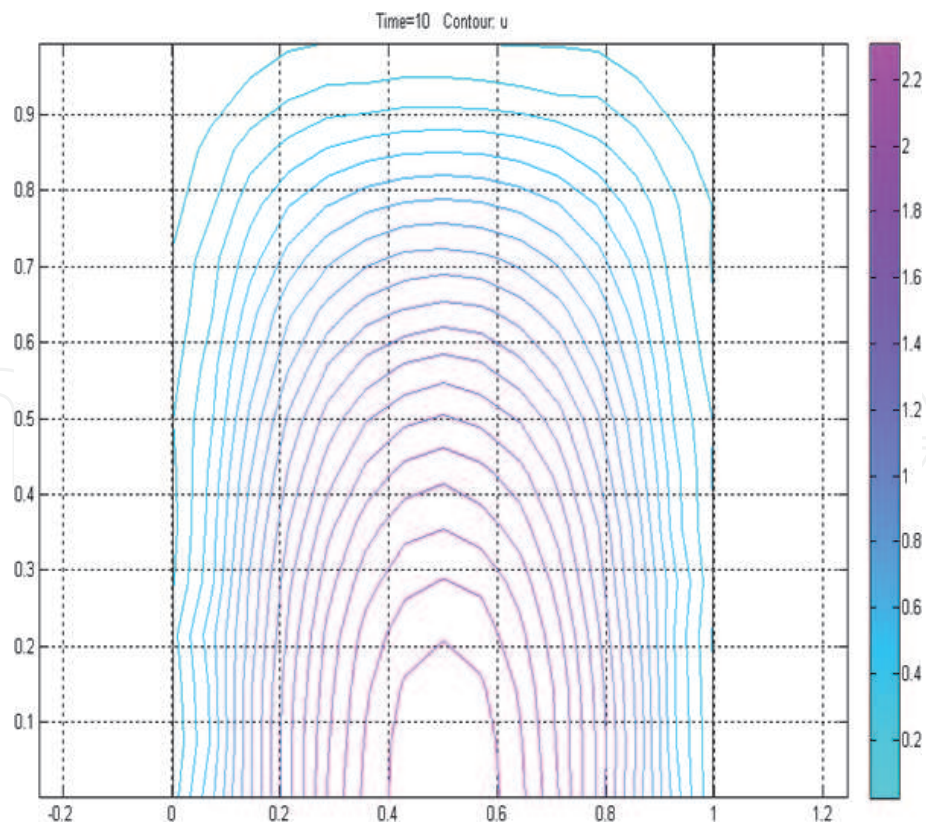


Figure 2.
Contour plotted solution.

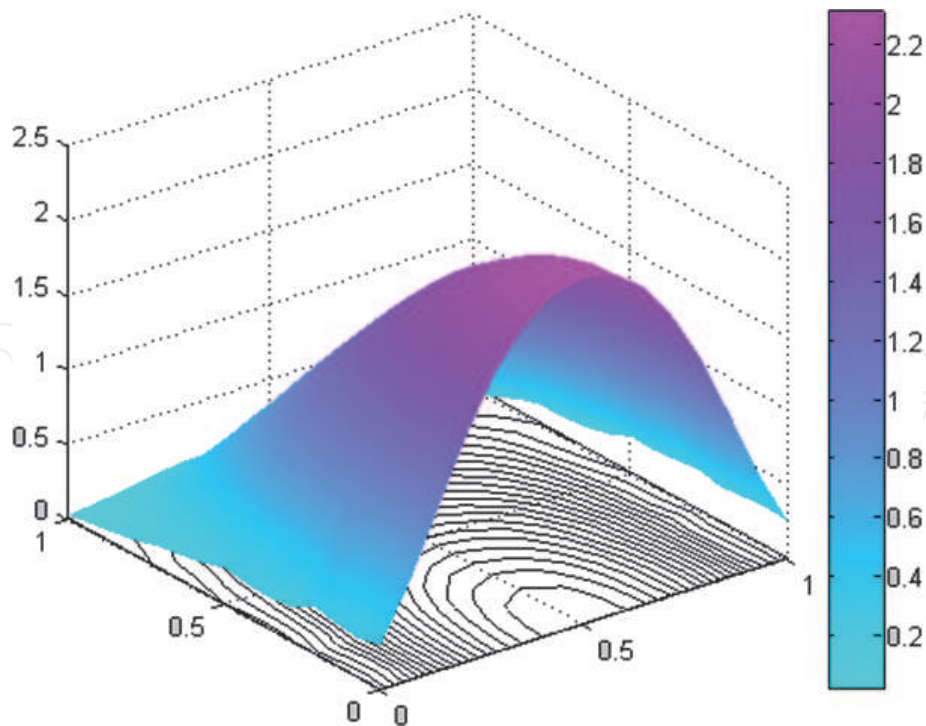


Figure 3.
Plotted solution in 3D.

where e is the error, as difference between the reference w and the feedback r and u_c is the command. The controller is tuned in accordance with the Kessler version of the module criterion:

$$T_i = T_1 = 0,4$$

$$K_R = \frac{T_1}{2K_L T_s} = 2,5 \quad (4)$$

In **Figure 4** shows how to arrange the work windows for this application on the screen. First, Matlab work space is open, then Simulink. A Simulink block diagram for the control system according the above theory is developed, with *transfer function* and *integrator* blocks from *Continuous* block library, *Add* and *Gain* blocks from *Math Operators*, *Step* and *Clock* blocks from *Sources*, *To Workspace* and *Scope* blocks from *Sinks*. The parameters are entered literally in the Simulink scheme, and their values are given in Matlab. The parameter values are saved in a data file, which is called with instruction *load* each time before the scheme is run. The values of the vectors *w*, *uc*, *v*, *u*, *y*, calculated at the tome values from vector *t*, are passed into Matlab workspace with the blocks *To Workspace* The time variations of the variables

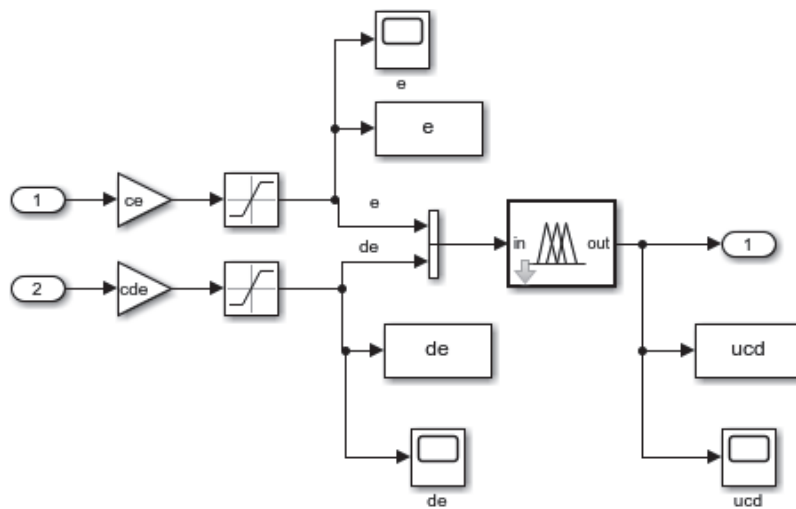


Figure 4.
 Simulink block diagram work screen.

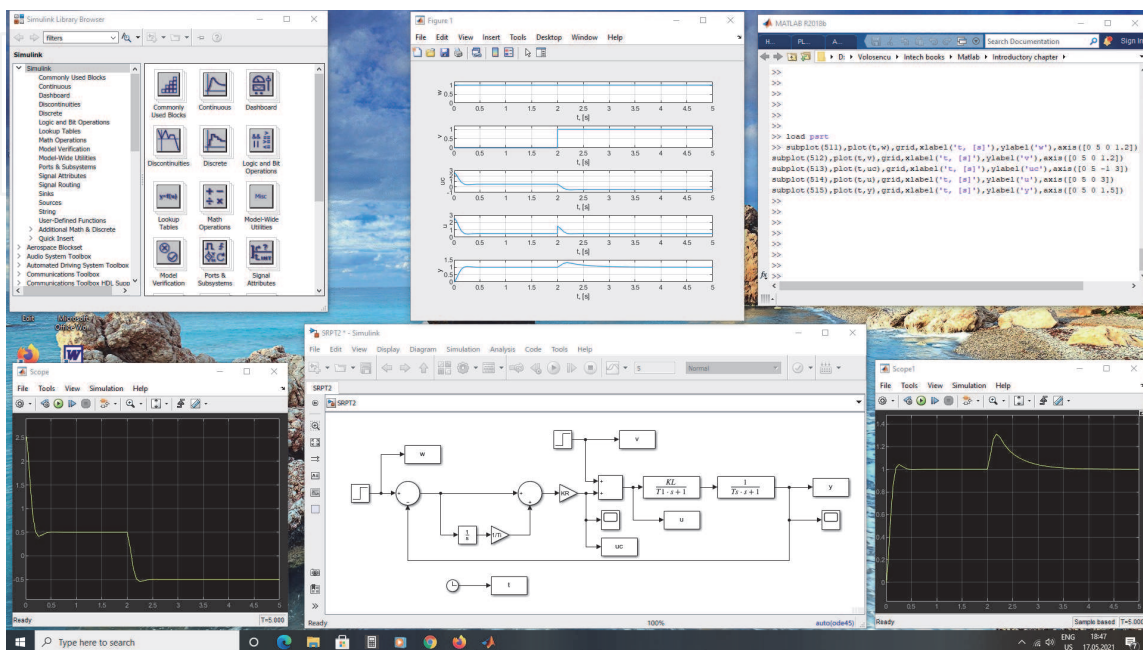


Figure 5.
 Simulink diagram for fuzzy block.

u_c and y are presented on the two scopes. The time variations of the variables w , v , u_c , u and y are presented using the instructions *subplot*, *plot*, *grid*, *xlabel*, *ylabel*, *axis*.

Analyzing the output graph y it can be seen that the overshoot $\sigma_1\% = 4,3\%$, the settling time $t_r = 8,4.T_s = 0,336$ s, in accordance with Kessler's tuning criterion.

2.3 Modeling and simulation of a fuzzy control system for the second order process

The third example presents a simple case of modeling and simulation of a basic fuzzy control system [26–29] for the same second order process like in the second example.

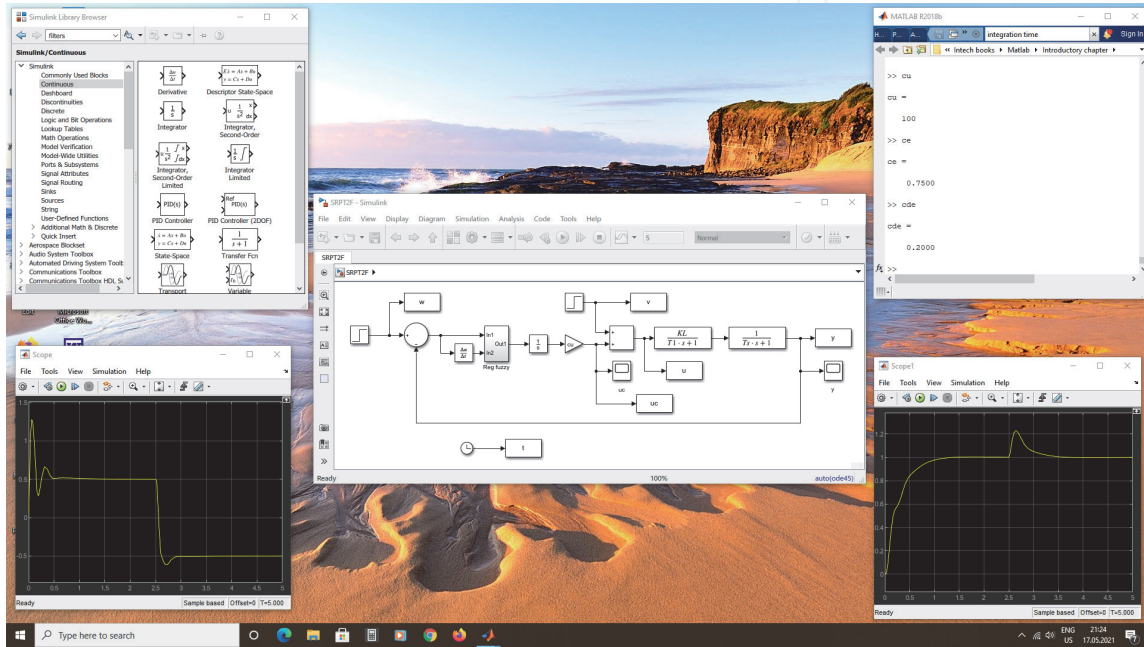


Figure 6. The screen for fuzzy control system.

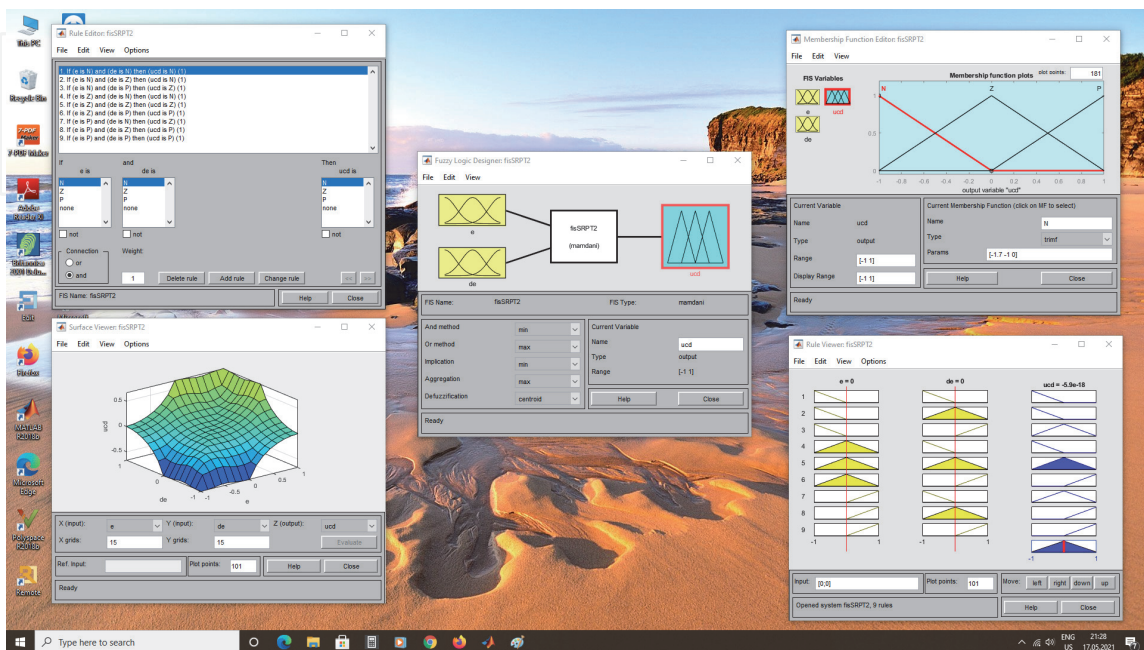


Figure 7. The screen for fuzzy system design.

The process has the same transfer function. A fuzzy PI controller is used.

In **Figure 5** shows Simulink block diagram for fuzzy controller. With the *fuzzyLogicDesigner* window the user may set membership functions, rule base and inference.

In **Figure 6** shows how to arrange the work windows for this application on the screen.

In **Figure 7** shows the work window for *fuzzyLogicDesigner*. The Simulink block diagram uses a fuzzy controller with derivation at the input and integration at the output. The working with Matlab and Simulink is the same like in the second example.

Conflict of interest


The author has no conflict of interest.

Author details

Constantin Volosencu
“Politehnica” University, Timisoara, Romania

*Address all correspondence to: constantin.volosencu@aut.upt.ro

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Chapman S.J. Matlab Programming for Engineers, 6th edition, Cengage Learning, 2020.
- [2] Xue D. Matlab Programming. Mathematical Problem Solutions, De Gruyter, 2020, DOI:10.1515/9783110666953.
- [3] Hahn B., Valentine D. Essential Matlab for Engineers and Scientists, Academic Press, 2019.
- [4] Lee H.H. Programming and Engineering Computing with Matlab, SDC Publications, 2019.
- [5] Moore H., Matlab for Engineers, 5th Edition, Pearson, 2018.
- [6] Attaway S. Matlab 5th Edition A Practical Introduction to Programming and Problem Solving Butterworth-Heinemann, 2018.
- [7] Kattan P.I. Matlab For Beginners: A Gentle Approach, Createspace, 2008.
- [8] Dittmar R. Model Predictive Control mit Matlab und Simulink, IntechOpen, London, UK, 2019, DOI: 10.5772/intechopen.86001.
- [9] Paluszek M., Thomas S. Practical Matlab Deep Learning: A Project-Based Approach, Apress, 2020.
- [10] Leite E.P., editor, Matlab Modelling, Programming and Simulations IntechOpen London UK 2010 DOI: 10.5772/242.
- [11] Assi A., editor, Engineering Education and Research Using Matlab IntechOpen London UK 2011 DOI: 10.5772/1532.
- [12] Ionescu C., editor, Ed. Matlab A Ubiquitous Tool for the Practical Engineer IntechOpen London UK 2011 DOI: 10.5772/82.
- [13] Leite E.P., editor, Scientific and Engineering Applications Using Matlab IntechOpen London UK 2011 DOI: 10.5772/1531.
- [14] Chakravarty S., editor,. Technology and Engineering Applications of Simulink IntechOpen London UK 2012 DOI: 10.5772/2414.
- [15] Katsikis V., editor, Matlab A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 1, IntechOpen, London UK, 2012, DOI: 10.5772/2557.
- [16] Katsikis V., editor,. Matlab A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 2 IntechOpen London UK 2012 DOI: 10.5772/3338.
- [17] Katsikis V., editor, Matlab. A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 3, IntechOpen, 2012, London UK, DOI: 10.5772/3339.
- [18] Bennett K., editor,. Matlab. Applications for the Practical Engineer, IntechOpen London UK, 2014, DOI: 10.5772/57070.
- [19] Valdman J., editor Applications from Engineering with Matlab Concepts IntechOpen London UK 2016 DOI: 10.5772/61386.
- [20] Saghafinia A., editor, Matlab. Professional Applications in Power System, IntechOpen London UK 2018, DOI: 10.5772/intechopen.68720.
- [21] Elmer, K.; Soffer, R.J.; Arroyo-Mora, J.P.; Kalacska, M. ASDToolkit: A Novel Matlab Processing Toolbox for ASD Field Spectroscopy Data. *Data* 2020, 5, 96. DOI: 10.3390/data5040096.
- [22] Kotze, B.; Jordaan, G. Investigation of Matlab as Platform in Navigation and

Control of an Automatic Guided Vehicle
Utilising an Omnivision Sensor.
Sensors 2014, 14, 15669-15686. DOI:
10.3390/s140915669.

[23] Lazar, A.L.; Konradt, S.C.;
Rottengruber, H. Open-Source Dynamic
Matlab/Simulink 1D Proton Exchange
Membrane Fuel Cell Model.
Energies 2019, 12, 3478. DOI: 10.3390/
en12183478.

[24] Voloşencu, C., Identification of
Distributed Parameter Systems, Based
on Sensor Networks and Artificial
Intelligence, WSEAS Transactions on
Systems, 2008, Issue 6, Vol. 7,
p. 785-801.

[25] Voloşencu, C. - Identification in
Sensor Networks, In: Proceedings of the
9th WSEAS International Conference on
Automation and Information (ICAI'08),
June 24-26, 2008; Bucuresti, WSEAS
Press, p. 175-183.

[26] Voloşencu, C., editor, Fuzzy Logic,
IntechOpen Ltd., London, UK, 2020,
DOI: 10.5772/Intechopen.77460.

[27] Volosencu, C., Introductory
Chapter: Basic Properties of Fuzzy
Relations, *Fuzzy Logic*, IntechOpen,
London, UK 2020, DOI: 10.5772/
Intechopen.77460.

[28] Volosencu, C., Tuning Fuzzy PID
Controllers, *Theory, Tuning and
Application to Frontier Areas*, edited by
Rames C. Panda, InTech, 2012. DOI:
10.5772/32750.

[29] Voloşencu, C., Stabilization of
Fuzzy Control Systems, WSEAS
Transactions on Systems and Control,
2008, Issue 10, Vol. 3, p. 879-896.