UNIVERSITY OF NOVI SAD
TECHNICAL FACULTY
"MIHAJLO PUPIN"
ZRENJANIN

ITROCONFERENCE 9.0

INFORMATION TECHNOLOGY AND EDUCATION DEVELOPMENT

ITROCONFERENCE 9.0
INFORMATION TECHNOLOGY AND EDUCATION DEVELOPMENT

PROCEEDINGS

ZRENJANIN, June 2018

UNIVERSITY OF NOVI SAD
TECHNICAL FACULTY ”MIHAJLO PUPIN”
ZRENJANIN
REPUBLIC OF SERBIA

IX INTERNATIONAL CONFERENCE OF

INFORMATION TECHNOLOGY AND DEVELOPMENT OF EDUCATION ITRO 2018

PROCEEDINGS OF PAPERS

IX MEĐUNARODNA KONFERENCIJA

INFORMACIONE TEHNOLOGIJE I RAZVOJ OBRAZOVANJA ITRO 2018

ZBORNIK RADOVA

ZRENJANIN, JUNE 2018

Publisher and Organiser of the Conference:
University of Novi Sad, Technical faculty „Mihajlo Pupin", Zrenjanin, Republic of Serbia

For publisher:
Dragica Radosav, Ph. D, Professor, Dean of the Technical faculty „Mihajlo Pupin", Zrenjanin, Republic of Serbia

Editor in Cheaf - President of OC ITRO 2018:
Vesna Makitan, Ph. D, Assistant Professor,

Proceedings editor:
Marjana Pardanjac, Ph. D, Professor

Technical design:
Ivan Tasic, Ph. D, Professor;
Dusanka Milanov MSc, Assistant

Lecturer:
Erika Tobolka, Ph. D, Professor

Circulation: **50**
ISBN: 978-86-7672-310-2

By the resolution no. 142-451-607/2018-01/02, Autonomous Province of Vojvodina, Provincial Secretariat For Science and Technological Development donated financial means for printing this Conference Proceedings.

The Conference is supported by the Autonomous Province of Vojvodina

CIP - Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

37.01:004(082)
37.02(082)

INTERNATIONAL Conference on Information Technology and Development of Education ITRO (9 ; 2018 ; Zrenjanin)
Proceedings of papers [Elektronski izvor] / IX International Conference on Information Technology and Development of Education ITRO 2018 = Zbornik radova / IX međunarodna konferencija Informacione tehnologije i razvoj obrazovanja ITRO 2018, Zrenjanin, June 2018. - Zrenjanin : Technical Faculty "Mihajlo Pupin", 2018. - 1 elektronski optički disk (CD-ROM) : ilustr. ; 12 cm

Nasl. sa naslovnog ekrana. - Tiraž 50. - Bibliografija uz svaki rad.

ISBN 978-86-7672-310-2
I. Međunarodna konferencija Informacione tehnologije i razvoj obrazovanja ITRO (9 ; 2018 ; Zrenjanin) International Conference on Information Technology and Development of Education ITRO (9 ; 2018 ; Zrenjanin)
a) Информационе технологије - Образовање - Зборници b) Образовна технологија - Зборници
COBISS.SR-ID 324298503

PARTNERS INTERNATIONAL CONFERENCE

South-West University „Neofit Rilski”
Faculty of Education, Blagoevgrad,
Republic of Bulgaria

Faculty of Electrical Engineering and Informatics
Department of Computers and Informatics of Kosice
Slovak Republic

University Goce Delcev Stip
Republic of Macedonia

THE SCIENCE COMMITTEE:

THE ORGANIZING COMMITTEE:

With this publication, the CD with all papers from the International Conference on Information Technology and Development of Education, ITRO 2018 is also published.

INTRODUCTION

Technical Faculty "Mihajlo Pupin" organized, now the traditional, IX International Conference on Information Technology and Education Development (ITRO 2018), which was held on June 29, 2018.

This year we managed to gather our colleagues, scientists, researchers and students from 10 countries (Serbia, Macedonia, Bulgaria, Bosnia and Herzegovina, Romania, USA, Great Britain, Albania, Montenegro, Slovakia). Many of them have been participating in the work of the Conference for many years and practically they are making an ITRO family. With their papers they managed to present and promote the results of research and scientific work in the field of information technology in education. More than 40 papers have been collected, which will be published in the Proceedings of the Conference website too (http://www.tfzr.rs/itro/index.html).

The main course in the work of the Conference was set up with introductory lectures in which the significance of following topics could be seen:

- Education for modern business and education from the perspective of employers nowadays when every company is directly or indirectly IT company – lecture with the topic "Digital transformation of the society – the role of education" was held by Goran Đorđević, director of the company Consulteer;
- Scientific research work in the field of information technology in education, whose results were published in one of the world's leading magazines – this novelty at the ITRO Conference was introduced by PhD Dragana Glušac with a lecture on "School without walls";
- The latest forms of education and practice of IT experts in the country and abroad – a lecture on the topic "Finding a space for "making" and digital fabrication in the education of Serbia" was held by PhD Dalibor Dobrilović.

The other presented papers have cast light on various aspects of contemporary education in our country and abroad, as well as on the experiences, problems, questions, etc. which are related to them.

The conference was an opportunity to connect again with researchers and scientists from other institutions and countries and ask questions about new forms of cooperation and projects that are relevant to all of us.

The conference was held thanks to the sponsorship of the Provincial Secretariat for Higher Education and Scientific Research, which also traditionally supports ITRO, as well as the Faculty, which provided the necessary technical conditions.

We thank everyone for participating and creating the ITRO tradition.

See you at the next ITRO Conference,

Chairman of the Organizing Committee
PhD Vesna Makitan

We are very grateful to:

Autonomous Province of Vojvodina

for donated financial means which supported printing of the Conference Proceedings and organizing of the Conference.

CONTENTS

INIVITED LECTURE

SCIENTIFIC PAPERS

Application of Machine Learning in Software Engineering

A. Krstev*, L. Beqiri*, D. Zlatev*, D. Krstev **, B. Krstev***, A. Donev****
* Faculty of Computer Science - Goce Delchev University, Stip, Republic of Macedonia
**Faculty of Mechanical Engineering - Saints Cyril and Methodius University, Skopje, R. of Macedonia
***Faculty of Natural and Technical Sciences - Goce Delchev University, Stip, Republic of Macedonia
**** University of Buckingham, Buckingham, United Kingdom
aleksandar.krstev@ugd.edu.mk, boris.krstev@ugd.edu.mk

Abstract - The purpose of the software manufacturing industry is to produce high-quality applications that meet the requirements of customers and users who live long, that are easy to use and have as few errors as possible. Building such an ideal software is a relatively difficult process. To be successful in this industry, a specific discipline is needed when designing and developing software. There is therefore an engineering perspective on the whole process.

Many companies and individuals still develop software chaotic, based on a poor analysis, which leads to unsuccessful outcomes such as software failures that fail to meet the expected requirements. Software Engineering applies to optimize these phenomena.

I. Introduction

Software Engineering - The purpose of the software manufacturing industry is to produce high-quality applications that meet the requirements of customers and users who live long, that are easy to use and have as few errors as possible. Building such an ideal software is a relatively difficult process. To be successful in this industry, a specific discipline is needed when designing and developing software. There is therefore an engineering perspective on the whole process.

Many companies and individuals still develop software chaotic, based on a poor analysis, which leads to unsuccessful outcomes such as software failures that fail to meet the expected requirements. Software Engineering applies to optimize these phenomena.

Machine learning does the statistical analyses is of data and extract and use most relevant data for solving the current situation or for prediction of future events. Machine Learning is the branch of Artificial Intelligence. Machine learning means skills to learn things from activities as well as conditions that give relevant results. It became a well-known topic due to its usages and therefore many people are trying to learn it. The Learning for machines is nothing but to become more familiar with the thing in such a way that can help in many ways like weather prediction, recommendation based on the taste, deciding route in traffic, diagnosing samples with most accurate output, etc.

Machine learning (ML) is not hard. Machine learners automatically generate summaries of data or existing systems in a smaller form. Software engineers can use machine learners to simplify systems development. This chapter explains how to use ML to assist in the Designing a simulator for monitoring CPU temperature through fan speed control based on Fuzzy Logic theory.

A. *Software applications and their categorization*

Whenever there is an algorithm for solving a problem, the latter may be subjected to a software application for solving it (except for some cases of Artificial Intelligence). Content of information and the selective are two very important factors that affect the nature of a software application.

- The content of the information relates to the meaning and form of input and output information circulating in the application. Examples of possible forms of information are files, databases and data structures, images, inputs from peripheral devices, and so on.
- Determination of information refers to the predictability of the order and the time of exchange and manipulation of information. Applications that receive a particular data format that apply a certain algorithm and instructions on a timely basis are called determinant applications. An application is called non-determinant if it has variants of information content, arbitrary execution of instructions and algorithms that may be interrupted by external factors whose output varies depending on the environment and time. A multiuser

operating system, for example, is not decisive. Such applications are usually more complex and more difficult to manage.

Though in the wide range of applications used today it is difficult to make a clear categorization, it is necessary to set inclusive spaces for study purposes for them. Such are:

- **Software systems.** System software is a set of programs that serve other programs. Such are compilers, parsers, file management systems, drivers, operating systems, and so on. These applications are characterized by:
  - Close interaction with hardware
  - Multi-user
  - Simultaneous actions
  - Complex data structure
  - Many external interfaces
- **Real-time systems.** These types of software monitor, analyze, and control real-world events as they occur. These applications are characterized by:
  - Components that collect data from an external environment and format them for manipulation.
  - Components that analyze information and transform according to application requirements.
  - Components that control output.
  - Steering components that coordinate the work of other components so that the system responds in real time.
- **Business Software.** This is also the largest category of software. These applications mainly deal with the storage and access to data related to business information. Typically, they are characterized by normalized and large-scale databases as well as high user interactivity.
- **Scientific and engineering software.** These software almost always include algorithms and complex calculations. Fields of application are astronomy, molecular biology, applied mathematics, physics and so on. Today, scientific software is not just software calculator but is trying to simulate, interact with systems and feature real-time software.
- **Interfaced Sotware.** Software that comes as part of the industry and its products. These software stay in the short-term memory of the device and serve to control and automate its work. Embedded software can perform only a few limited functions, limited is the interaction of users with these systems.
- **Software for personal computers.** The software market for personal computers has occupied a lot in the overall production of software industry. There are many examples: word-processing software, graphics, personal, multimedia, games, access to databases, etc.
- **Web-based software.** They are accessed through browsers and are based on the worldwide network of computers, the Internet. The data in it is presented in various forms and easily accessible to the user, such as hypertext and multimedia formats. Every day the importance and demand for these kinds of applications increases due to increased demands for communication and exchange of information between individuals and companies.
- **Artificial Intelligence Software.** Use non-numeric algorithms to solve complex problems that cannot be solved by traditional methods. Such are expert systems, knowledge-based systems, neutral nets, proofing of theorems, intelligent games etc.

In addition to the above categories, the software is also divided into two large groups: custom and custom-built custom software. One of the most noticeable changes from the point of view of engineering of these two different categories is the fact that in the first case the specifications are controlled by the development organization itself, whereas in the second case the specifications are determined by the organization which orders the program. However, the boundary between these categories is somewhat unclear. It often happens that companies start with a general product and later begin to tailor the product depending on the requirements of potential customers.

B. Methodology and research results - Machine Learning in Software Engineering

Machine learning deals with the issue of how to build computer programs that improve their performance at some tasks through experience. Machine learning algorithms have proven to be of great practical value in a variety of application domains. Not surprisingly, the field of software engineering turns out to be a fertile ground where many software development and maintenance tasks could be formulated as learning problems and approached in terms of learning algorithms.

Machine learning is practical for software engineering problems, even in data-starved domains. When data is scarce, knowledge can be farmed from seeds; i.e. minimal and partial descriptions of a domain. These seeds can be grown into large datasets via simulations. The datasets can then be harvested using machine learning techniques. Examples of this knowledge farming approach, and the associated technique of data-mining, is given from numerous software engineering domains.

Machine learning (ML) is not hard. Machine learners automatically generate summaries of data or existing systems in a smaller form. Software engineers can use machine learners to simplify systems development. This chapter explains how to use ML to assist in the Designing a simulator for monitoring CPU temperature through fan speed control based on Fuzzy Logic theory.

C. Fuzzy Logic

The Fuzzy Logic Toolbox for use with MATLAB is a tool for solving problems with fuzzy logic. Fuzzy logic is a fascinating area of research because it does a good job of trading off between significance and precision-something that humans have been managing for a very long time.

Fuzzy logic sometimes appears exotic or intimidating to those unfamiliar with it, but once you become acquainted with it, it seems almost surprising that noone attempted it sooner. In this sense fuzzy logic is both old and new because, 1-3 although the modern and methodical science of fuzzy logic is still young, the concepts of fuzzy logic reach right down to our bones.

D. Create Application of Machine Learning using Matlab – Fuzzy Logic Toolbox

The Fuzzy Logic Toolbox allows you to do several things, but the most important thing it lets you do is create and edit fuzzy inference systems. You can create these systems by hand, using graphical tools or command-line functions, or you can generate them automatically using either clustering or adaptive neuro-fuzzy techniques.

If you have access to Simulink, the simulation tool that runs alongside MATLAB, you can easily test your fuzzy system in a block diagram simulation environment. If you have RealTime Workshop capabilities available, you can generate realtime or non-realtime code from the Simulink environment.

The toolbox also lets you run your own stand-alone C programs directly, without the need for Simulink. This is made possible by a stand-alone Fuzzy Inference Engine that reads the fuzzy systems saved from a MATLAB session (the stand-alone code, unlike that generated by the Real-Time Workshop, does not run in real time). You can customize the stand-alone engine to build fuzzy inference into your own code. All provided code is ANSI compliant.

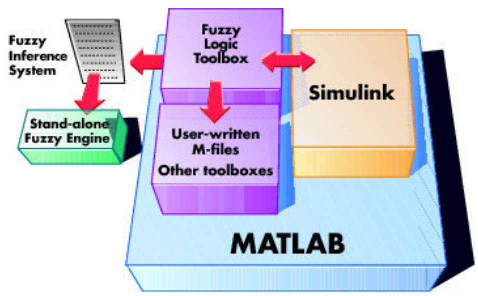


Figure 1 - Matlab's integration scheme

Because of the integrated nature of MATLAB's environment, you can create your own tools to customize the Fuzzy Logic Toolbox or harness it with another toolbox, such as the Control System, Neural Network, or Optimization Toolbox, to mention only a few of the possibilities.

E. Designing a simulator for monitoring CPU temperature through fan speed control based on Fuzzy Logic theory

The CPU temperature ranges from 0 to 110°C, the work of the computer must be stopped above this temperature. The fan usually works at four rotational speeds.

Let's look at the use of the Fuzzy Logic Regulator in this case.

1. If the temperature is 0 to 30°C then the fan rotates at a low speed, speed 1
2. If the temperature is between 30°C and 60°C then the fan rotates at an increased speed: speed 2
3. If the temperature is between 60°C and 90°C then the fan rotates at normal speed: speed 3
4. If the temperature is between 90°C and 110°C then the fan rotates at a high speed: speed 4

These are also the rules of the Fuzzy Logic Regulator.

The FL-FIS file Regulator is the CPU temperature of 0°C to 110°C. The FIS-FL file will be named the Sugeno type *ventilatori*.
Exit from Regulator FL should be the speed of fan rotation, i.e. speeds 1, 2, 3 and 4.
Let's present the FL rules in tabular form:

Speed of fan / Temperature of the CPU	Low Speed (shV)	Average speed (shm)	Speed normal (shN)	High speed (shM)
0°C to 30°C (tV)	X			
30°C to 60°C (tm)		X		
60°C to 90°C (tN)			X	
90°C to 110°C (tM)				X

Input, CPU temperature "T" it should be assumed that it may be in the interval [0, 110], and this interval is divided into sub-intervals that are accompanied by language names:

tV: Low Temperature [0, 30] ,
tm: Average temperature [30, 60],
tN: Normal temperature [60, 90],
tM: High temperature [90,110], see Figure 2.

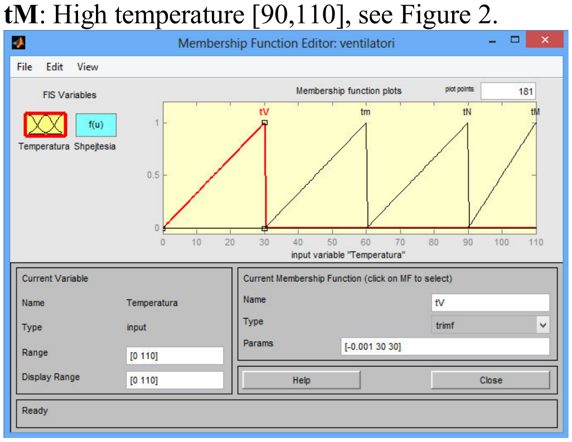


Figure 2 - Incoming Variable - CPU Temperature

Output, the swing speed of the fan "sh" should be assumed to be within the interval [1, 4], (because the fan starts rotating as soon as the computer is switched on) and this interval is divided into subintervals that are associated with the language names:
shV: the speed of the fan rotation is small,
shm: the speed of the fan rotation is average,
shN: the speed of the fan rotation is Normal,
shM: the speed of the fan rotation is high, Figure 3.

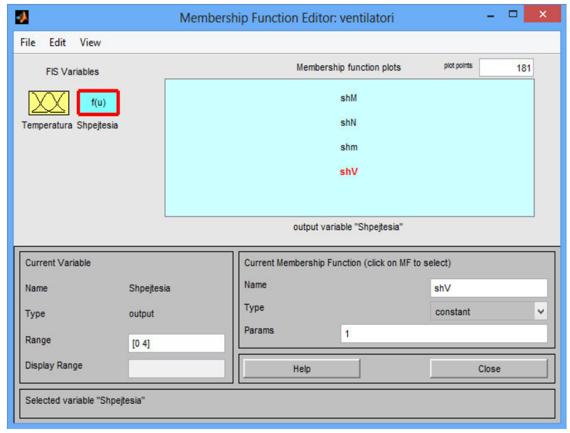


Figure 3 - Outbound Variable - Fan Speed Rotation

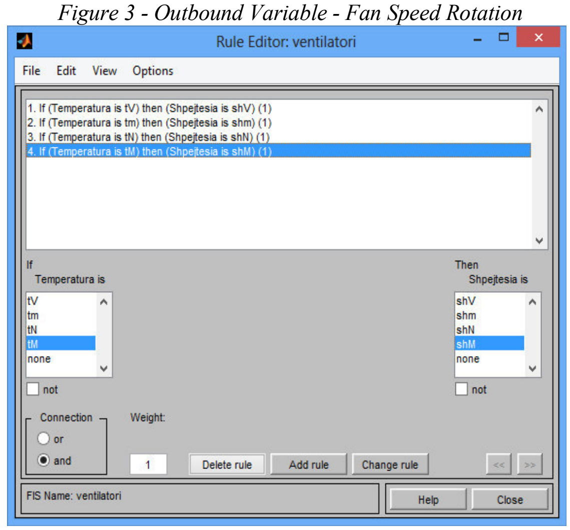


Figure 4 - Rule set for FIS-FL

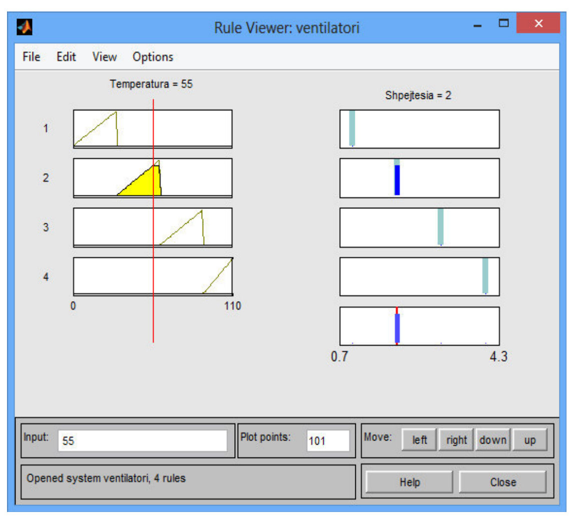


Figure 5 - Rules from the "View> Rules" menu

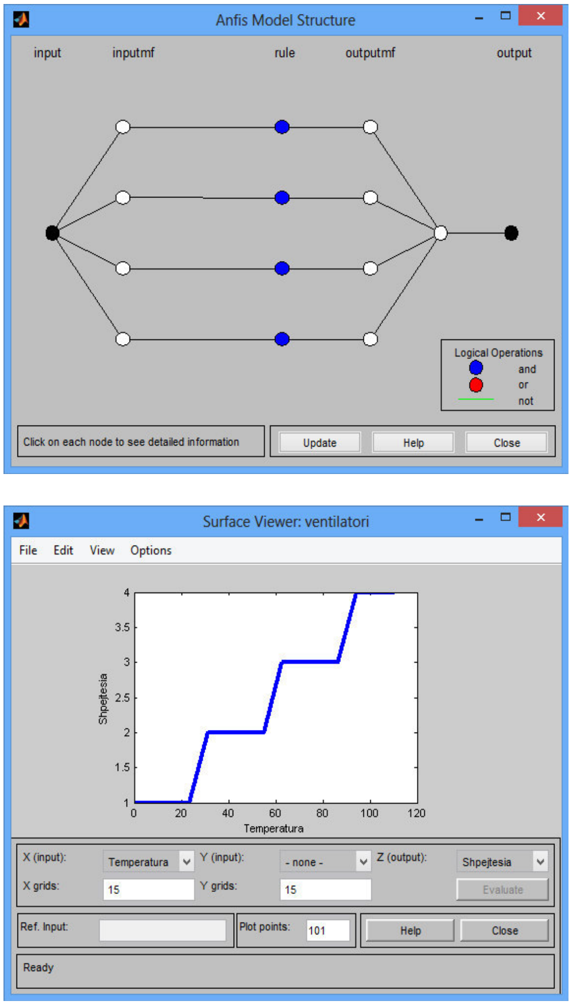


Figure 6 - Edit view> Anfis> Structure of FIS-FL and Rules from the "View> Surface" menu

After we have saved FL in the fan, in the file-fis and "File> Export> To Workspace". In the "Workspace" window, click on the right mouse button and save it as **ventilatori.mat** so that you can more easily activate FIS-FL in the **ventilatori**.

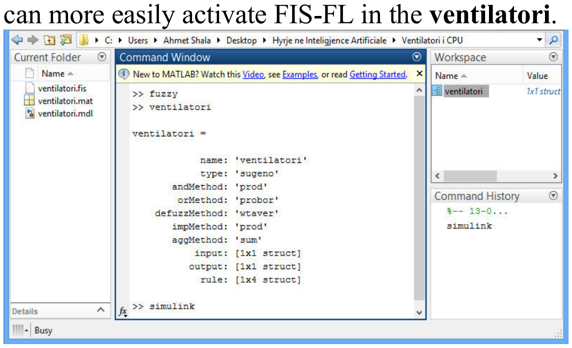


Figure 7 - Maintain FIS-FL

Using FL 'ventilatori" in Matlab> simulink

In the "Command Window" we write "simulink" and open a new model "File> New Model", we name ventilatori.mdl, we also simulated the change of CPU temperature through the absolute value of sinusoidal function with amplitude 120 and frequency 0.5, designed as follows:

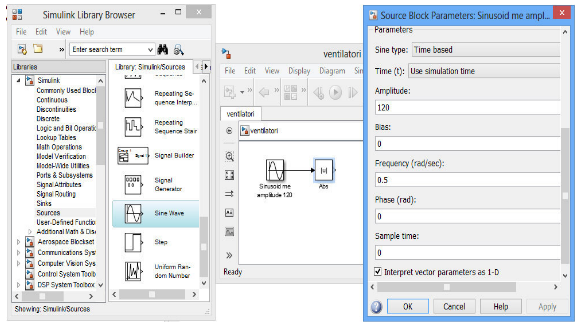
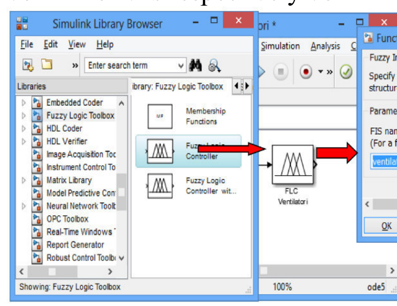


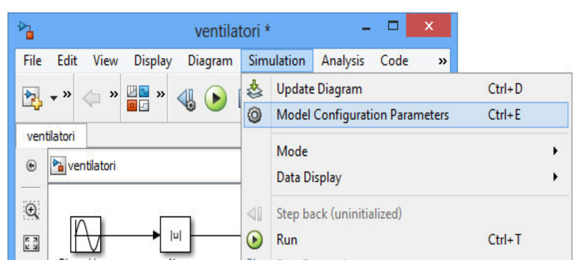
Figure 8 - Simulation of the CPU fan controller, being designed

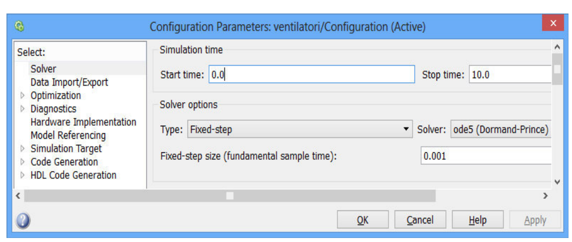
We have designed the Fuzzy Logic Regulator for CPU temperature control and saved it as **ventilatori.fis** respectively **ventilatori.mat**.



At the output, set a "Scope" to display the fan speed chart.

The simulation parameters are given below:





The simulation is finally presented in Figure 9.

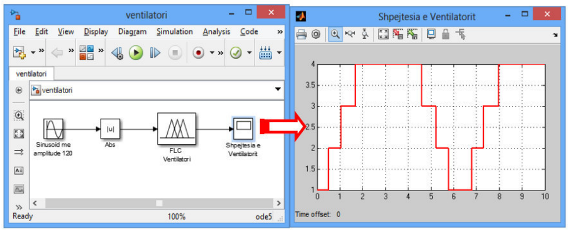
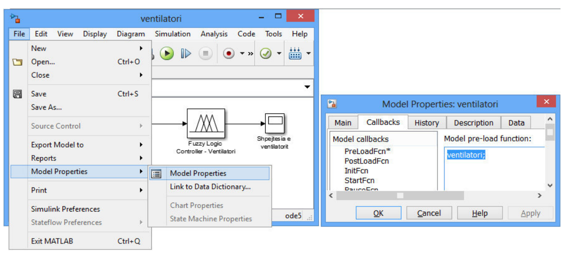


Figure 9 - CPU fan speed control simulation model

To simulate the next time, you do not need to activate the regulator in Command Window, we will automatically run the ventilatori.mat, when launching the ventilatori.mdl model, by doing the following:



II. Conclusion

Software engineering really involves a lot of analytical and documentary work and less coding. Usually as software engineers have been named creative individuals, with new ideas, who know how to manage a project and who certainly have enough programming experience. Professional ethics and software applications and their categorization.

Design of a CPU temperature monitor simulator through fan speed control based on Fuzzy Logic theory. The step-by-step process of processing the application for CPU temperature monitoring through the fan speed control is described. The result of the work, this is going to work machine-learning.

We used 4 types of speed and the machine should know which fanter quickly worked. The first speed works if the temperature is between 0 to 30°C, the second speed between 30°C and 60°C, the third speed between 60°C and 90°C and the speed 4 between 90°C and 110°C. Application of Machine Learning in Software Engineering is done in MatLab, and the result is successful based on the presentation.

References

[1] J.S. Roger Jang, Ned Gulley: MatLab Fuzzy LogicToolbox
[2] Nathaniel Borenstein: Programming as if People Mattered
[3] Grady Booch: Object Oriented Analyses and Design
[4] S. K. Chang: Machine Learning Applications in Software Engineering
[5] Michael Marsalli: McCulloch-Pitts Neurons